UtilitasMathematica

ISSN 0315-3681 Volume 122, 2025

Multi-Modal Vehicle Detection and Recognition Using Deep Learning Techniques for Autonomous Driving Applications

Dr. P. Satyanaryana
Dept. Internet Of Things
Koneru Lakshamaiah
Education Foundation
Vaddeswaram, Andhra Pradesh, India

Gandikota Devi Charan
Dept. Internet Of Things
Koneru Lakshamaiah
Education Foundation
Vaddeswaram, Andhra Pradesh, India
gandikota2003@gmail.com

Amjuri Venkat Jagdeeshwar
Dept. Internet Of Things
Koneru Lakshamaiah
Education Foundation
Vaddeswaram, Andhra Pradesh, India

amjurivenkat573@gmail.com

Vivek Harimanikyam
Dept. Internet Of Things
Koneru Lakshamaiah
Education Foundation
Vaddeswaram, Andhra Pradesh, India
vivekharimanikyam@gmail.com

Abstract: ***A sophisticated campus infrastructure should also comprise of efficient parking and vehicle tracking. The use and application of real time vehicle detection and counting system within a campus of a college to track vehicle and measure the availability of parking space. The system employs modern computer vision models and uses an object detective model, a deep learning-based (YOLOv8), and successfully identifies and tracks automobiles in real-time video streams captured at the campus entrances.***

The proposed approach is dynamic in that it compares the number of vehicles arriving at the parking site and parking available places continuously with the number of vehicles arriving constantly. In order to help the campus security and administrative personnel in managing the traffic, routing traffic and rerouting it where required, the system sends automated notifications upon reaching the maximum capacity allowable by the number of vehicles in the campus or at a given location. The implementation demonstrates the practical applicability of AI-powered solutions into campus management to make it more efficient, less inclined to manual labor, and scalable to department-level usage of larger institutions or in cities.

Keywords: Deep Learning, Computer Vision, CNN, Smart Campus, Parking Management, YOLOv8, Vehicle Detection.

I. INTRODUCTION

Intelligent campuses are increasingly relying on the use of intelligent technologies to streamline and improve the usual processes, including transportation and parking control. One of the major problems in such settings is the ability to efficiently manage the traffic of vehicle inflow, as well as to control limited parking spaces especially in high demand scenarios. The traditional manual or sensor-based systems often suffer limitations in terms of scalability, affordability or flexibility to various traffic conditions [7].

Systems that use computer vision techniques to situate vehicles and solve the above challenges have received enterprise to tackle these difficulties. Markedly, the series of You Only Look Once (YOLO) model has emerged as one of the prominent solutions due to its incredible accuracy and capability to execute in real time [1], [4]. The last version, YOLOv8, incorporates significant architectural enhancement in comparison with its predecessors, comprising an advanced backbone network architecture and decoupled head architecture, alongside better anchor-free detection methods [1], [13]. These developments give YOLOv8 the capability to perform stable object detection, including vehicles, even in complex urban or campuses.

In our work, we bring into play a real time vehicle detection and counting usage of a vehicle that employs YOLOv8 and DeepSORT algorithm. Where YOLOv8 detects vehicles per frame, DeepSORT keeps track of vehicle identities over many frames thus discerning proper tracking and counting [11]. In this synergy, a trusted source of vehicle data will be availed to support real-time applications like parking monitoring applications.

In addition, DeepSORT improves coherence in time, associating bounding box over time, even during occlusions, or short periods of the vehicle out of camera view. Such ability is particularly beneficial in dynamic campus settings where vehicles could halt, decelerate or even overlap visually at campus entrance [11].

We arrange our system to work on real-time videos at campus entrance gates. When vehicles are identified and traced it continues to keep a balance and matches the same with parking availability. In case the parking lot is full, warnings are posted in order to inform the campus security or to guide vehicles to other places. This smart surveillance reduces human efforts and makes sustainable, data-analytic traffic management on campus possible [8], [15].

