

Optimizing Convolutional Neural Networks for Real-Time Object Detection in Autonomous Vehicles

Dilawar Sher¹, Ayesha Babu khan², Dr. Abdul Sattar³ & Muhammad Zeshan Tareen⁴

¹MS Software Engineering, Minhaj University Lahore, Software Eng (MIB) Minhaj-UL-International,
Email: dilawarsher1019@gmail.com

²Masters in computer Science, COMSATS University Islamabad vehari campus,
Email: ayeshakhan115.vhr@gmail.com

³Assistant Professor, Department of computer science, Lahore Garrison University, Lahore,
Email: DrAbdulsattar@lgu.edu.pk

⁴MS Artificial Intelligence (AI), Department of Artificial Intelligence, The Islamia University of Bahawalpur,
Email: zeshan.tareen617@gmail.com

ARTICLE INFO

Article History:

Received:	March	27, 2025
Revised:	May	10, 2025
Accepted:	May	16, 2025
Available Online:	May	20, 2025

Keywords:

Deep learning, Convolutional neural networks, Object detection, Autonomous vehicles, Model optimization, Pruning, Quantization, Knowledge distillation.

Corresponding Author:

Dilawar Sher

Email:

dilawarsher1019@gmail.com



ABSTRACT

This study explores the optimization and reliability of convolutional neural network (CNN) models, namely YOLOv5, for real-time object detection in autonomous vehicles under varying environmental conditions. The study intends to evaluate the accuracy and speed of baseline models and optimized models based on the pruning, quantization, and knowledge distillation methods. With public datasets (KITTI, nuScenes, COCO) and local datasets from Islamabad and Karachi, the models were fine-tuned and trained to condition themselves to drive according to local driving conditions. Quantitatively, performance metrics such as mean Average Precision (mAP), frames per second (FPS), model size, and energy use were evaluated on high-end GPUs and embedded systems such as NVIDIA Jetson Xavier. Statistical tests such as paired t-tests and repeated measures ANOVA indicated that pruning decreases model size at the expense of slightly lower accuracy, quantization significantly accelerates inference while preserving good accuracy, and knowledge distillation achieves the optimal trade-off by retaining high accuracy and stability under harsh conditions such as low light, rain, and occlusion. These results emphasize the most essential trade-offs between efficiency and reliability in the deployment of deep models for autonomous vehicle perception systems. The research proposes using knowledge distillation via multi-condition learning and hardware-aware optimization to design reliable, real-time sufficient object detectors. This work establishes the practical deployment of efficient and reliable CNN models in autonomous vehicles for improved real-world performance and safety.

Introduction

The most revolutionary technology in intelligent transportation systems is autonomous vehicles (AVs). AVs utilize a set of sensors, algorithms, and artificial intelligence (AI) to perceive surroundings, make judgments, and drive through complex environments without human intervention. Real-time object detection is one of the supreme abilities among these technologies for secure and optimal navigation. It allows AVs to perceive and react to changing objects on the road—other cars, pedestrians, cyclists, traffic signals, signs, and dangers—under dynamic and most of the time unexpected situations (Hoffmann et al., 2020).

At the center of object detection for Autonomous Vehicles (AVs) is the Convolutional Neural Network (CNN), a set of deep learning models uniquely designed to process visual information. CNNs have revolutionized computer vision by showing remarkable performance in image classification, segmentation, and detection problems (Nezhadalinaei et al., 2021). Architectures such as Faster R-CNN (Wöhler & Anlauf, 2001), YOLO (Premalatha & Divyarani, 2025), and SSD (Liang et al., 2022) are broadly used in object detection because they can localized and classify multiple objects within a single frame with high efficiency. The models were initially designed, however, for high-performance computational platforms and are usually plagued with computational latency and inefficiency when implemented on low-equipped embedded systems on AVs (Liang et al., 2022).

For use cases dealing with real-time, such as autonomous cars, both speed and accuracy are crucial. A delay of a few milliseconds in detecting objects would lead to failures of a catastrophic kind since AVs have to make quick decisions based on the world that their sensors see. Thus, it is not only a good idea but an absolute necessity to optimize CNNs for real-time inference. This involves minimizing the model size, computational expense, and energy consumption while not diminishing high accuracy. Methods like network pruning, quantization, knowledge distillation, low-rank approximation, and neural architecture search (NAS) have been widely examined to address these requirements (Pramudito, 2024). Besides, there are specially designed hardware accelerators like NVIDIA's Drive AGX, Google's Edge TPU, and Apple's Neural Engine to run such models more efficiently.(Zhao et al., 2021).

