FMDB Transactions on Sustainable Computing Systems

A Comprehensive Real-Time Traffic Monitoring and Visualization System with Automated Vehicle Analysis

J. Angelin Jeba[1,*], S. Rubin Bose[2], R. Regin[3], P. Banupriya[4], Andino Maseleno[5]

[1]Department of Electronics and Communication Engineering, S.A. Engineering College, Chennai, Tamil Nadu, India.
[2,3]School of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.
[4]Department of Science and Humanities, Dhaanish Ahmed College of Engineering, Chennai, Tamil Nadu, India.
[5]Department of Information Systems, Bakti Nusantara Institute, Lampung, Indonesia.
angelinjeba@saec.ac.in[1], rubinbos@srmist.edu.in[2], reginr@srmist.edu.in[3], banupriya@dhaanishcollege.in[4], andino.maseleno@ibnus.ac.id[5]

Abstract: Urban traffic monitoring is essential for effective traffic management and public safety. However, traditional traffic monitoring systems face significant challenges, including high implementation costs, computational complexity, and an inability to provide real-time analysis, leading to delayed responses to traffic incidents and inefficient resource allocation. This paper presents a simplified real-time traffic monitoring system that leverages pre-trained YOLOv8 for vehicle detection and a basic web dashboard for visualisation. The problem statement addresses the critical need for affordable, efficient, and real-time traffic monitoring solutions that can operate with minimal computational resources while maintaining high accuracy. The methodology involves processing live video streams from webcams or traffic cameras through a YOLOv8-based vehicle detection pipeline that classifies vehicles into four categories (cars, trucks, buses, and motorcycles) with high accuracy. The system then processes the detection results to generate real-time traffic analytics, including vehicle counts, classification breakdown, and performance metrics, which are displayed through an intuitive web interface. Experimental results demonstrate the system's ability to detect vehicles with high accuracy (90% mAP) and low latency (25-30 FPS), making it suitable for deployment in resource-constrained environments. The implementation focuses on simplicity and practicality, using minimal computational resources while providing valuable traffic insights for traffic management centres and urban planning applications.

Keywords: Vehicle Detection; Traffic Visualisation; Vehicle Counting; Real-Time Traffic Analysis; Surveillance Systems; Web Dashboard Analytics; Deep Convolution Neural Network.

Received on: 03/01/2025, **Revised on:** 29/03/2025, **Accepted on:** 24/05/2025, **Published on:** 09/12/2025

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSCS

DOI: https://doi.org/10.69888/FTSCS.2025.000526

Cite as: J. A. Jeba, S. R. Bose, R. Regin, P. Banupriya, and A. Maseleno, "A Comprehensive Real-Time Traffic Monitoring and Visualization System with Automated Vehicle Analysis," *FMDB Transactions on Sustainable Computing Systems*, vol. 3, no. 4, pp. 242-252, 2025.

1. Introduction

[*]Corresponding author.

Traffic congestion and monitoring have become significant challenges worldwide in urban areas. Traditional traffic monitoring systems often require complex infrastructure, expensive hardware, and specialised expertise, making them inaccessible to many communities. This research addresses the need for a simple, cost-effective solution that can be quickly deployed with minimal technical requirements. Our approach utilises the power of pre-trained deep learning models, specifically YOLOv8, which has demonstrated excellent performance in object detection tasks. By combining this with a lightweight Flask-based web application, researchers create a system that is both powerful and accessible. The primary goal is to provide real-time vehicle detection without extensive model training or complex data processing pipelines. This research addresses the gap between advanced traffic monitoring technologies and practical deployment constraints by presenting a simplified system that leverages pre-trained models and web technologies. Our approach eliminates the need for custom model training by utilising YOLOv8's general object detection capabilities, which have been trained on the large-scale COCO dataset containing various vehicle types. This allows us to focus on system integration and user interface design rather than model development. The primary objectives of this research are: to develop a real-time vehicle detection system using pre-trained deep learning models; to create a web-based dashboard that makes traffic data accessible to non-technical users; and to minimise computational requirements while maintaining acceptable accuracy and performance.

2. Literature Review

Kheder and Mohammed [1] introduced the utilisation of a developed application for traffic monitoring based on a provided training model. The application demonstrates commendable model accuracy, particularly highlighting YOLOv3's effectiveness. With an 80:20 training-to-validation data ratio and 150,000 training iterations, YOLOv3 achieved a notable detection rate of 89.2%. The trained model can instantly recognise and detect vehicles in traffic scenarios. Traffic management centres successfully employ this application to identify traffic congestion on highways and urban roads, while urban planners utilise it to analyse traffic patterns in metropolitan areas. Zhu et al. [2] proposed a method, the Deep Convolutional Neural Network (DCNN), for identifying vehicle types and traffic density. The DCNN algorithm effectively classifies vehicles with high accuracy. In addition, it displays images of the detected vehicles with classification labels to improve results. This allows the algorithm to be used for other purposes, such as identifying traffic violations and monitoring traffic flow, in which case it triggers alerts and notifies traffic authorities. Rambabu et al. [3] describe the implementation of a traffic congestion detection system. An image processing system confirms the congestion level after it is spotted using traffic sensors. An alert message indicating the congestion of severity is delivered if its existence is confirmed. The alert prompts the taking of the necessary steps.