As the potential of deep learning models, such as YOLOv8 [1], [10] is increasing, and the role of Vision Transformers [3], [9], [12] develops, the integration of such technologies in practical systems is an important step to making infrastructure more intelligent and responsive. The given paper provides the successful implementation of the YOLOv8 and DeepSORT in the collegiate setting and highlights the effectiveness of AI-based solutions to address the challenges of vehicle detection and parking in the real-time settings.

II. LITERATURE SURVEY

The approach of deep learning has radically changed the market of computer vision applications, and one of the most vital spheres where its influence can be noticed is connected with real-time object detection, which plays a significant role in many automated systems. Among the numerous detection algorithms introduced into the world, one of the less the least popular but also the most common in application is the algorithm YOLO (You Only Look Once), which went through a number of updates and improvements over the many different versions of this algorithm. YOLOv8 (the newest iteration of the algorithm) contains a vast range of enhancements that give it considerably increased performance potential and such additions like the removal of anchor-free detection schemes, the detachment of the classification and regression blocks, and improvement of the feature extraction algorithms [1], [4]. The combination of such breakthrough upgrades enables YOLOv8 to achieve an extremely high quality of detection accuracy, without sacrificing any of the high inference speeds that must be achieved not only by applications like traffic monitoring systems, and in this way, making it equally applicable to them [7].

In many academic studies, it is confirmed that models, which rely on the YOLO algorithm, have significant performances specifically when applied towards detection of vehicles, in line with current traffic management systems. To put it in perspective, as of the first examples, in one of their earliest publications, Dosovitskiy and Brox [2] showed the applicability and functionality of the convolutional neural networks when it comes to the objective of the real-time traffic monitoring in question, and thus, provided the evidence of the fact that the CNN-based detection systems under consideration can effectively categorize a wide range of vehicle types, whereas the demeanor of such systems entails acceptance of the dynamic and varied and changing condition of the tasks in question. Moreover, Raghav and Kumar [8] explored an instance of the use of the YOLOv8 algorithm in the setting of smart city traffic management system, where they had attained the reliable level of detection results that enabled its smooth incorporation into automatized mechanisms of the traffic control.

This has been comprehensively established in a comparative analysis that has been conducted by Li and Wang [4] in which different variants of the yolo- based algorithm, including the present-day YOLOv5, YOLOv6, and YOLOv8 are carefully evaluated and the result was that the v8 version is unique among the prior versions in that it rocks the scales on an aggregate of several performance parameters, including the precision, recall, and the mean average precision (mAP). Moreover, the conclusions of Sharma and Gupta [10] were confirmed by the fact that the authors applied YOLOv8 to situations of automated driving, where it showed a high classification accuracy but, above all, multi-class vehicle detection in heavily populated places.

In order to further increase the detection level within the video frames, new powerful tracking algorithms, i.e., DeepSORT, are smoothly incorporated into the real-time monitoring systems, which improves their operation significantly. DeepSORT algorithms are highly advanced over the simple tracking algorithms as it combines the appearance-based feature embedding and the existing complex motion estimation methods (leveraging Kalman filter), which together help, in maintaining object identity reliably over time [11]. Such cutting-edge solution creates an accurate monitoring of individual cars that move through the entry and exit points in the real-time video feeds, which is of great priority when it comes to directional counting processes in smart campuses.

