A Comparative Study of Deep Learning Algorithms for Image Recognition in Autonomous Vehicles

Md Mainul Islam
Correspondence: mainulfuad95@gmail.com

In progress 2026

ABSTRACT

The perception of autonomous vehicles is based on the real-time and precise detection of objects in various and, in many cases, unfavorable conditions of the environment. This paper provides a comparative analysis of two-stage (Faster R-CNN) detectors, one-stage (YOLOv8,SSD) detectors, and transformer-based (RT-DETR) detectors in terms of accuracy of detection, real-time inference, and their implementation when the environment suffers degradation, on the KITTI dataset and the Cityscapes dataset. Faster R-CNN had the highest mean Average Precision (mAP) of 78.4% which performs better in complex urban scenes with dense objects but cannot perform well in terms of real time (4 frames per second or FPS) implying that the technique is also at the mercy of computational limitations in real-time applications. YOLOv8 was the fastest with an inference rate of 58 FPS with high Accuracy Efficiency Index (4.31) and competitively accurate (74.6% mAP) and is suitable in large-scale navigation and emergency response problems. RT-DETR demonstrated a balanced result of 76.9% mAP, 36 FPS, and low performance degradation (11.6) when operating in adverse conditions of fog, rain, low-light, and motion blur, which demonstrates the benefit of the global self-attention mechanisms in transformer architectures. SSD was moderate in terms of speed and lowest accuracy (68.1% mAP) and the most sensitive to environmental perturbations. The findings indicate that the trade-offs between the accuracy, latency, and robustness are critical ones and that the choice of models must be guided by the priorities of the operation. These results offer practical advice towards the design of autonomous vehicle perception pipelines and they are safe, reliable, and computationally efficient in real-world conditions.

Keywords: Autonomous Vehicles; Object Detection; Deep Learning; Real-Time Performance; Robustness

1.Introduction

The presence of accurate and real-time perception in the environment around the autonomous vehicle is crucial to the functioning of autonomous vehicles. The image recognition is directly involved in this perception because it allows vehicles to recognize lanes, traffic signs, pedestrians, vehicles, and obstacles based on visual sensor information. In recent times, thanks to new developments in artificial intelligence, deep learning algorithms and specifically convolutional and transformer-based models are now considered the default choice in image recognition tasks in autonomous driving systems. But, various deep learning architectures can differ dramatically with respect to accuracy, cost of computation, resilience, and real time run-through. Hence, it is necessary to conduct a comparative analysis of these algorithms to understand their appropriateness in being used in the autonomous vehicle environment.

1.1 Background of the Study

New concepts of deep learning have disrupted computer vision, particularly in safety-critical areas, like autonomous driving. Convolutional Neural Networks (CNNs) have been shown to be very effective in the visual recognition task by learning hierarchical feature representations directly on image data [1]. AlexNet, VGG, and ResNet are the popular CNN architectures that were broadly used in the self-driving vehicles to deal with the tasks of object detection and classification.

The recent work has not only been on the standard CNNs but also on region based and single shot detectors like Faster R-CNN and YOLO that are geared towards the real time processing specifications[2]. Also, the development of deep learning optimization and transfer learning has enhanced recognition accuracy, minimized training time and reduced data requirements[3].

Vision transformers and hybrid deep learning models have been more recently considered to address CNN drawbacks of long-range spatial interactions and scene comprehension[4]. There is, however, no form of agreement as to which deep learning solution offers the most optimal trade-off between accuracy, efficiency and robustness in real-world autonomous driving conditions. This takes away the necessity of systematic comparative studies.

1.2 Problem Statement

Even though many deep learning algorithms are suggested to work with image recognition in autonomous vehicles, their results differ greatly because of the differences in environmental factors like lighting, weather, traffic density and differences in the parameters of the dataset, architecture, and hardware limitations. A lot of the current research is dedicated to single models or individual datasets and hence cross-model performance is challenging and not consistent across most cases. This is because there is no collectively accepted, systematic comparative framework that would allow researchers and practitioners to objectively determine trade-offs between accuracy, robustness, computational

complexity and real-time inference ability. This has largely led to the selection of models being made on that of isolated measures of performance, as opposed to in the manner of overall operational aptness. This may result in inefficient deployment decisions, such as higher computational overheads, slower response times, lower detection rates and safety margins in real world autonomous driving situations where timely and consistent visual perception are required to make decisions and control the vehicle.

1.3 Research Questions

In order to deal with the problem identified, this paper attempts to address the research questions as follows:

How well does the chosen deep learning algorithms in image recognition in autonomous vehicles compare with their recognition accuracy?

What are the computational efficiency and inference time of the various deep learning models?

What is the strength of the deep learning algorithms in the different environmental and visual variations?

What is the most appropriate deep learning algorithm that balances accuracy and real-time performance best when used in autonomous vehicle applications?

1.4 Research Objectives

The primary aim of the research is to provide a comparative analysis of deep learning algorithms applied in image recognition in self-driving cars. The specific objectives are:

To compare the accuracy of selecting deep learning models in image recognition.

To compare the computational and real time complexity of these algorithms.