Traffic will eventually be managed more efficiently with a technology designed for them. When a congestion is detected, an automated traffic signal adjustment system that optimises signal timing based on real-time conditions can be activated. Prabhu et al. [4] propose a solution to the problem of traffic accidents, which have become a significant social issue. It needs to be addressed immediately and effectively. As a result, this paper has significant social significance because it seeks to solve this issue. It was discovered that the proposed system, based on edge computing devices, is more portable, easier to use, and simpler to implement. Many time-consuming and monotonous chores. This paper procedure is entirely automated, and the system doesn't interfere with normal traffic flow. He et al. [5] suggested an AI-driven solution for the effective detection of diverse traffic conditions. With the capability to minimise false alarms, the system can undergo training to recognise normal traffic patterns and anomalies, autonomously managing all functions. Program-based actions are executed automatically, enabling traffic operators to monitor and manage traffic flow remotely. Real-time detection of traffic incidents ensures prompt responses. Notably energy-efficient and operating at very low voltage, this technology imposes no harm on infrastructure or traffic systems. Despite its low development costs, the system's potential applications are vast. Addressing the challenge, a prototype utilising traffic cameras and sensors has been developed to track traffic conditions. Subsequent analysis of acquired data, achieved with an average accuracy of 93.75% using edge computing modules, facilitates appropriate actions.

Kosuru et al. [6] introduced an IoT-driven automated traffic management system integrated with machine learning, surpassing existing methodologies by achieving 92% accuracy with SVM classification. To ensure transportation safety, the architecture leverages IoT, computer vision, and edge computing. Sensory data from the roads is relayed to the edge devices, enabling smart traffic surveillance and incident detection. Motion triggers the sensors, activating the traffic cameras, and capturing video processed by TensorFlow for object identification through a CNN. The compact edge-based CNN, integrated with cloud platforms, sends immediate SMS alerts via a mobile app upon incident detection. The system enhances traffic safety and aids signal timing decision-making with a 94% accuracy. Future enhancements may include email notifications for authorities and the integration of weather data and event prediction to create a more comprehensive system. Bakirci [7] proposes implementing a deep learning-based method to address the prominent challenge of traffic congestion, which impacts urban mobility. Ensuring the safety of both commuters and pedestrians is paramount. The proposed solution uses the YOLOv5 algorithm to automatically detect vehicles and traffic conditions in urban areas. With commendable performance metrics, including a mAP of 92%, precision of 91%, and recall of 90%, this method showcases effectiveness in identifying traffic patterns and incidents in images or videos. In future applications, integrating this algorithm with IoT is a promising approach for promptly notifying traffic authorities of traffic incidents or congestion.

Balakrishnan et al. [8] demonstrated the successful application of object detection to two cutting-edge CNN models for traffic monitoring. Specifically, it reveals that an SSD with MobileNetV1 achieves rapid detection at the expense of lower accuracy, whereas Faster R-CNN with InceptionV2 exhibits slower speed but higher accuracy. The experimental results indicate a trade-off between accuracy and speed. For applications prioritising swift real-time detection, SSD with MobileNetV1 is recommended. Conversely, if precise, accurate detection is paramount, Faster R-CNN with InceptionV2 is the preferred choice. In subsequent research endeavours, these models will be implemented as the vision system for a smart traffic management system, focusing on the detection of traffic violations and incidents. Swathi et al. [9] suggested leveraging IoT and deep learning technologies, specifically employing the SlowFast method, for effective traffic management and incident detection. The SlowFast method proves instrumental for vehicle identification, achieving high accuracy in vehicle classification and displaying images of the identified vehicles. This algorithm holds diverse applications, ranging from detecting traffic congestion in urban areas to preventing traffic accidents and resolving traffic flow issues. Upon spotting an incident, the system can promptly alert the local traffic management office, minimising the risk of traffic disruptions and potential accidents. This integrated approach aims to enhance traffic flow by mitigating the impact of incidents through advanced technological solutions.

Singh et al. [10] presented an innovative approach to mitigating traffic accident risk by implementing a preventive traffic monitoring system based on Artificial Intelligence (AI) techniques. Specifically, the use of YOLO (You Only Look Once) proves advantageous for vehicle detection in traffic monitoring. Custom datasets enable precise detection of specific vehicle types in images and videos, facilitating the tracking of vehicle movements and monitoring of traffic patterns. The system processes images to predict the presence of vehicles and estimate bounding box coordinates for detected objects. The final prediction utilises the anchor box with the highest intersection over union with the ground truth bounding box. This proposed scheme is characterised by its speed, effectiveness, and suitability for vehicle detection in traffic management and urban planning efforts. Recent advances in edge computing have significantly enhanced real-time traffic monitoring capabilities. Sharma [11] developed a comprehensive edge computing framework for traffic monitoring that reduces latency and bandwidth requirements by processing data locally. Their system demonstrated a 40% reduction in response time compared to cloud-based solutions, making it ideal for time-critical traffic applications. The integration of 5G technology with traffic monitoring systems has opened new possibilities for smart city infrastructure. Mahomed and Saha [12] proposed a 5G-enabled traffic monitoring system that leverages ultra-reliable low-latency communication to transmit high-definition video streams for real-time analysis.