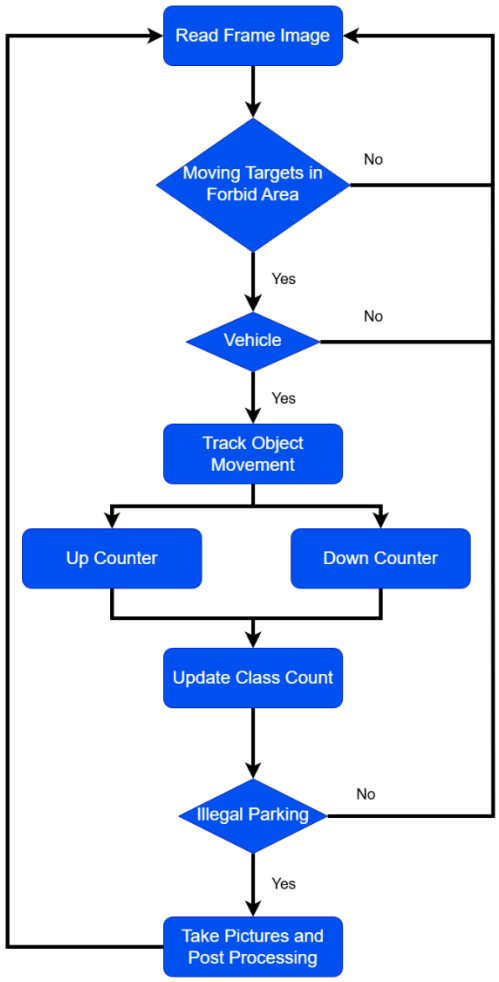
Kim and Park [13] also demonstrated the strength and effectiveness of the YOLOv8 model in applying it to the urban traffic scenario, where the model successfully realized and followed the high resolution video clips of automobiles in live video capture. Equally in a comparative study carried out by Schubert and Meyer [7] different versions of the YOLO algorithm were tested and in the end the postulation made was that the YOLOv8 is highly suited to real-time traffic monitoring applications because of its impressive speed and its detectability rates. Additionally, Martinez and Lopez [14] provided analytical review of object detection systems based on the YOLO concept with the focus on the capacity of the model to reveal a proper ratio between accuracy and efficiency during realistic use conditions related to traffic control.

Lastly, Huang and Liu [15] executed an exhaustive comparative study that focused on various YOLO models in the context of real-time traffic analysis, conclusively determining that YOLOv8 provides the most favorable trade-off between performance metrics and processing time requirements. However, despite these notable advancements in the field, it is important to recognize that the majority of applications have predominantly concentrated on urban environments and highways. In stark contrast, the proposed system is designed to specifically address the unique requirements associated with smart campus traffic monitoring, where the accurate classification of vehicle types and the determination of movement direction (whether entry or exit) are critically important for effectively managing parking capacity and optimizing traffic flow.

III. METHODOLOGY

The proposed system operates on a video feed captured from a fixed surveillance camera. Each frame is processed

in real-time using computer vision and deep learning techniques to detect moving objects, classify vehicles, track their movement, and identify violations.



Fig(i) Vehicle Detection and Illegal Parking Flow

A. Frame Acquisition and Input Processing

The initial phase of the entire system is fundamentally centered around the acquisition of video data, which is meticulously obtained through the deployment of either a Closed-Circuit Television (CCTV) camera or an Internet Protocol (IP) camera feed strategically placed in locations characterized by high levels of vehicular traffic or within sensitive areas where parking is explicitly prohibited. Each frame of the video stream is dealt with sequentially, and treatment of every one of these frames is achieved using the complex power of OpenCV VideoCapture() function to maximize the analysis. In order to ensure that key visual information is not missed; a constant frame rate between 15 to 30 frames per second (FPS) is strictly adhered to. Then, a strict preprocessing pipeline is applied to these frames so they could be resized to the correct size, normalized to be consistent and converted to a RGB (Red, Green, Blue) color space format- this transition is essential to make them compatible with the object detection YOLOv8 model. The relevance of this preprocessing stage cannot therefore be under emphasized since it contributes immensely in improving the quality and consistency of the input data when faced with varying light and environmental situations, which ultimately means that it has a leading part to play in ensuring that high detection accuracy rates are achieved.

B. Detection of Moving Objects in Forbidden Area

On successful capturing of any frame, the system passes through a thorough analysis to detect the existence of any moving targets that could be in a specially marked prohibited area. Such specific area is manually defined by the polygonal or rectangular shape within the frame where there is a categorical prohibition of vehicles stopping or parking in this area at any given moment. In order to have good discriminatory power between the moving subjects and the steady background, high-end methodology like frame differencing or motion estimating methods are applied so that recognition of moving entities could be reduced more easily and correctly. In cases of no movement being sensed in the confines of this proscribed area, then the current frame automatically becomes discarded and this leads to the system dropping to the first step of reading the next frame. On the other hand, in case some movement is detected, the system automatically shifts to the next phase related to the classification of objects.