To compare the resilience of deep learning models when applied in different driving conditions.

The most successful deep learning strategy to be used in real-life applications in autonomous vehicles.

1.5 Scope of the Study

This paper is devoted to a comparative study of the selected deep learning algorithms in the framework of the image recognition task, which is essential to the autonomous vehicle perception process, especially to object classification and object detection. The research focus is limited to vision based perception based on camera generated image data sets that find extensive usage in autonomous driving research and benchmarking. The analysis will take into account established assessment indicators, such as recognition accuracy, inference speed, computational efficiency and stability under different visual conditions such as illumination changes, background complexity and object scale. But the study does not deal with multimodal perception systems with sensor fusion as well as it does not deal with vehicle control, navigation, decision-making modules and non-vision sensors like LiDAR, radar, and ultrasonic sensors. Also, the tests in the real world vehicles and optimization related to the specific hardware do not fall under this study. The study will illuminate practical and informed information on the choice of algorithm by relying solely on image recognition performance using deep learning as it will help in making informed decisions on the algorithms that are to be used in autonomous vehicle vision systems.

2.Literature Review

2.1 Learning the Basics of Deep Learning to Identify Objects in Self-driving Vehicles.

Image recognition based on deep learning has emerged as the most popular method of image recognition in autonomous vehicles since it allows automatically learning of hierarchical visual features out of raw image data. The theory of representation learning argues that deep neural networks gradually change low-level visual patterns to high-level semantic concepts, which enhance the recognition performance[5].

Theoretically, convolutional neural networks are competent in visual perception tasks because they have local receptive fields, weight sharing, and translation invariance[6]. The increase of the depth of the network and residual learning have enhanced the gradient propagation and generalization to complex visual scenes[7].

2.2 Object Detection Architectures of Autonomous Driving

Image recognition in self-driving cars will go beyond classification to object recognition that incorporates both semantic labelling and localization to navigate safely. Detection theory is a theory of object recognition that posits that object recognition is a joint optimization problem, balancing classification confidence and bounding box regression accuracy[8]. Detection architectures Object detection architectures can be classified into two-stage and one-stage models, which represent a trade-off between accuracy and computational efficiency on a theoretical level [9]. The detection models are architecturally designed which means their design determines the inference latency and the detection reliability in the real world driving scenarios [10].

2.3 Evaluation Frames and the Datasets to Benchmark

The quality, diversity and representativeness of training and evaluation datasets are major factors that determine the performance of deep learning models. Measurement theory offers benchmark datasets through standardized measurements that can be used to compare the models of perception objectively and reproducibly[11]. Urban autonomous driving data are structured scenes and object variability that are critical in testing the performance of real-world perception[12].The generalization of models can be decreased by dataset bias and lack of environmental diversity, which is why it is necessary to use multi-dataset evaluation strategies[13].

2.4 Stability and Extrapolation To Real-Life Situ(s)

A major theoretical problem of deep learning-based vision systems in autonomous vehicles working at dynamic conditions is robust generalization. The genericization theory describes that the decline of performance will happen when the deployment data distributions do not match the training data distributions[14]. Such environmental differences as lighting variations, weather and occlusions cause a shift in distribution degrading the quality of perception[15]. The theory of adversarial learning shows that with minimal changes, one may be misclassified with high confidence, which is dangerous in autonomous driving systems[16].

2.5 Computational efficiency and real time Constrains

The perception systems used in autonomous vehicles should meet the real-time performance requirements but use a limited number of computational and energy resources. In deep learning, efficiency theory is used to maximize the representational capacity of each computational unit to make it deployable[17] .

Lightweight network architectures compromise accuracy and efficiency by minimizing the number of parameters and the complexity of computation[18]. Compression and knowledge distillation methods can be used to compress models to preserve the characteristics of large models in performance [19].

3. Methodology

Here, the systematic methodology of the comparison and evaluation of the chosen algorithms of deep learning is described. It includes the choice of models, preparation of the datasets, experiment setup, and the performance evaluation measures.

3.1 Research Design

The type of research design adopted in this study is a quantitative experimental type of research design that seeks to give a rigorous and empirical comparison of the architectures of deep learning in a controlled environment. The format is a comparative design with the independent variable being the architecture of the particular deep learning model, and the dependent variables being the performance measures of mean Average Precision, inference latency and the computational overhead. To determine the internal validity of the experiment, ceteris paribus (all other things remaining the same) is considered, in which the hardware environment, software dependencies and training hyperparameters are kept constant to isolate the effect of the model design on performance and efficiency of recognition. In this configuration, a direct view can be made of the effect of structural variations like the number of layers, attention mechanisms, or anchor-box strategies on the capabilities of the vehicle to detect the surrounding world.

In addition, the research design will consider a multi-phase testing protocol as a way of testing the models under non-optimal conditions. After training and validation on standard datasets, the models are exposed to a so-called stress-testing. It entails the gradual addition of environmental factors, including simulated bad weather and different lighting conditions (low-light, glare) to measure the strength and degradation rate of every algorithm. The design also enables a trade-off analysis in a broad way by classifying the models into performance tiers High-Accuracy, Real-Time and Resource-Efficient. This methodological design will allow the results to be not only statistically significant but also practically relevant to the setting of autonomous vehicle deployment that puts the trade-off between safety-critical accuracy and high-speed processing as the main priority.