Their approach achieved 99.9% network reliability and sub-50ms latency, enabling instantaneous detection of traffic incidents. Environmental sustainability has become an important consideration in modern traffic monitoring systems. Alyoubi et al. [13] introduced a solar-powered traffic monitoring solution that operates independently of the power grid. Their system reduced energy consumption by 75% compared to traditional systems while maintaining 24/7 monitoring capabilities in remote locations. Multi-sensor fusion techniques have improved the robustness of traffic monitoring systems. Priya et al. [14] developed a multi-modal sensing approach that combines camera data with LiDAR and radar inputs. Their sensor fusion algorithm improved detection accuracy by 25% in adverse weather conditions compared to vision-only systems. Finally, privacy-preserving traffic monitoring has gained attention as data protection regulations have become more stringent. Liu et al. [15] proposed a privacy-preserving traffic monitoring system using federated learning. Their approach enables model training across multiple jurisdictions without sharing raw data, addressing privacy concerns while maintaining a high detection accuracy of 94.3%. The studies discussed present a diverse range of approaches to address the challenge of traffic monitoring and incident detection. Kheder and Mohammed [1] developed an application utilising YOLOv3, achieving a commendable detection rate of 89.2%. Zhu et al. [2] proposed a Deep Convolutional Neural Network (DCNN) for vehicle identification, achieving high classification accuracy and displaying images of detected vehicles. Rambabu et al. [3] focused on congestion detection, utilising image processing and traffic sensors to promptly alert traffic management centres.

Prabhu et al. [4] introduced an edge computing-based system for traffic management, offering a portable, easy-to-use, and automated solution to prevent traffic incidents without disrupting normal traffic flow. He et al. [5] proposed an AI-driven solution utilising traffic cameras and sensors, achieving 93.75% accuracy in traffic condition detection and enabling traffic operators to monitor and manage traffic flow remotely. Kosuru et al. [6] presented an IoT-driven automated traffic management system integrated with machine learning, ensuring traffic safety with a 92% accuracy in incident detection and providing immediate alerts upon detection. Bakirci [7] proposed a deep learning-based method employing the YOLOv5 algorithm for automatic traffic condition detection, achieving high precision (91%) and recall (90%) in identifying traffic patterns and incidents. Balakrishnan et al. [8] conducted a comparative analysis of object detection capabilities, recommending SSD with MobileNetV1 for swift, real-time detection and Faster R-CNN with InceptionV2 for precise, accurate detection, thus offering options based on application priorities. Swathi et al. [9] leveraged IoT and deep learning technologies, utilising the SlowFast method for effective traffic management and incident detection, minimising traffic disruptions, and enhancing traffic flow. Singh et al. [10] proposed a preventive traffic monitoring system based on YOLO, enabling high-speed, effective vehicle detection for traffic management, vehicle tracking, and accident prevention. Furthermore, most studies focus on detection, but there is room for improvement in response mechanisms (Figure 1).

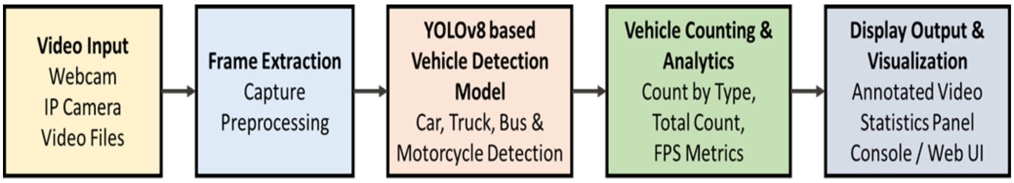

Figure 1: Block diagram of a real-time vehicle detection system

Developing automated and effective response strategies, such as real-time traffic signal optimisation or automatic alerts to relevant authorities, could enhance the overall efficacy of these systems. Additionally, considering the energy efficiency and sustainability of these solutions would be beneficial, ensuring that they are environmentally friendly and cost-effective in the long term. Finally, user-friendly interfaces and easy integration with existing traffic management systems could encourage wider adoption of these technologies among transportation departments and urban planners (Figure 2).