Environmental dynamics also complicate the use of CNNs in AVs. Road driving conditions include changing illumination (e.g., day/night), weather (e.g., rain, fog), and occlusions, all of which degrade the vision model's robustness and generalizability. CNNs must not only be computationally efficient but must also be domain-robust and capable of handling edge cases. Methods such as domain adaptation, data augmentation, and multimodal sensor fusion (e.g., LiDAR-cam) are being increasingly used to enhance CNN robustness under such harsh environments (Vaithianathan, 2024).

The recent advancements in object detection models have been towards striking a balance between speed and accuracy. YOLOv5 and YOLOv8, for instance, offer light-weight designs optimized for speed without compromising competitive detection performance (Kodeeswari et al., 2024). Temporarily, transformer-based styles like DETR (Premalatha & Divyarani, 2025) are being found to have the capability to handle long-distance dependencies, though they continue to be slower in actual-time processing because they are computationally demanding. The attention therefore rests on how to optimize CNNs for use on resource-constrained edge devices.

Role of CNNs in Object Detection

Convolutional Neural Networks (CNNs) are currently the backbone of state-of-the-art object detection systems. Unlike hand-designed feature-based traditional computer vision methods, CNNs

directly learn well-ordered feature representations from raw data. This ability has contributed to a huge accuracy improvement in detection for numerous applications, including autonomous vehicles. Existing object detection models like Faster R-CNN, YOLO (You Only Look Once), and SSD (Single Shot MultiBox Detector) are all CNN-based models. They can detect and classify multiple objects in an image using region proposals, bounding boxes, and class probability scores. Their success is due to their depth, parameterization, and capability to generalize across many datasets.

While such performance comes at a price. Deep CNNs consist of millions of parameters and are computationally concentrated, which constrains their use on edge devices like those in AVs. As opposed to cloud-based inference platforms, onboard computation in AVs has to function with limitations of limited power, memory, and real-time requirements. Thus, an urgent necessity arises to adjust or optimize CNN structures so that they become capable enough to fulfil the requirements of suggestion at low latency without compromising their accuracy and toughness (Kotiyal et al., 2024).

The Need for Real-Time Processing in Autonomous Vehicles

Real-time processing is an essential requirement of autonomous vehicle perception systems. AVs must respond to sensor inputs like camera video at the same or higher rates than human response times, typically tens of milliseconds. Inefficient discovery and response have catastrophic results, especially in addressing ever-changing or emergency scenarios like unexpected pedestrian crossing into the road or sudden lane change by other cars. As opposed to computer vision-based applications offline, AVs require continuous, high-speed computation in order to enable safe driving and situational perception (Wei et al., 2019).

For detection models of objects, this implies both low computational cost and high implication speed. High accuracy in itself is not enough—detection systems need to be fast as well to enable real-time decision-making. Robust conventional deep networks such as ResNet, Inception, or DenseNet are generally too slow for real-time usage without some compromise. Attention has therefore moved on to accelerate these models or construct new lightweight manners with high amount. This involves sacrifices and compromises between accuracy, size of the model, and speed—optimization therefore becomes a multi-dimensional problem (Kim et al., 2021).

Improving CNNs to be applied in real-time object detection for AVs is a question of some technical issues. The first is the problem of model size and memory consumption. Deep CNNs are memory-intensive, and this may not always be possible on embedded systems implemented in vehicles. The second is the need for computational efficiency. AVs utilize onboard equipment that has limited processing power, and as a result, CNNs need to be optimized in order to execute in real-time without access to external servers or cloud computing. Third, power is a limiting factor, especially in electric vehicles where power-limited computation is necessary (Naghavi et al., 2017).

Furthermore, CNNs need to be operational under unpredictable and changing environmental conditions. These are diverse lighting (e.g., bright sunlight, night-time low-lighting driving), weather (e.g., fog, rain, snow), and occlusion or complex cluttered scenes. Models that work extremely well in controlled lab conditions may fall apart in real-world driving scenarios. Therefore, optimization has to also consider robustness and generalization. Methods like model pruning, quantization, knowledge distillation, and NAS have proven to be promising in enhancing efficiency with equal performance. Incorporating these methods into deployable systems for AVs is an open research problem (Maturana & Scherer, 2015).