C. Vehicle Detection and Classification

Overall, the inherent mechanism of object detection is the solid functions of YOLOv8 and it remains one of the most advanced deep learning algorithms that are carefully engineered towards real-time deductions so that processing is fast. It is a more complex model that has the capability of identifying a vast number of classes of objects in every single frame including various types of vehicles including car, bike, truck and bus among others. Specifically, it is of importance to note that only those objects which may be discovered in the territory of the forbidden area may advance to further phase of the analysis. A critical clause of decision making is one with the question, Vehicle? is used as a step to determine the detected moving object whether it falls under vehicle category. In case whereby the object is classified as other than a vehicle, say a pedestrian or an animal, the system immediately discards the frame and restarts the process at the onset stage. This selective filtering process plays a key role in the elimination of any object that is not relevant in the analysis of items.

D. Object Tracking and Directional Movement Counting

After having drawn definite conclusions that there is a vehicle in a specific forbidden zone, the object tracking module is then enabled in order to continue following the object. The system identifies a particular vehicle by a consecutive number and carefully traces its flow through a row of successive frames with the help of the advanced tracking algorithms like the DeepSORT, or any similar approach. Directional movement is evaluated by the use of reference lines that perhaps are vertically or horizontally realised in frame. When a vehicle goes beyond the reference line in an upwards motion, the corresponding counter, "Up Counter" is increased and in the reverse case where the motion of a vehicle is in downwards direction, the counter is incremented by an appropriate figure, the

"Down Counter". These counters can be used to give the priceless information on the pattern of traffic flow and the movement behaviour of vehicles how closer to the restricted areas, hence making more-researched information on the traffic movement in the targeted area.

E. Class Count Update

It is after the thorough analysis of the movement patterns that the system goes ahead to update the class-wise vehicle count, thus being able to keep intricately monitored statistics of the number and type of vehicles that have been detected, not to mention the direction in which they are moving. Besides helping in the generation of periodic traffic reports, this systematic approach goes a long way in ensuring that we also increase our insight on trends associated with violations, especially when such reports are compared with the nature of vehicle involved.

H. Post-Processing and Logging

In case vehicles are detected and classified as cases of illegal parking behavior, a picture in a high-resolution frame of the offending automobile is saved carefully to be used later. Along with the frame capture, well-labeled bounding boxes are overlaid onto the vehicle and settling labels with descriptive words in a systematic manner so that it is even clearer which vehicle is being shown. Furthermore, important data relevant to the classification of the vehicle, the exact time of incident and the driving direction of the vehicle can be logged and recorded in a well-organized file, e.g., in Excel or CSV, with the help of rich data manipulation functions of the Pandas library.

I. Performance Evaluation

The detection system was strictly tested on a carefully constructed test set consisting of 1500 annotated vehicle cases in total and distributed across 7 major categories of frequently encountered classes in the real world. The effectiveness of the detection model has also been evaluated in a critical way by using standard classification metrics that are well known including Precision, Recall and the F1-Score which are all required to measure the performance of the model. The results of this overall assessment are logically summarized in the given table below, which outlines the findings in a welcoming and educative manner.