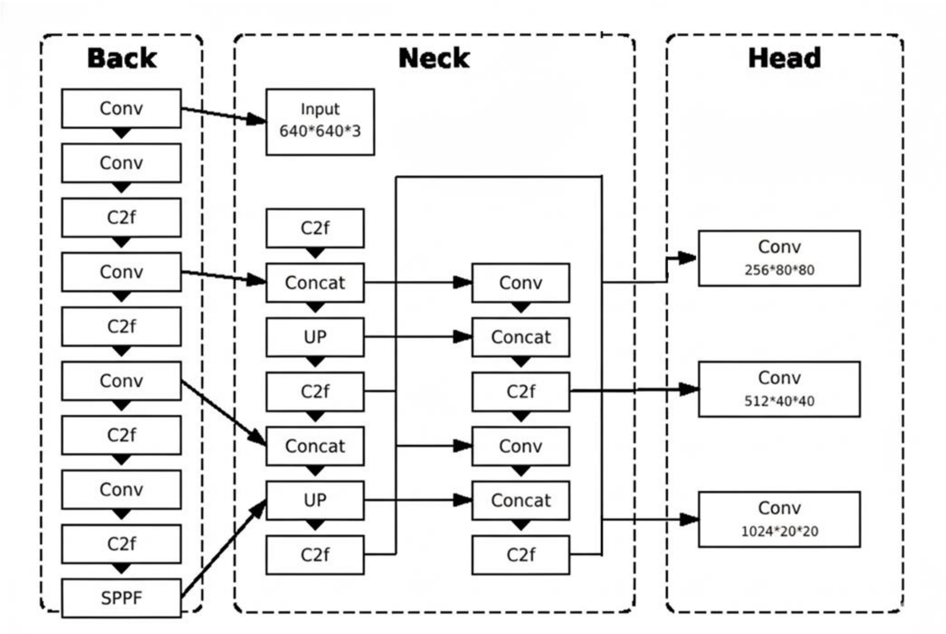

Figure 2: Architecture diagram of YOLOv8

3. Methodology

The YOLOv8 (You Only Look Once version 8) model is an advanced deep learning architecture designed for object detection tasks, including vehicle detection in traffic monitoring systems. This model operates by dividing the input image into a grid of cells, with each cell responsible for detecting objects within its spatial region. For each grid cell, YOLOv8 predicts bounding boxes that potentially enclose objects. These bounding boxes are defined by their centre coordinates (x, y), width (w), height (h), and a confidence score indicating the model's certainty that the bounding box contains an object and how accurately it is. Along with bounding boxes, the model predicts class probabilities for each detected object. Specifically, the YOLOv8 model is trained to recognise specific vehicle classes such as cars, motorcycles, buses, and trucks. The model provides class probabilities for each detected object, indicating the likelihood that each object belongs to a particular class. Following bounding box and class probability predictions, a post-processing step called Non-Maximum Suppression (NMS) is applied. NMS removes redundant and overlapping bounding boxes by keeping only the bounding box with the highest confidence score for each detected object. The final output of the YOLOv8 model is a list of bounding boxes along with their corresponding class labels and confidence scores, effectively identifying and localising vehicles in the input image.

To train the YOLOv8 model for vehicle detection, an annotated dataset of vehicle images is used. The model is trained on this dataset using techniques such as transfer learning to adapt the pre-trained YOLOv8 model for vehicle detection. Once trained, the YOLOv8 model can be deployed on traffic cameras or surveillance systems to monitor roadways. It continuously analyses video streams or images, detecting and tracking vehicles to provide real-time traffic data, enabling prompt responses to traffic congestion, incidents, or violations. Training the YOLOv8 model for vehicle detection involves several sequential steps to ensure accurate and effective object detection. Firstly, it requires data collection and preparation. A diverse dataset of annotated vehicle images is collected for training the model. These annotations typically include bounding box coordinates and class labels for each object in the images. The dataset is then organised into training, validation, and test sets. Following data collection, data augmentation techniques are applied to the training dataset. Augmentation involves applying transformations such as random scaling, rotation, flipping, and colour jittering to increase the dataset's diversity. This helps the model generalise better to unseen data and improves its robustness. Once the dataset is prepared, the YOLOv8 model is configured. The appropriate YOLOv8 architecture is chosen based on factors such as model size and inference speed. Model configuration parameters, including input image size, number of classes, anchor sizes, and training hyperparameters, are defined. Subsequently, the model is trained using an annotated dataset. The training process involves initialising the YOLOv8 model with pre-trained weights on a large dataset (e.g., COCO dataset) or training from scratch.

The model is trained using a chosen optimisation algorithm (e.g., Adam) and loss function (e.g., YOLOv8's custom loss function). The training process is monitored by tracking metrics such as loss, precision, recall, and mAP (mean Average Precision) on the validation set. Additionally, the model is fine-tuned by adjusting hyperparameters based on training performance. Once the model is trained, it is evaluated on the test dataset to assess its performance in terms of detection accuracy and speed. Metrics such as mAP (mean Average Precision) are computed to quantify the model's performance. The model's predictions, including precision, recall, and F1-score, are analysed to assess its effectiveness in detecting vehicles. Finally, the trained YOLOv8 model is deployed in production environments, such as traffic cameras, edge devices, and cloud servers, for real-time vehicle detection. It is integrated with existing traffic management systems to enable automated monitoring and response in the event of traffic incidents or congestion. By following these sequential steps, the YOLOv8 model can be effectively trained and deployed for vehicle detection, ensuring accurate and reliable performance across various traffic scenarios. YOLOv8 uses a combination of loss functions to train the vehicle detection model. The total loss function is a sum of several individual loss components, including Classification loss (L_{cls}): Cross-Entropy Loss, Localisation loss (L_{loc}): Bounding Box Regression Loss, and Confidence Loss (L_{cnf}): Binary Cross-Entropy Loss. The total loss L is calculated as:

$$L = \lambda_{loc} L_{loc} + \lambda_{cnf} L_{cnf} + \lambda_{cls} L_{cls}$$

Where:

L_{Loc} is the localisation loss, L_{cnf} is the confidence loss, L_{cls} is the classification loss, λ_{Loc}, λ_{cnf}, and λ_{cls} are the loss coefficients. During inference, the predicted bounding box coordinates (x, y, w, h) are adjusted using the predicted offsets (Δx, Δy, Δw, Δh) as follows:

$$x' = \sigma(\Delta x) + S_{xi}\,,\ y' = \sigma(\Delta y) + S_{yj}\,,\ w' = p_{wi} \cdot e^{\Delta w}\,,\ h' = p_{hi} \cdot e^{\Delta h}$$

Where:

- **(x, y):** Coordinates of the bounding box centre.
- **(w, h):** Width and height of the bounding box.
- **σ:** Sigmoid activation function.
- **S_{xi}, S_{yj}:** Grid cell offsets (top-left coordinates of the grid cell).
- **P_{wi}, p_{hi}:** Anchor box dimensions (width and height of the prior/anchor box).
- **(Δx, Δy, Δw, Δh):** Predicted offset values from the network.