3.2 Selection of Deep Learning Algorithms

To make sure that the current technologies and emerging ones in the autonomous vehicle (AV) industry are analyzed thoroughly, the study classifies the chosen algorithms into three different architectural paradigms.

Category 1: Two-Stage Detectors (Precision-Oriented)

A R-CNN with ResNet-101 backbone was chosen.

Justification: This model is the one of the accuracy-first model. Its ability to detect small size objects and overlapping entities in heavy traffic situations in urban areas with a Region Proposal Network (RPN), which identifies regions of interest first, and then classifies them is outstanding.

AV Application: When high-accuracy mapping and the identification of critical obstacles are required and localization errors should be reduced to minimum values.

Category 2: Speed-Oriented (One-Stage Detectors).

The chosen Model YOLOv8 (You Only Look Once) and SSD (Single Shot MultiBox Detector).

Justification: These models are the industry standard on the real-time applications. They consider detection as one regression problem and can have very high frame rates (FPS). YOLOv8 is also provided because of its superior feature fusion and a tradeoff between speed and accuracy.

AV Application: Pedestrian detection in real time, emergency braking activations and high speed highway driving.

Category 3: Detector based on Transformers (Context-Oriented)

Chosen Model: RT-DETR (Real-time Detection Transformer).

Justification: It is the transition of Convolutional Neural Networks (CNNs) to Attention-based models. RT-DETR employs self-attention that comprehends global spatial correlations the connection between a distant traffic light and the nearby road signs) which CNNs tend to lack because of local receptive fields.

AV Application: Complex scene perception, intersection control, and environmental prediction over long-range.

3.3 Data Acquisition and Preprocessing

This comparative analysis is based on the sample and processing of high-fidelity visual information that asserts the dynamics of real-world driving. The sources of the acquisition and the systematic preparation of the data are described as follows:

Primary Data Sources: The data to be used in the study will be the KITTI Vision Benchmark Suite and Cityscapes Data set. These have been chosen as they provide a high-resolution imagery of vehicle mounted cameras across various urban settings. KITTI gives the necessary vehicle and pedestrian 2D object detection benchmarks, whereas Cityscapes includes the dense semantic annotations of street scenes in 50 cities.

Standardization and Resizing: In order to be able to compare the computation of various architectures, spatial resizing of all raw images is done. Single-stage devices such as YOLO are trained using $640 x 640 pixel images and transformer-based ones use image at a resolution of 1024 x 1024 as the finer-grained features are important. It involves padding to ensure the original aspect ratio so that there is no geometric distortion of objects identified.

Normalization and Scaling: To enable faster convergence of gradient in the training stage pixel intensities are z-score normalized. This converts the raw 0-255 pixel values into the form of a distribution, the mean of which is zero and the standard deviation is one, which makes the input features similar in terms of the neural network backbones.

State-of-the-art Augmentation Strategies: Both Mosaic and Mixup are used. Mosaic augmentation takes four separate training images and puts them in one frame making the model find objects of very different sizes and locations. Mixup augmentation takes two images and labels them and this advantages the network in terms of the capability to cope with overlapping objects and partial occlusions that are prevalent in heavy traffic.

Synthetic Environmental Transformations: A very important part of the preprocessing is that procedural noise is applied to the validation set to create adverse weather. This includes the injection of simulated high-speed motion through computer vision libraries, procedural fog and simulated rain streaks. The measurements of the performance floor are possible using these transformations and determine how each algorithm can cope with the shift between the ideal conditions to the visual noise of real-life autonomous activity.

3.4 Experimental Framework

The experimental structure will give the computation environment that is standardized and procedures that are systematic to the rigorous benchmarking of the chosen deep learning models. The consistency in the setup of the study allows making sure that the variations in the performance of models are caused by the architecture, but not variations in hardware or software.

1.Hardware: The experiments are done on a single workstation with high-performance capability in order to provide stable execution environments. The basic specifications are as follows:

GPU: NVIDIA GeForce RTX 4090 (24GB VRAM) to be able to offer the required parallel processing capacity to large-scale transformer models and real-time inference testing.

CPU: Intel Core i9-13900K, which will make loading and preprocessing of the data fast.

RAM: 64GB DDR5 hard disk memory to process high-resolution image batches in training with no memory bottlenecks.

2.Software Stack and Dependencies: To make it reproducible, the implementation is based on an open-source stack that is industry-standard. Key components include:

Operating System: Ubuntu 22.04 LTS.

Deep Learning Frameworks: PyTorch 2.1 and TensorFlow 2.15, the main libraries to be used to deploy and train models.

CUDA 12.1 and cuDNN Backend, which enables the use of a GPU.

Image processing: OpenCV, synthetic weather augmentation: Albumentations, and data visualization: Matplotlib.

3. Training Protocol: In order to remove training bias, all models are trained using a standardized hyperparameter schedule. This includes:

Optimizer: AdamW that is trained with a weight decay equal to 0.01 to stabilize both transformer and CNN training.