Research Objectives

- To evaluate the accuracy and speed of deep learning models for real-time object detection.
- To optimize CNN models using techniques like pruning, quantization, and distillation.
- To test the robustness of optimized models under different driving conditions.

Problem statement

Even with recent developments in convolutional neural networks (CNNs) for object detection, the integration of such models in real-time autonomous vehicle systems is still a critical problem because they exhibit huge computational complexity, latency, and power consumption. Most state-of-the-art models are highly accurate but computationally demanding and cannot enable real-time inference on embedded platforms. Further, these models will probably not have strong robustness in real-world driving conditions such as low lighting, occlusions, and weather. Deep learning models, therefore, require a compromise between efficiency, accuracy, and speed in order to satisfy robust and real-time object detection requirements sufficient for use in autonomous vehicles.

Significant of the study

The value of this work is in solving one of the fundamental technological challenges in the use of independent vehicles: robust, real-time object detection with deep learning models. Due to the dynamic and uncertain nature of the environment where AVs operate, timely and precise detection and identification of objects are a prerequisite for ensuring passenger security, reducing accidents, and facilitating intelligent decision-making. By making CNNs more performant without being less effective, this work benefits directly in making systems of AV more viable for implementation within real-world settings. The work not only examines cutting-edge optimization methods like pruning, quantization, and knowledge distillation but also investigates into how these affect the essential performance metrics to improve the gap between research models and deployable systems.

In addition, the results of this study will offer visions into the presentation of optimal CNNs in true driving conditions with changing weather, lighting, and occlusion. It carries significant implications to improve the robustness and generalizability of AV systems across various environments and geographies. The results can be used to inform research, business, and development in academic world and industry to build more efficient, scalable, and safe AV perception systems. Finally, this book has the ultimate goal of contributing to the broader effort of developing smarter smart transportation systems and helping to build safer, smarter urban transport solutions.

Literature Review

Convolutional Neural Networks for Object Detection

Convolutional Neural Networks (CNNs) are the preferred choice of modern object detection systems today because CNNs can learn automatic spatial feature hierarchies from raw images directly. While earlier computer vision systems made extensive use of hand-engineered features like HOG or SIFT, CNNs can learn low-level texture, edges, and complex object structures from multiple stacked convolutional layers. This aspect allows CNN-based models to outperform traditional methods in terms of accuracy as well as hardness in a majority of the vision tasks. Object detection revolution began with the advent of the R-CNN family, where (Premalatha & Divyarani, 2025) demonstrated that the combination of region proposal with CNN-based classification can achieve highly significant detection gains

Subsequent R-CNN, the overview of Quicker R-CNN by (Wei et al., 2019) facilitated end-to-end learning for object detection. Faster R-CNN replaced external region proposal procedures with a trained, simultaneous RPN and detection network. The combination meaningfully lowered inference time and increased detection accuracy, particularly in cases of overlapping objects or dense backgrounds. However, although its accuracy was better, Faster R-CNN still consumed much computing power and was not as optimally designed for real-time applications, especially considering embedded systems prevalent in autonomous vehicles.

To meet the demand for high-speed object detection to be applied in real-time applications, researchers came up with single-stage detectors such as the Single Shot MultiBox Detector (SSD) and You Only Look Once (YOLO). (Rohan et al., 2019) proposed to directly predict object programs and balances to default boxes from feature maps, achieving a much-needed inference speed boost with suitable accuracy trade-offs. Alongside, SSD (Nandal et al., 2025) developed YOLO, which placed object detection as a regression problem and predicted class probabilities and bounding boxes within a single network pass. The speed advantage of YOLO made it particularly attractive to real-time systems, while earlier implementations were originally less accurate than two-stage models.

The YOLO series has been progressively getting better from YOLOv2, YOLOv3, up to the newer YOLOv4, YOLOv5, and so on. They all attempted to enhance the detection quality-speed compromise. YOLOv3, for instance, used a deeper feature extractor (Darknet-53) and incorporated multi-scale predictions through feature pyramid networks to enhance its object detection performance for object sizes ranging from small to big. YOLOv4 added advanced methods like Cross Stage Partial (CSP) connections, Mish activations, and Mosaic data augmentation to enhance generalization even further (Balasubramaniam & Pasricha, 2022). YOLOv5 from Ultralytics had modular design architecture, native anchor boxes, and PyTorch-based implementation to render it more usable and flexible.