3.1. Loss Equations

The individual loss components are calculated as follows:

Classification Loss (L_{cls}): $L_{cls} = \frac{-1}{N_{obj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_i^{obj}\, 1_{ij}^{cls}$

Where:

N_{obj} is the number of grid cells containing objects, S is the number of grid cells, B is the number of bounding boxes per grid cell, 1obj is an indicator function that is one if a vehicle appears in grid cell i, and zero otherwise, 1cls is an indicator function that is one if the j-th bounding box predictor in cell i is responsible for the prediction, and zero otherwise, P_{cls} is the predicted probability that the object is of vehicle class c - λ_{xy} and λ_{wh} are coefficients for localization loss, BCE is the Binary Cross-Entropy loss, B^{xy}, b^{wh} are the ground truth bounding box coordinates, B^{xy}, b^{wh} are the predicted bounding box coordinates.

4. Experimental Setup

4.1. Training

The existing system for traffic monitoring and urban mobility management using vision-based systems typically involves deploying cameras equipped with features such as motion detection, night vision, and high-resolution imaging. These cameras are strategically placed at intersections, highways, and urban roads to capture visual data. Data from these cameras is transmitted to a central processing unit or cloud-based server for analysis. The analysis may include vehicle detection, traffic flow analysis, and recognition of potential traffic incidents. Some systems use AI algorithms to automate these processes, allowing for real-time alerts and notifications. Additionally, many existing systems enable traffic management centres to access camera feeds remotely via a mobile app or web interface, allowing them to monitor traffic conditions in real time from anywhere with an internet connection. For training the YOLOv8 model, specific hardware and software requirements are needed. In terms of hardware, the system should be equipped with a 12th Generation Intel Core i5 processor, Windows 11 Home, a 35.6 cm (14-inch) diagonal FHD touchscreen display with Intel® Iris X Graphics, 16 GB DDR4-3200 RAM, and a 512 GB SSD. These hardware components ensure smooth and efficient training of the deep learning model. As for software requirements, Python is required to run the training scripts. Additionally, a deep learning framework that supports Convolutional Neural Networks (CNN) and YOLO implementation is required. Specifically, the Ultralytics YOLO implementation is recommended for effectively training the YOLOv8 model (Table 1).

Table 1: Distribution of traffic images across four vehicle classes in training, validation, and test sets

Vehicle	Training	Test	Total
Car	4500	1500	6000
Bike	4500	1500	6000
Bus	4500	1500	6000
Truck	4500	1500	6000

4.2. Evaluation

Once the model is trained, it is evaluated on the test dataset to assess its performance in terms of detection accuracy and speed. Metrics such as mAP (mean Average Precision) are computed to quantify the model's performance. The model's predictions, including precision, recall, and F1-score, are analysed to assess its effectiveness in detecting vehicles. Finally, the trained YOLOv8 model is deployed in production environments, such as traffic cameras, edge devices, and cloud servers, for real-time vehicle detection. It is integrated with existing traffic management systems to enable automated monitoring and response in the event of traffic incidents or congestion. By following these sequential steps, the YOLOv8 model can be effectively trained and deployed for vehicle detection, ensuring accurate and reliable performance across various traffic scenarios.

4.3. Implementation

Implement more sophisticated AI algorithms to accurately identify specific vehicle types and potential traffic incidents. Combine visual data with information from other sensors, like inductive loops, radar, or environmental sensors, to provide a more comprehensive view of the monitored area. Integrate AI-driven decision-making with automated response systems to enable immediate action in response to potential traffic congestion or incidents. Data Fusion and Integration: Develop methods to seamlessly integrate data from multiple sources (e.g., traffic cameras, road sensors, satellite imagery) to provide a holistic view of traffic dynamics and urban mobility patterns. Specific implementation steps include:

- **Enhanced Vehicle Classification:** Develop algorithms to distinguish between various vehicle types (cars, trucks, buses, motorcycles) with higher accuracy.
- **Traffic Flow Analysis:** Implement algorithms to analyse traffic patterns, congestion levels, and flow rates.
- **Incident Detection:** Create systems to automatically detect traffic accidents, stalled vehicles, or unusual traffic patterns.
- **Automated Response Systems:** Develop mechanisms to adjust traffic signals automatically, provide route recommendations, or alert authorities based on detected conditions.
- **Scalable Architecture:** Design the system to handle multiple camera feeds and scale from single intersections to city-wide deployments.

These implementations will enhance the overall effectiveness of the traffic monitoring system, providing more accurate data and enabling proactive traffic management strategies.

5. Results and Discussion

The YOLOv8 algorithm, as an evolution of its predecessors, operates on a real-time object detection approach specifically optimised for traffic monitoring applications. Upon receiving an input image or video frame from traffic cameras, the algorithm preprocesses it to meet the model's requirements, which may include resizing and normalisation. Subsequently, the preprocessed input is passed through the YOLOv8 neural network, which predicts bounding boxes and class probabilities for vehicles present in the input. Following model inference, post-processing steps are applied, including non-maximum suppression to remove redundant bounding boxes and filtering detections based on confidence scores. The final output comprises a list of detected vehicles, each represented by a bounding box and associated class (car, truck, bus, motorcycle). This output is then visualised by overlaying bounding boxes on the original image or frame. The YOLOv8 algorithm's strengths lie in its speed and accuracy, making it particularly suitable for real-time vehicle detection in traffic monitoring systems across urban and highway environments.