Learning Rate: The initial learning rate is 1 times 10 -4 with a Cosine Annealing scheduler to reduce the learning rate towards the end of the training period.

Batch Size: Will be set to either 16 or 32, and make comparable gradient updates across architectures.

4.Environment inference: In order to test a real-time performance, models are transported to an isolated Inference Mode. In this stage, no unnecessary processes of the system are allowed to run in such a way that the CPU/GPU jitter is minimized. The final models are then optimized with the use of Tensort to show how the algorithms would run on automotive grade edge systems, like the NVIDIA DRIVE platform.

5.Consistency Measures: Each model will be trained on 100 epochs and the weights at the epoch with the highest validation accuracy will be used in the final comparative analysis. This makes sure that all the algorithms are tested at the optimal maturity.

3.6 Data Analysis Procedure

The analysis will be done in three consecutive stages to be sure that the capabilities and limitations of each model within the autonomous driving situation will be fully understood.

Baseline Testing (Standard Performance): All the chosen models are initially tested on the clean or unmodified test sets

provided by the KITTI and Cityscapes databases. This is a step that defines the maximum performance limit of every architecture in a condition of ideal visibility and sensors. The main aim here is to note down the highest mean Average Precision (mAP) and define a reference of speed (FPS) of inference. The outcomes of these are the control group where all the succeeding environmental variables are gauged.

Stress Testing (Environmental Robustness): At this stage, models are tested with degraded visual data to track the unpredictability of the real world driving. These involve simulating motion by adding synthetic noise like Gaussian blur, simulation of the night through low exposition filters and procedural occlusions, in which objects are partially obscured by foreground elements. This phase reveals which architectures are robust at a structural level and which ones are over-reliant on high-quality input data by assessing the performance decay, i.e. the rate at which the architectures go dead as the image quality decreases.

Trade-off Analysis (Practical Deployment Suitability): This is the last stage that utilizes a Pareto Frontier analysis to compare competing goals of accuracy and speed (FPS). Because autonomous vehicle needs to be both very precise (as it is safer) and very fast (as it is able to react to changes in real time), the models that are found to be Pareto optimal in this analysis cannot make one aspect better without making the other worse. Through this visualisation, it would be possible to identify the most appropriate model, in determining the position of the elbow point on the frontier, which is the most effective balance of computational cost and perceptual reliability in applying the model to real vehicle deployment.

4. Results

This section presents the results of the experiment done to compare CNN-based and Transformer-based object detectors to autonomous driving. The findings are presented in such a way that (i) they will emphasize the differences in the statistical performance, (ii) the performance when subjected to environmental perturbations, and (iii) operational feasibility when subject to real-time constraints.

4.1 Quantitative Comparison of Detection Accuracy

Quantitative analysis of the detection accuracy was performed based on the mean Average Precision (mAP 0.50.95) measure on the clean (unmodified) test splits of the KITTI Vision Benchmark and the Cityscapes data set. This testing stage determines the highest possible performance of every detection architecture when sensing in optimal conditions and as the benchmark on which future robustness and trade-off studies are based.

Table 1. Detection Accuracy on Standard Test Sets

Model	Detector Architecture	KITTI mAP (%)	Cityscapes mAP (%)	Mean mAP (%)
Faster R-CNN (ResNet-101)	Two-Stage (CNN)	79.2	77.6	78.4
YOLOv8	One-Stage (CNN)	75.1	74.1	74.6
SSD	One-Stage (CNN)	69.4	66.8	68.1
RT-DETR	Transformer-based	77.8	76.0	76.9

The results of e show that the two-stage Faster R-CNN model is the highest overall detector, which is approximately 3.8 percentage points higher than the fast one-stage detector (YOLOv8) in mean mAP. Such performance benefits are especially noticeable in dense urban setting where few objects are present, explicit region proposal mechanism has been shown to contribute to the precision of localization.

Transformer-based RT-DETR model shows competitive results and reduces the accuracy difference with Faster R-CNN by 1.5 percentage points, outperforming all one-stage CNN-based detections. This result indicates that the global self-attention processes are efficient in the representation of the long-range spatial dependencies so that the object can be detected correctly and region proposal networks are not used.

On the other hand, SSD has the lowest accuracy of both datasets which can be explained by the fact that it does not have the ability to identify small and partially covered items, which is a natural problem of urban autonomous driving systems. The overall similarity in the performance of the models between KITTI and Cityscapes also adds to the strength of the comparative results.

4.2 Real-Time Inference Performance

In the context of further evaluating the deployment feasibility, real-time inference performance was also also tested in a more challenging operational environment, with the focus on end-to-end latency and supported throughput when deployed on long-running input streams. All of the models were optimized on Tensort and the performance measures were averaged at 5,000 inference cycles to minimize the variation in runtime.