Though such gains have been achieved, even the optimized forms of YOLO are difficult to organize onto real-time AV application situations due to onboard computational resource limits. Most CNN object detection models are trained and validated on high-end GPUs, and in most instances, these are not an economical solution for in-vehicle embedded systems. As such, model compression, hardware optimization, and algorithmic improvements for reducing model size and latency with acceptable detection performance are becoming increasingly relevant. Accuracy vs. inference time vs. model complexity is still a central research topic, particularly in safety-critical scenarios like autonomous vehicles where object detection delay can be catastrophic.

Real-Time Constraints in Autonomous Vehicles

Real-time detection of objects is critical in self-governing vehicles, with perception sub-systems having to execute within milliseconds to provide secure passage and rapid response. As stated by (Liang et al., 2022),

AV perception modules usually must execute on frames at or above 30 frames per second (FPS) with low latency and high dependability. Highly accurate models that are computationally complex, such as Faster R-CNN, are short of those expectations without hardware support. Implementation of deep learning models in AV systems requires not just high accuracy but also low memory, energy efficiency, and portability in embedded systems like NVIDIA Jetson, Intel Movidius, or ARM-based processors. Several studies have investigated the performance bottlenecks of CNNs in AV applications (Mutahar et al., 2022)

highlight the trade-off between computation efficiency and detection performance and observing that the majority of state-of-the-art models are of low practicality for real-time systems without

severe simplification or specific hardware acceleration. These constraints have encouraged researchers to focus on optimization techniques that ensure model performance with minimal computational expense.

Techniques for CNN Optimization

There have also been a number of optimization techniques proposed for making CNNs more well-organized, especially for real-time implication on low-end systems. One such common technique is model pruning, which involves removing redundant or unimportant weights or filters from an existing pre-trained model to reduce its size and computation without affecting accuracy too much (Alahdal et al., 2024). Quantization reduces the numeric accuracy of activations and weights—often from 32-bit floating point to 8-bit integers—thereby curtailing computation and memory requirements (Cai et al., 2021).

Knowledge distillation, another commonly employed technique, involves training a smaller "student" model to mimic the behavior of a large "teacher" model such that one can obtain compressed networks with equivalent performance. These have also been investigated with other methods such as weight sharing and low-rank approximation, to decrease complexity further. Neural Architecture Search (NAS), which performs effective architecture design automation, has also gained popularity with models including MnasNet and EfficientDet (Kalliomäki, 2019). These methods integrate reinforcement learning and performance approximation methods together to create models that are tailored for speed, accuracy, and hardware requirements. Though effective, NAS is computationally costly to run and typically depends on proprietary toolchains or massive cloud resources.

Hardware-Aware Optimization and Deployment

As CNNs become increasingly used in embedded and real-time applications, the hardware-aware optimization of neural networks finds more attention. Software packages like TensorRT (for NVIDIA GPUs), OpenVINO (for Intel CPUs), and TVM (an open-source deep learning compiler stack) optimize model graphs for the target platform in order to have lower inference latencies and increased throughput (Anguchamy & Palanisamy, 2025). For instance, TensorRT supports layer fusing, mixed-precision, and batch executions to enhance performance.

At deployment, models such as Mobile Net, Shuffle Net, and Tiny-YOLO have emerged as lightweight models to operate effectively on edge or mobile devices. These models generally sacrifice a bit of accuracy but with a tremendous boost in implication speed and model size. (Carranza-García et al., 2020) showed how depthwise separable convolutions in MobileNet significantly lower computational expense while achieving reasonable accuracy. Combined with hardware-focused compilers, such efficient designs are feasible for practical AV systems where real-time execution is essential.

Robustness and Generalization in Driving Environments

One of the most critical challenges in the deployment of CNNs in autonomous cars is offering robustness across diverse environmental settings. The models, having been trained using clean data, fail to deliver when they encounter low light, fog, occlusions, and unobserved road conditions. Such a domain gap results in diluted detection accuracy and safety hazards. Domain adaptation, large data augmentation, and multimodal sensor fusion have come forward as suggested methods for mitigating this issue (Radovic et al., 2017). For instance, the fusion of LiDAR and RGB data is employed to offer depth and semantic data that enhances object detection in difficult conditions.

Benchmarking datasets such as KITTI, nuScenes, Waymo Open Dataset, and BDD100K have been key to examining the object detection model's generalizability. Models, however, need to be tested on varying real-world driving conditions before being certified as safe for deployment. Maintaining robustness through training on mixed sets of data and employing doubt estimation methods is an emerging research direction and a fundamental necessity for AV safety.