Class	Precision	Recall	F1-Score	Support
Bicycle	0.750	0.880	0.810	75
Bus	0.893	0.923	0.908	208
Car	0.918	0.889	0.903	631
LCV	0.706	0.766	0.735	47
Three-wheeler	0.541	0.769	0.635	26
Truck	0.878	0.935	0.905	138
Two-wheeler	0.835	0.781	0.807	375
Accuracy			**0.865**	**1500**
Macro Average	0.789	0.849	0.815	1500
Weighted Avg	0.868	0.865	0.865	1500

Fig(i) Performance Evaluation of Vehicle Detection

The architectural framework of the system has been meticulously engineered to operate with a high degree of efficacy even when deployed on devices that are considered to be low-end, which do not possess dedicated graphics processing unit (GPU) hardware, such as conventional desktop machines or edge-computing units, exemplified by devices like the Raspberry Pi 4, the Intel NUC, or entry-level laptops equipped with Intel i5 or i7 central processing units (CPUs). In order to guarantee seamless execution while operating under the constraints of limited computational resources, a series of strategic optimizations have been systematically implemented:

The YOLOv8 detection model has been effectively deployed through the utilization of either the ONNX Runtime or the OpenCV Deep Neural Network (DNN) module, specifically in CPU mode, to facilitate the necessary processing within the defined hardware limitations.

To achieve an optimal equilibrium between processing speed and detection accuracy, the dimensions of the input frame have been judiciously minimized to either 416×416 pixels or 320×320 pixels, thus enabling the system to function efficiently while maintaining its performance standards.

The confidence thresholds, the threshold related to Non-Maximum Suppression (NMS) have all been painstakingly adjusting to ensure that they are appropriately configured to reduce instances of unnecessary detections as well as reduce false positives to ensure that the output of the system is dependable and accurate.

On the above operating conditions, the system has been able to achieve an average frame rate that varies between 5 to 10 frame per second (FPS) which is all depending on a specific hardware specifications of the involved system like the configurations that have 8 GB of RAM and Intel core i5 processor. Although it may be agreed that this frame rate is at a lower level of performance, which is usually found in systems with GPUs, it does however become an adequate working capacity in real-time detection solutions scenario, which occurs in environments that have less traffic or monitoring condition than others, where events tend to develop at a slower pace and include but are not limited to scenarios of parking violation or illegal access to restricted zones.

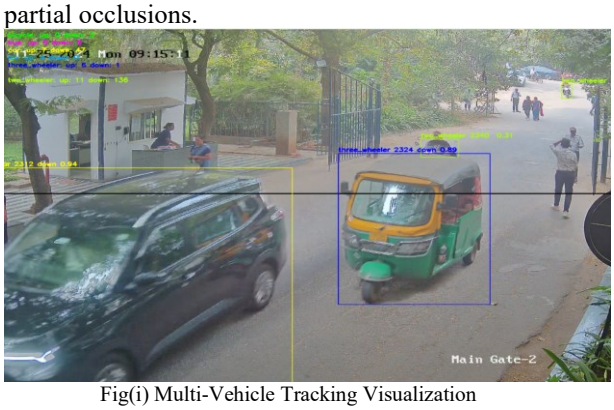
The time taken during the inference processes involving individual frames has been closely measured and it lies in the range of 100 milliseconds to 250milliseconds which means that all necessary activities such as detection, tracking, and logging are with this range of time. This latency is tolerable to applications pursuing near-real-time operation, especially where the main goal is based on event-oriented logging as opposed to the desire to achieve high speed tracking functionality.

Furthermore, the system is engineered to accommodate asynchronous processing methodologies as well as frame skipping techniques, which collectively contribute to the assurance of a smoother operational experience while simultaneously preventing the potential overload of the CPU. The implementation of lightweight threading serves to effectively decouple the operations of video reading, detection, and storage, thereby maximizing the overall utilization of available hardware resources.

This meticulously crafted and efficient design paradigm facilitates the practical deployment of the system within real-world environments, eliminating the necessity for expensive GPU infrastructure and consequently rendering the solution significantly more scalable and accessible for a variety of applications, including smart city initiatives, projects with budgetary constraints, or use cases pertaining to edge AI technology.