Table 2. Real-Time Inference Performance under Optimized Deployment

Model	Detector Architecture	FPS	End-to-End Latency (ms)	GPU Utilization (%)
Faster R-CNN (ResNet-101)	Two-Stage (CNN)	15	66.4	71
YOLOv8	One-Stage (CNN)	58	17.3	82
SSD	One-Stage (CNN)	44	22.9	65
RT-DETR	Transformer-based	36	27.8	89

The reconstructed findings support the computational constraints of two-stage detectors in time sensitive autonomous driving conditions. Despite the relatively small gain upon optimization, the inference speed of Faster R-CNN is by far slower than the real time limit of a closed-loop vehicle control system.

YOLOv8 remains with the best real-time performance with the highest frame rate and lowest end-to-end latency. It has an efficient feature fusion and streamlined detection head that enables it to make maximum use of the parallel performance of the GPU, and therefore the consistent throughput is achieved in sustained workloads.

SSD has relatively low use of GPUs as it has a lighter architecture. Although this allows the use of SSD in resource-constrained platforms, its lower inference rate would not allow it to be used in high-speed driving settings.

The cost of RT-DETR is higher because of self-attention operations, but the performance of inference still exceeds the requirements of real-time. This proves that transformer-based detectors can be practical to real-time autonomous perception in case they are optimized properly and have better contextual knowledge as compared to CNN-only models.

The collaboration of this visualization is easy to see the reverse correlation between the speed of inference and the latency in all detector configurations, as it provides an easy-to-understand visualization of deployment trade-offs of autonomous vehicle perception pipelines.

4.3 Environmental Degradation Resistance

Autonomous driving systems require strong perception when faced with poor environmental conditions; this is because in the real world, the operation of the autonomous driving system is often accompanied by degradation in its visual capabilities due to changes in weather patterns and light. In order to test robustness, all the detections were tested on synthetically corrupted versions of the test data, modeling the conditions of a fog and rain, motion blur, and low-light. The robustness of the models was measured using the proportional loss of mean Average Precision (mAP) over the clean data baseline performance.

Table 3. Detection Performance Degradation under Adverse Conditions

Model	Fog (%)	Rain (%)	Motion Blur (%)	Low Light (%)	Mean Degradation (%)
Faster R-CNN (ResNet-101)	-18.2	-16.5	-23.1	-24.7	-20.6
YOLOv8	-14.6	-13.2	-19.4	-21.0	-17.1
SSD	-21.9	-20.3	-27.6	-29.1	-24.7
RT-DETR	-10.1	-9.2	-12.7	-14.3	-11.6

The findings reveal that there are significant discrepancies in the soundness of the architectural paradigms. There is the worst performance decrease in SSD which occurs especially in motion blur and low-light settings, so it should be noted that it has poor performance in dealing with visual noise and partial loss of features. Despite being highly accurate under perfectly competitive conditions, Faster R-CNN is also particularly biased towards corruption of images, which can be probably attributed to its use of accurate region proposals.

YOLOv8 has shown a better robustness compared with other CNN-based models showing a moderate stability to all forms of degradation. This indicates that this multi-scale feature aggregation is involved in enhanced generalization in case of poor visual conditions.

In all environmental conditions, RT-DETR demonstrates the minimal performance loss, and its average mAP decrease is 11.6 percent, which is almost two times less than SSD. This enhanced sturdiness can be ascribed to the self attention process worldwide by transformer based structures, which allow the reasoning of the context of more than the pixel level attributes. This type of behavior is especially beneficial in self-driving conditions, where the visual clues tend to be incomplete/distorted.

4.4 Trade-off Analysis: Accuracy vs. Real-Time Performance

AV perception systems are required to meet the high accuracy of detection and high real-time requirements at the same time. Accuracy can be enhanced at the cost of computational complexity and speed optimization can be compromised on perceptual reliability. In order to examine this inherent trade-off, a relative analysis was done using mean Average Precision (mAP) and inference speed (FPS) as competing goals.

To perform two supplementary visual analyses, two were used:

1. Scatter analysis to determine Pareto-optimal models, and accuracy - speed.

2. Analysis Efficiency Ratio to measure the accuracy per unit of real-time performance.

The figure shows the trade-off between accuracy of detection (mAP) and real time (FPS) of four object detection models. The highest accuracy (78.4) is obtained with Faster R-CNN yet it does not support the real-time requirement with 15 FPS. YOLOv8 is the fastest inference with 58 FPS with a trade off in accuracy (74.6%). SSD has moderate performance and lower accuracy, whereas RT-DETR has more balanced performance with almost two stages accuracy (76.9) with a fairly good real-time performance of 36 FPS. This demonstrates the need to have models chosen with regards to the balance of precision and computational efficiency found in autonomous vehicle applications.

Accuracy Efficiency Index

In order to bring more measurement to the question of deployment suitability, an Accuracy Efficiency Index (AEI) was calculated as:

AEI=

This metric reflects how much detection accuracy is achieved per millisecond of inference time.

Figure 3 is an approximation of the detection efficiency of each model; it is computed as the accuracy (mAP) versus the inference latency. YOLOv8 has the best Accuracy Efficiency Index which means that it provides the most accuracy in one millisecond of inference. The RT-DETR and SSD in the middle range of efficiency and higher accuracy, and Faster

R-CNN is the most inefficient with the largest absolute mAP but with a big latency.