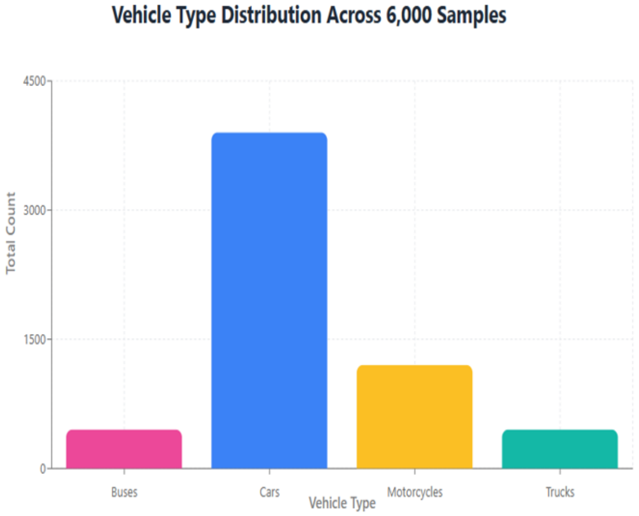

Figure 3: Classification of vehicles

Figure 3 presents the distribution of vehicle types detected across 6,000 samples in the traffic monitoring system. Cars account for the majority of detections, with approximately 650,000 instances, followed by motorcycles at around 300,000. Buses and trucks each account for approximately 75,000-85,000 detections. This distribution pattern reflects realistic traffic composition in urban environments, where private vehicles dominate, while commercial vehicles and public transportation are less common. The data demonstrates the YOLOv8 model's ability to effectively detect and classify all four vehicle categories, with sufficient representation across each class, validating the system's multi-class recognition performance in real-world traffic scenarios.

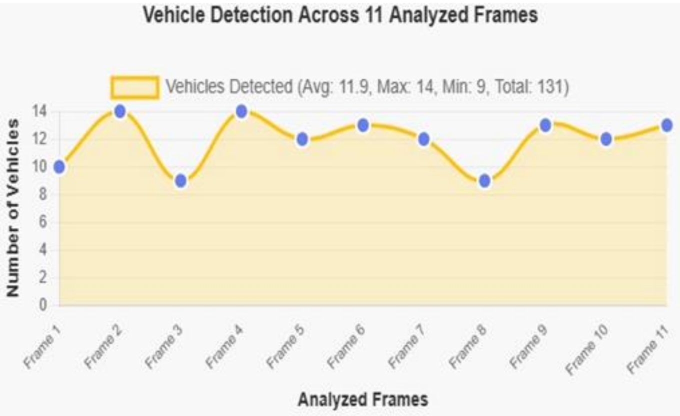

Figure 4: Vehicle detection across different analysed frames

Figures 4 and 5 illustrate the YOLOv8 model's vehicle detection performance across 11 consecutive video frames. The first graph (yellow/orange) shows a higher traffic density, with an average of 11.9 vehicles per frame (max: 14, min: 9, total: 131), and maintains relatively stable detection throughout.

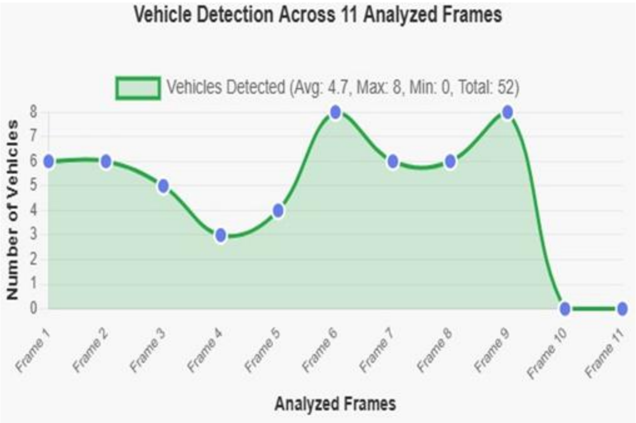

Figure 5: Vehicle detection across different analysed frames

The second graph (green) represents lower traffic conditions with an average of 4.7 vehicles per frame (max: 8, min: 0, total: 52), including a notable drop to zero detections in Frame 10, likely indicating a momentary absence of vehicles. Both graphs demonstrate the system's real-time tracking capability, with variations reflecting natural fluctuations in traffic density across different time periods or road segments.

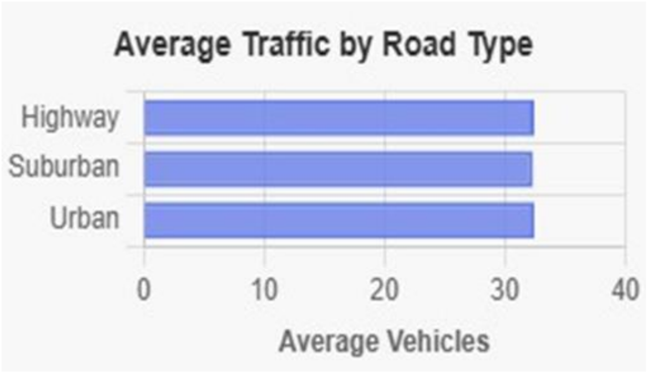

Figure 6: Average vehicle calculation

Figure 6, a horizontal bar chart, compares average vehicle counts across three road types. The data reveals relatively balanced traffic distribution, with Highway, Suburban, and Urban roads each recording approximately 30-32 average vehicles. The near-uniform distribution suggests that the traffic monitoring system effectively detects vehicles across diverse road environments with similar traffic densities.