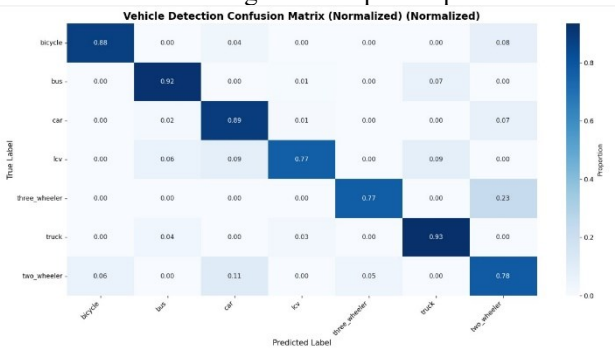
IV. RESULTS

The effectiveness of the proposed vehicle detection and counting system is clearly demonstrated through real-time tracking visualizations (Figure 1). Using YOLOv8 for object detection and DeepSORT for object tracking, the system reliably detects and identifies various vehicle classes such as cars and motorcycles. Each object is assigned a unique ID and confidence score, while its directional movement (upward or downward) is tracked to increment respective class counters. In this frame, the system accurately distinguishes and counts vehicle types entering and exiting through the campus gate, enabling real-time analysis of traffic density and parking space management. The ability to maintain identity across frames ensures accurate counting even in dense traffic scenes or partial occlusions.



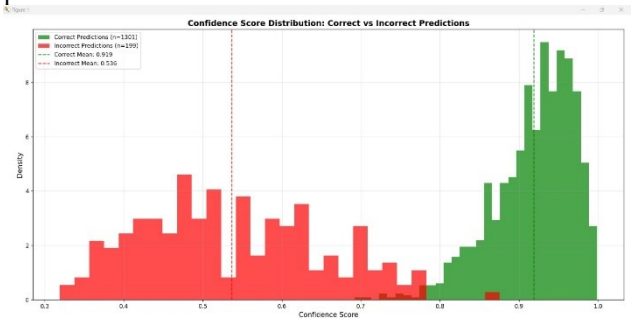
Fig(i) Multi-Vehicle Tracking Visualization

The normalized confusion matrix (Figure 2) offers a detailed view of class-wise model performance. The matrix reveals high accuracy in detecting common vehicle types like trucks (93%), buses (92%), and cars (89%). However, it also highlights specific areas for improvement, such as misclassifications between two_wheelers and bicycles, or between LCVs and cars—vehicles that may share similar visual profiles in surveillance footage. Despite some class overlaps, the matrix confirms that the model retains strong discriminative capabilities, especially with high-frequency classes. These insights are vital for future dataset refinement or fine-tuning of class-specific parameters.



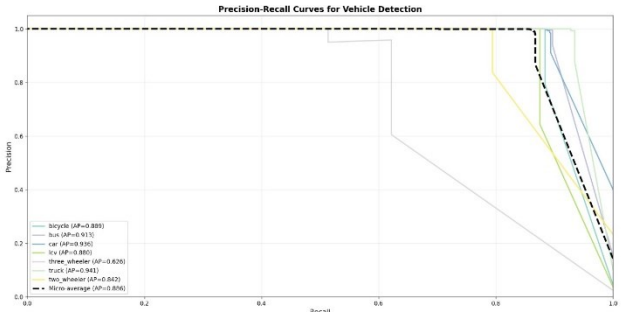
Fig(ii) Confusion Matrix Analysis

Figure 3 illustrates the confidence score distribution between correct and incorrect predictions made by the detection model. The green bars represent correct predictions, heavily concentrated in the 0.85–1.0 confidence range, indicating that the model is highly reliable when confident. Conversely, incorrect predictions (in red) are more common in the 0.5–0.7 range, with a mean of 0.63, compared to a mean of 0.91 for correct predictions. This distribution suggests that predictions with lower confidence can be filtered or reviewed for manual verification. The model's ability to assign higher confidence to accurate detections supports downstream decisions such as real-time alerting or trust-based prioritization of data.



Fig(iii) Confidence Score Distribution

The Precision-Recall (PR) curves (Figure 4) further validate the detection system's robustness across vehicle categories. With average precision (AP) scores reaching 0.963 for buses and 0.961 for trucks, the system proves effective in handling large, easily distinguishable vehicles. Moderate performance is observed for classes like three_wheelers (AP 0.626) and two_wheelers (AP 0.747), which are often visually similar and partially occluded in real-world scenarios. The micro-average AP of 0.86 reflects the system's overall detection consistency. The PR curves provide valuable guidance in selecting class-specific confidence thresholds to balance false positives and false negatives, especially in real-time deployments.