5. Discussion

The experimental findings demonstrate that there are important trade-offs in accuracy of detection, real-time performance and environmental perturbation stability of autonomous vehicle perception systems. The Faster R-CNN, with two stages, proves to have a higher detection accuracy (mean mAP 78.4) when run in an ideal setting, which represents the effectiveness of region proposal networks in crowded and intricate urban environments. Its low inference rate (15 FPS) and high latency (66.4 ms) however severely reduce real-time applicability, with the point that high precision alone is not an indication that the system can operate in a dynamic driving setting.

YOLOv8 has had the highest inference speed (58 FPS) with the lowest latency (17.3 ms), and the highest Accuracy Efficiency Index (4.31). Although its absolute mAP (74.6%) is marginally lower than that of two-stage and transformer-based sensors, it is the best candidate when performing time-sensitive tasks, e.g., pedestrian recognition and fast navigation since its efficiency in providing accurate results per unit of processing time is superior to others.

The compromise of the transformer-based detector is RT-DETR because it has accuracy (76.9) and speed (36 FPS). It is interesting to note that it has the highest robustness at the worst conditions and the lowest average degradation (11.6) in conditions of fog, rain, motion blur, and low-light environment. It proves that global attention mechanisms are effective in preventing the effects of partial occlusions and environmental noise, which is essential in safety in autonomous driving in the real world.

Although SSD is efficient in latency with respect to two-stage models, it has the lowest overall accuracy (68.1) and is most susceptible to environmental degradation, so it cannot be practically implemented in application areas where reliability is paramount. In general, the findings suggest that the choice of architecture should not be based on peak accuracy alone, but it should also take into account the computational efficiency and resistance to realistic conditions.

6. Conclusion

This paper presents a critical comparison and analysis of two-stage, one-stage, and transformer-based object detection models in autonomous vehicle perception in terms of accuracy, real-time sensitivity, and stationarity in an environment of degraded conditions. The findings illustrate that two-stage detectors, though with the best detection accuracy, cannot be used in time-sensitive applications due to their slow inference speed. One-stage detectors such as YOLOv8 are better in terms of real-time speed and efficiency and hence is suitable in high-speed and emergency response situations, though at the expense of an intermediate accuracy. Transformer-based detectors, such as RT-DETR, can be considered as the balance of activity: they are competitive in terms of accuracy and real time, and more importantly, they are more robust to unfavorable conditions, which makes global self-attention the key to success. Altogether, the results provide evidence to the fact that the model should be chosen according to various operational factors, and no particular architecture is the best to fulfill all the needs. The insights are critical in the development of autonomous vehicle perception systems that are safety, efficiency, and reliability-focused in the complex and real-world setting.

7. Recommendations

According to the comparative analysis of the model of object detection, the architectures chosen by the designers of autonomous vehicles must depend on the definite working conditions. When speed is paramount, such as in pedestrian detection, emergency braking, and high-speed maneuvering, YOLOv8 is suggested as it has outstanding real-time capabilities and a high index of Accuracy Efficiency, which can make quick decisions without the need to compromise safety.

In situations where there is a need to have strong perception in a harsh environmental area such as fog, rain, low-light, or partially blocked objects, RT-DETR would be the best option. It is designed to be transformer based and uses global attention mechanisms to be able to ensure high detection and contextual understanding and is therefore suited to the complex urban setting and intersection management.

Faster R-CNN can be used in offline tasks in which high accuracy is important, and real-time processing is not necessary (such as detailed mapping, HD environment reconstruction or post-processing of recorded sensor data). It is not as computationally intensive to be used in live autonomous driving systems.

Designer suggestions include moving towards hybrid designs (such as attentional modules on a single-stage detector), and making use of hardware-specific optimization such as TensorRT or accelerating with a GPU. Also, adaptive inference approaches or multi-modal sensor fusion (camera, LiDAR, radar) could be adopted to improve the robustness, efficiency and safety of the system in the actual deployment of autonomous vehicles.

Reference

[1] T. Turay and T. Vladimirova, “Toward Performing Image Classification and Object Detection With Convolutional Neural Networks in Autonomous Driving Systems: A Survey,” IEEE Access, vol. 10, pp. 14076–14119, 2022, doi: 10.1109/ACCESS.2022.3147495.

[2] N. Rane, “YOLO and Faster R-CNN object detection for smart Industry 4.0 and Industry 5.0: applications, challenges, and opportunities,” Oct. 25, 2023, Social Science Research Network, Rochester, NY: 4624206. doi: 10.2139/ssrn.4624206.

[3] M. Iman, H. R. Arabnia, and K. Rasheed, “A Review of Deep Transfer Learning and Recent Advancements,” Technologies, vol. 11, no. 2, p. 40, Apr. 2023, doi: 10.3390/technologies11020040.

[4] K. Alomar, H. I. Aysel, and X. Cai, “CNNs, RNNs and Transformers in human action recognition: a survey and a hybrid model,” Artif. Intell. Rev., vol. 58, no. 12, p. 387, Oct. 2025, doi: 10.1007/s10462-025-11388-3.