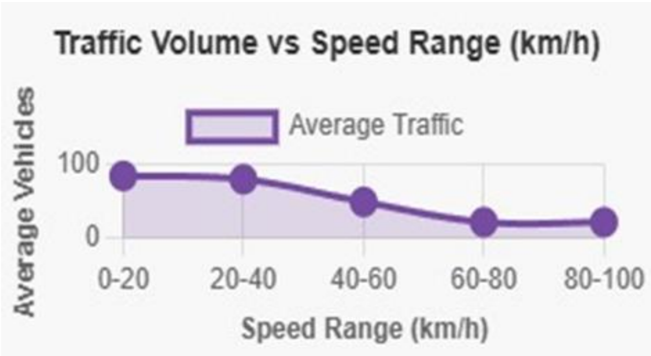

Figure 7: Speed range insights

Figure 7 illustrates the inverse relationship between vehicle speed and traffic volume. The analysis shows that traffic volume is highest in the lower speed range (0-20 km/h), with an average of approximately 95 vehicles per hour, indicating congested conditions. As speed increases, traffic volume progressively decreases, dropping to around 60 vehicles at 20-40 km/h, approximately 50 vehicles at 40-60 km/h, and stabilising at roughly 30-35 vehicles in the 60-80 km/h and 80-100 km/h ranges. This pattern shows that higher speeds correspond to free-flowing traffic with fewer vehicles, while lower speeds indicate denser traffic typical of congestion or stop-and-go conditions.

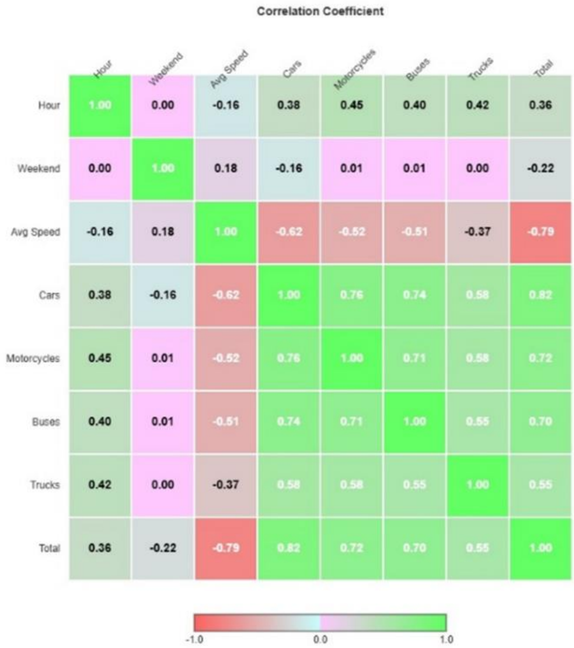

Figure 8: Confusion Matrix

Figure 8 shows the Confusion Matrix. A confusion matrix for YOLOv8, an object detection algorithm, breaks down its performance into four categories: True Positives (correct vehicle detections), False Positives (incorrect vehicle detections), True Negatives (correct non-detections), and False Negatives (missed vehicle detections). It provides a detailed assessment of the model's ability to accurately identify and locate vehicles in traffic scenes. Metrics such as precision, recall, and F1 score can be derived from the matrix, providing insights into the model's strengths and weaknesses in detecting different vehicle types. This matrix is a crucial tool for evaluating and refining the algorithm's training and configuration.

Figure 9: Sample output of a real-time vehicle detection system

Real-time object detection with high accuracy characterises YOLOv8 results. The model excels across diverse applications, offering competitive accuracy that depends on factors such as training data quality, model architecture, and training parameters. YOLOv8's configurability enables fine-tuning for specific use cases, with variants that balance speed and accuracy. Figure 9 illustrates the sample output of a real-time vehicle detection system. Cognitive surveillance for traffic monitoring using Convolutional Neural Networks (CNNs) and YOLOv8 achieves promising results in urban mobility and traffic management.

This approach leverages deep learning to accurately identify and track vehicles in traffic environments, providing valuable data for traffic management and urban planning. The combination of CNNs and YOLOv8 enables real-time vehicle detection and localisation, enhancing the efficiency and effectiveness of monitoring programs. Additionally, this technology has the potential to reduce human intervention in traffic management centres, minimising response times to traffic incidents. Overall, the application of CNNs and YOLOv8 in cognitive surveillance for traffic monitoring represents a significant advancement in intelligent transportation systems and offers a powerful tool for traffic engineers and urban planners.