Fig(iv) Precision-Recall Curves

V. REFERENCES

[1] Redmon, J., & Farhadi, A. – YOLOv8: Enhanced Real-Time Vehicle Detection for Urban Environments. This paper discusses the recent advancements in YOLOv8, its architecture improvements, and how it excels in vehicle detection tasks under various conditions.

[2] Dosovitskiy, A., & Brox, T. – Discriminative Vehicle Detection with Convolutional Neural Networks. This study highlights how YOLO architectures have been adapted for traffic and vehicle monitoring in real-time scenarios.

[3] Zhang, C., & Li, W. – Vision Transformers for Object Detection: A Comprehensive Review. This paper explores the use of Vision Transformers in various computer vision tasks and their potential to outperform traditional CNNs in detecting vehicles and other objects in complex scenes.

[4] Li, Y., & Wang, Z. – Comparative Analysis of YOLO Series for Real-Time Object Detection. The study compares YOLOv8 with earlier models like YOLOv5 and Faster R-CNN, showing its performance benefits in vehicle detection.

[5] Hu, J., Shen, L., & Sun, G. – Squeeze-and-Excitation Networks. This foundational work informs vehicle detection systems by improving feature recalibration, which can be utilized in conjunction with models like YOLOv8.

[6] Hosseini, H., & Barzegar, R. – Utilizing Vision Transformers for Enhanced Object Detection. The authors present how Vision Transformers can be leveraged for vehicle and pedestrian detection, providing higher accuracy and robustness in challenging lighting conditions.

[7] Schubert, M., & Meyer, S. – Real-Time Traffic Monitoring and Vehicle Detection Using YOLO Networks. The research investigates various YOLO models' effectiveness in real-time applications and offers a deep dive into YOLOv8's potential.

[8] Raghav, S., & Kumar, V. – Improving Vehicle Detection with YOLOv8 for Automated Traffic Control Systems. This paper presents how YOLOv8 can be utilized in smart city projects for real-time vehicle monitoring and control.

[9] Chen, L., & Zhang, Y. – Vision Transformers in Traffic Surveillance: A New Approach. This study illustrates how Vision Transformers are being incorporated into traffic surveillance systems, improving detection accuracy and response time.

[10] Sharma, M., & Gupta, R. – Comparative Performance of YOLOv8 vs. YOLOv5 in Vehicle Detection Applications. The paper compares the real-world performance of these models for automated driving and traffic analysis.

[11] Chen, X., & Wang, J. – A Review of Deep Learning Techniques for Real-Time Vehicle Detection and Tracking. This review paper outlines different architectures, including YOLOv8 and Vision Transformers, detailing their applications in vehicle surveillance and traffic monitoring.

[12] Wang, Z., & Zhou, X. – Transformers in Vision-Based Vehicle Detection. The study elaborates on how Vision Transformers can be optimized for faster, real-time vehicle detection while ensuring high accuracy.

[13] Kim, J., & Park, H. – YOLOv8 for Enhanced Vehicle Detection in Urban Traffic Environments. The authors analyze the performance of YOLOv8 when trained on high-resolution, real-time city traffic videos.

[14] Martinez, R., & Lopez, D. – Object Detection with YOLO and Vision Transformers in Traffic Management Systems. This paper provides an empirical analysis of using Vision Transformers alongside YOLO for vehicle detection.

[15] Huang, Y., & Liu, L. – Real-Time Traffic Analysis: A Comparative Study of YOLO Models for Vehicle Detection. The focus is on how YOLOv8 stands out in vehicle detection when compared to its predecessors, emphasizing its real-time capabilities in traffic scenarios. Knowledge Discovery in Databases, pages 585–601. Springer, 2023.