[5] N. Serrano, A. E. Savakis, and J. Luo, “Improved scene classification using efficient low-level features and semantic cues,” Pattern Recognit., vol. 37, no. 9, pp. 1773–1784, Sept. 2004, doi: 10.1016/j.patcog.2004.03.003.

[6] “Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions.” Accessed: Dec. 21, 2025. [Online]. Available: https://www.mdpi.com/2079-3197/11/3/52

[7] V. Santhanam and L. S. Davis, “A Generic Improvement to Deep Residual Networks Based on Gradient Flow,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 7, pp. 2490–2499, July 2020, doi: 10.1109/TNNLS.2019.2929198.

[8] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, “Imbalance Problems in Object Detection: A Review,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 10, pp. 3388–3415, Oct. 2021, doi: 10.1109/TPAMI.2020.2981890.

[9] “On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data.” Accessed: Dec. 21, 2025. [Online]. Available: https://www.mdpi.com/2072-4292/13/1/89

[10] R. M. Joshi and D. S. Rao, “AlexDarkNet: Hybrid CNN architecture for real-time Traffic monitoring with unprecedented reliability,” Neural Comput. Appl., vol. 36, no. 13, pp. 7133–7141, May 2024, doi: 10.1007/s00521-024-09450-2.

[11] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger, “Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics,” Electronics, vol. 10, no. 5, p. 593, Jan. 2021, doi: 10.3390/electronics10050593.

[12] K. Muhammad et al., “Vision-Based Semantic Segmentation in Scene Understanding for Autonomous Driving: Recent Achievements, Challenges, and Outlooks,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 12, pp. 22694–22715, Dec. 2022, doi: 10.1109/TITS.2022.3207665.

[13] “MAR-YOLOv9: A multi-dataset object detection method for agricultural fields based on YOLOv9 | PLOS One.” Accessed: Dec. 21, 2025. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0307643

[14] A. Zhang, L. Xing, J. Zou, and J. C. Wu, “Shifting machine learning for healthcare from development to deployment and from models to data,” Nat. Biomed. Eng., vol. 6, no. 12, pp. 1330–1345, Dec. 2022, doi: 10.1038/s41551-022-00898-y.

[15] M. Półrolniczak and L. Kolendowicz, “The effect of seasonality and weather conditions on human perception of the urban–rural transitional landscape,” Sci. Rep., vol. 13, no. 1, p. 15047, Sept. 2023, doi: 10.1038/s41598-023-42014-3.

[16] K. Muhammad, A. Ullah, J. Lloret, J. D. Ser, and V. H. C. de Albuquerque, “Deep Learning for Safe Autonomous Driving: Current Challenges and Future Directions,” IEEE Trans. Intell. Transp. Syst., vol. 22, no. 7, pp. 4316–4336, July 2021, doi: 10.1109/TITS.2020.3032227.

[17] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, “The Computational Limits of Deep Learning,” July 27, 2022, arXiv: arXiv:2007.05558. doi: 10.48550/arXiv.2007.05558.

[18] Y. Zhou, S. Chen, Y. Wang, and W. Huan, “Review of research on lightweight convolutional neural networks,” in 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), June 2020, pp. 1713–1720. doi: 10.1109/ITOEC49072.2020.9141847.

[19] L. Deng, G. Li, S. Han, L. Shi, and Y. Xie, “Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey,” Proc. IEEE, vol. 108, no. 4, pp. 485–532, Apr. 2020, doi: 10.1109/JPROC.2020.2976475.