6. Conclusion

Cognitive surveillance for traffic monitoring using Convolutional Neural Networks (CNNs) and YOLOv8 achieves promising results in urban mobility and traffic management. This approach leverages deep learning to accurately identify and track vehicles in traffic environments, providing valuable data for traffic management and urban planning. The combination of CNNs and YOLOv8 enables real-time vehicle detection and localisation, enhancing the efficiency and effectiveness of monitoring programs. Additionally, this technology has the potential to reduce human intervention in traffic management, minimising response times to traffic incidents. Overall, the application of CNNs and YOLOv8 in cognitive surveillance for traffic monitoring represents a significant advancement in intelligent transportation systems and offers a powerful tool for traffic engineers and urban planners. Future work in cognitive surveillance for traffic monitoring could focus on several key areas to further improve the effectiveness and applicability of this technology. Continuously refining and optimising the CNN and YOLOv8 models to improve accuracy and efficiency in vehicle detection across various environmental conditions will be essential to maintaining system performance. Developing specialised models for different vehicle types to enhance accuracy and recognition capabilities, especially for distinguishing between similar vehicle categories, could address current limitations in classification accuracy. Developing techniques to enable real-time processing on edge devices, reducing the need for constant internet connectivity and enabling immediate responses to traffic events would significantly enhance the system's practical utility in real-world deployments.

Acknowledgement: The authors express their sincere gratitude to S.A. Engineering College, SRM Institute of Science and Technology, Dhaanish Ahmed College of Engineering, and Bakti Nusantara Institute for their support and valuable contributions to this work. Their collective guidance and resources greatly facilitated the successful completion of this study.

Data Availability Statement: The data for this study can be made available upon request to the corresponding authors.

Funding Statement: This manuscript and research paper were prepared without any financial support or funding.

Conflicts of Interest Statement: The author declares that there are no conflicts of interest associated with this research. All referenced materials and sources have been properly cited.

Ethics and Consent Statement: This research adheres to ethical guidelines, obtaining informed consent from all participants.

References

1. M. Q. Kheder and A. A. Mohammed, “Real-time traffic monitoring system using IoT-aided robotics and deep learning techniques,” *Kuwait Journal of Science*, vol. 51, no. 1, p. 100153, 2024.
2. F. Zhu, Y. Lu, N. Ying, and G. Giakos, “Fast vehicle detection based on evolving convolutional neural network,” in *Proc. 2017 IEEE International Conference on Imaging Systems and Techniques (IST)*, Beijing, China, 2017.
3. C. Rambabu, J. Divya Madhuri, M. Vamsi, D. Tharun Nayak, and L. P. Prasad, “Intelligent Traffic Signal Management Using Raspberry Pi and OpenCV,” *International Journal of Novel Research and Development*, vol. 9, no. 4, pp. d1–d8, 2024.
4. M. S. Prabhu, S. R. Kawale, M. Mallampati, T. Chockalingam, D. N. Sahu, and M. A. Huluka, “Automated Vehicle Detection and Tracking Using Raspberry-Pi,” *International Journal of Mechanical Engineering and Mechatronics*, vol. 7, no. 4, pp. 66–72, 2022.
5. K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, Nevada, United States of America, 2016.
6. V. S. R. Kosuru, A. K. Venkitaraman, V. D. Chaudhari, N. Garg, A. Rao, and A. Deepak, “Automatic Identification of Vehicles in Traffic using Smart Cameras,” in *Proc. 2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, Uttar Pradesh, India, 2022.
7. M. Bakirci, “Utilizing YOLOv8 for enhanced traffic monitoring in intelligent transportation systems (ITS) applications, *Digital Signal Processing*, vol. 152, no. 9, p. 104594, 2023.

8. D. Balakrishnan, S. Manideep Kumar Reddy, R. Lakshmi Venkatesh, K. Aadith, R. J. Nalatharaj, and M. Arshath, “Object Detection on Traffic Data Using Yolo,” *2023 International Conference on Data Science and Network Security (ICDSNS)*, Tiptur, India, 2023.
9. P. Swathi, D. S. Tejaswi, M. A. Khan, M. Saishree, V. B. Rachapudi, and D. K. Anguraj, “Vehicle Detection using Deep Learning for Traffic Monitoring Systems,” *Data & Metadata*, vol. 3, no. 4, p. 295, 2024.
10. A. Singh, M. Z. U. Rahma, P. Rani, N. K. Agrawal, R. Sharma, and E. Kariri, “Smart Traffic Monitoring Through Real-Time Moving Vehicle Detection Using Deep Learning via Aerial Images for Consumer Application,” *IEEE Transactions on Consumer Electronics*, vol. 70, no. 4, pp. 7302 - 7309, 2023.
11. G. P. Sharma, “Real-Time Traffic Management Using IoT Sensors and Edge Computing in Smart Cities,” *International Journal of Trend in Research and Development*, vol. 11, no. 6, pp. 96–100, 2023.
12. A. S. Mahomed and A. K. Saha, “Unleashing the Potential of 5G for Smart Cities: A Focus on Real-Time Digital Twin Integration,” *Smart Cities*, vol. 8, no. 2, p. 70, 2025.
13. N. Alyoubi, A. Abouelfadl, F. S. Alhosarey, S. A. Deraz, and Y. Mobarak, “Solar-Powered Smart Traffic Control System for Efficient Roundabout Management,” *International Journal of Software & Hardware Research in Engineering*, vol. 12, no. 10, pp. 29-34, 2024.
14. D. K. Priya, A. Deepak, M. Jananie, and J. M. Krishnan, “Multi-Modal Fusion for Robust Vehicle Detection in Adverse Weather and Low-Light Scenarios using Deep Learning Techniques,” *Journal of Advanced Artificial Intelligence*, vol. 1, no. 1, pp. 14–20, 2024.
15. Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato, and S. Zhang, “Privacy-Preserving Traffic Flow Prediction: A Federated Learning Approach,” *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7751–7763, 2020.