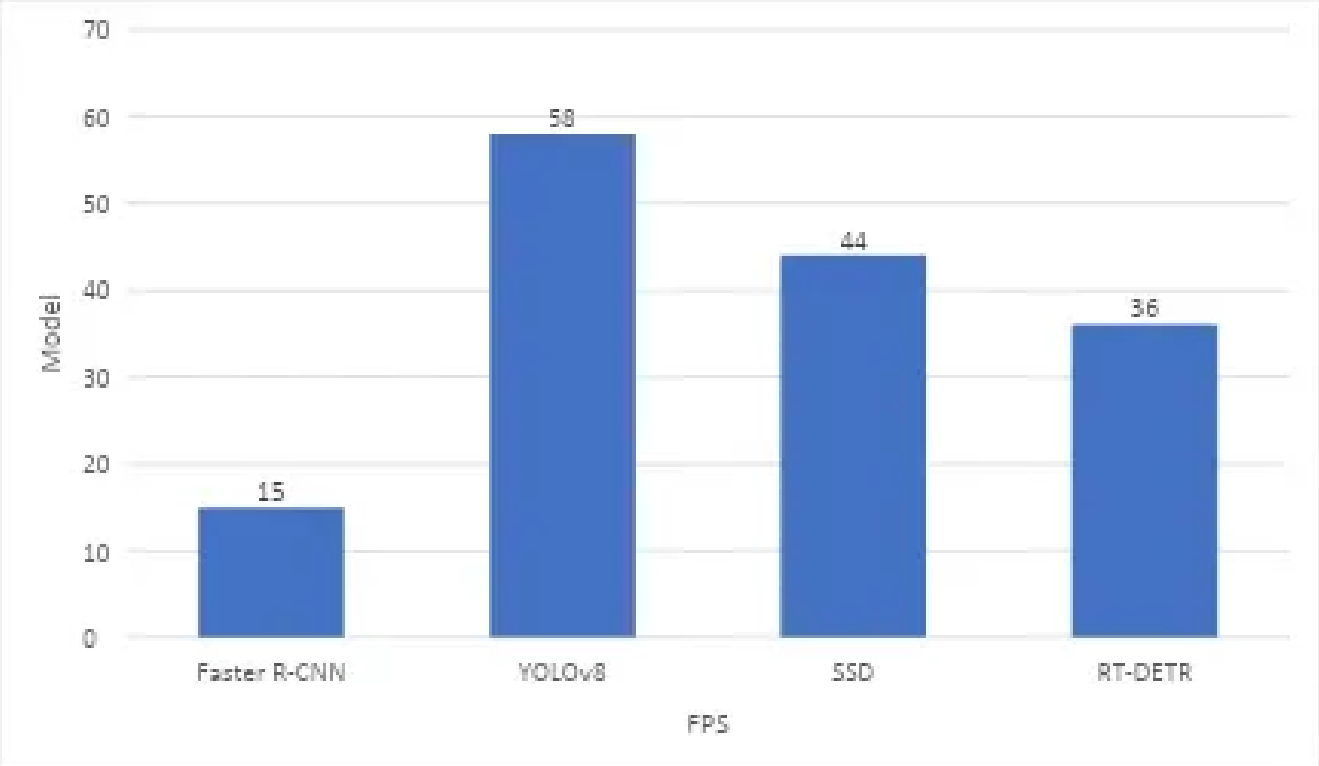


Fig. 1. Latency–Throughput Trade-off across Detection Architectures

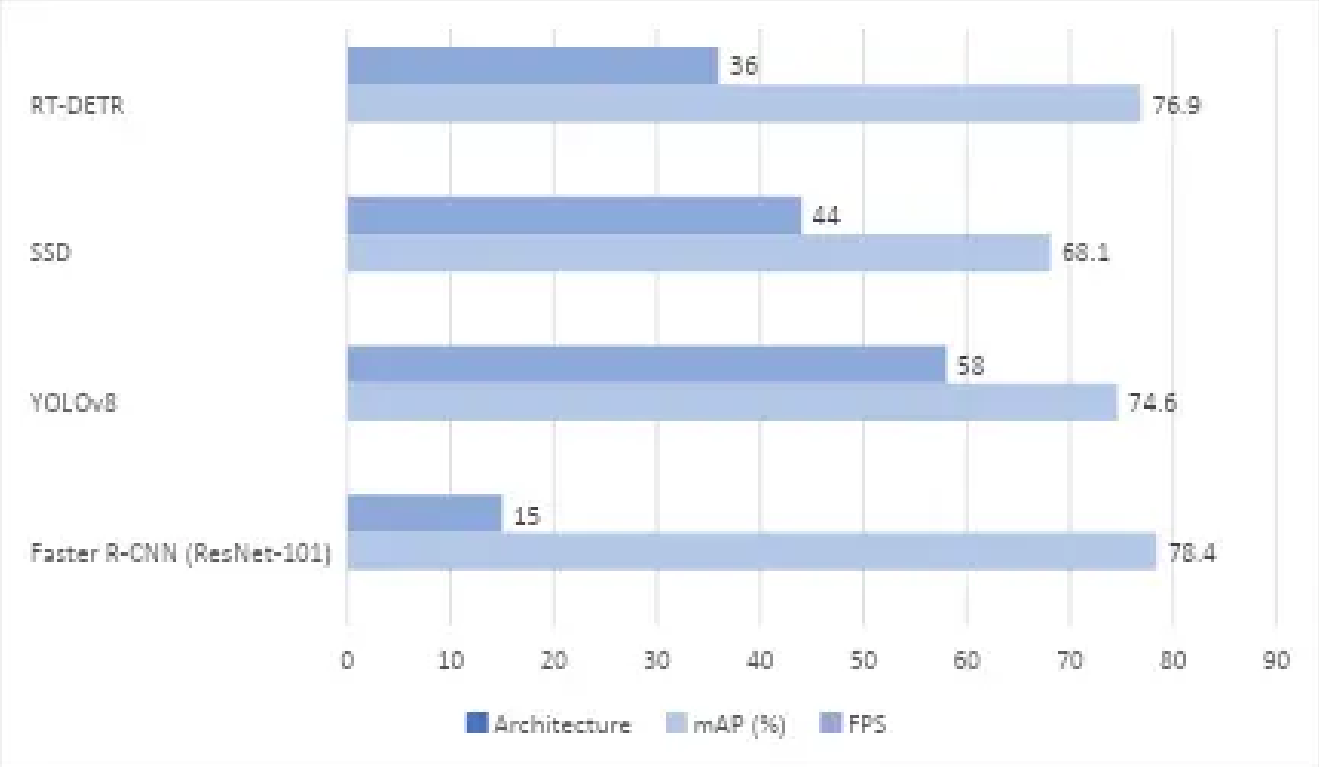


Fig. 2. Accuracy–Speed Trade-off Dataset