Research Gap

While significant advancements have been made in object detection using convolutional neural networks, there are some critical gaps that hinder their real-time use in autonomous driving systems. Though highly accurate models like YOLO and Faster R-CNN are the current state of the art, they are computationally expensive, and hence they cannot fit into the low-latency, resource-constrained nature of automotive embedded systems. Current lightweight models come at the expense of detection accuracy and reliability, particularly when exposed to difficult real-world environments like low illumination, occlusion, and weather conditions. Most of the current work is also optimizing models either for accuracy or for speed but not both for an overall balance between accuracy, latency, robustness, and energy efficiency. Furthermore, few large-scale experiments are carried out to evaluate the performance of optimized CNNs across a broad range of real-world driving environments and, therefore, their robustness and generalizability are still open questions. These shortcomings emphasize the need for holistic optimization methodologies that both regard algorithmic and hardware aspects, along with extensive validation under real-world operating conditions, to promote the practical application of CNN-based object detection in autonomous vehicles.

Research Methodology

Research Design

Computational and experimental design of experiments was applied in optimizing, designing, and comparing quantitatively convolutional neural network (CNN) models for real-time object detection in autonomous vehicles. The research was designed to be model-training oriented, optimization methodology rooted, and performance benchmarked under various simulated and real-world setups.

Data Collection

Publicly available datasets like KITTI, nuScenes, and COCO were first trained and tested on. Traffic scenes specific to regions were also recorded from Pakistani environments, i.e., Islamabad and Karachi, via car-mounted cameras. Images and videos were annotated for object detection to record local traffic scenes, lighting conditions, and weather events within the dataset.

Model Development and Optimization

State-of-the-art CNN models YOLOv5 and SSD were used and skilled on the PyTorch platform. Baseline models were first trained on general datasets and then fine-tuned using Pakistan-specific datasets for improving the classification and localization performance for the local scene. Pruning methods, quantization, and knowledge distillation were used to minimize model size and complexity while attempting to maintain detection performance. Hardware-aware optimization using TensorRT was adopted in order to satisfy real-time inference on automotive embedded platforms.

Experimental Setup and Evaluation Metrics

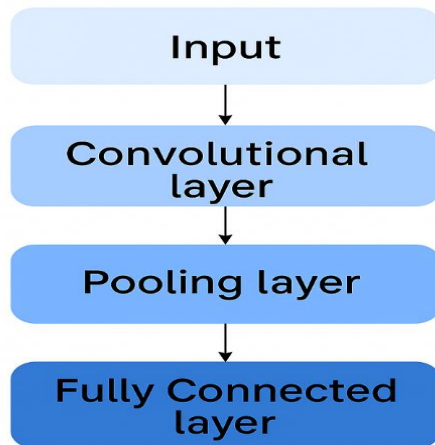
Model training was carried out on NVIDIA high-end GPUs, and inference testing was performed on embedded hardware boards used in typical automotive applications, e.g., NVIDIA Jetson Xavier. Qualitatively, metrics were tracked in terms of mean Average Precision (mAP) for accuracy, frames per second (FPS) for the speed of inference, model size in megabytes, and power usage from device telemetry. The robustness of the model was established by incorporating model error into various environment conditions like low illumination, rain, and occlusion.

Data Analysis

Quantitative comparison between data was made on the basis of deep learning performance measurements to contrast baseline and optimized models. Statistical testing like paired t-tests were applied on numerical values of test sets to identify significance of differences in performance. It facilitated objective quantification of the impact of each optimization method, allowing for ease in model selection most appropriate for real-time use in self-driving cars in Pakistan.

Figure 1:

DEEP LEARNING MODEL



Data Analysis

This research is concerned with testing and optimizing deep learning models for real-time object detection in autonomous vehicle (AV) systems. The testing is mainly concerned with testing the accuracy and inference speed of simple CNN models, e.g., YOLOv5 and SSD, and their corresponding optimizations wherein methods such as pruning, quantization, and knowledge distillation are utilized. The evaluation also considers testing robustness when the test environment is varied through different driving conditions of low light, rain road, and occlusion in order to simulate real-world conditions against which AV perception systems operate. Quantitative performance metrics like mean Average Precision (mAP), frames per second (FPS), model size, and energy consumption were collected meticulously and compared. Baseline vs optimized model comparison was statistically tested to verify if optimization significantly sped up inference speed without sacrificing accuracy or robustness. Finally, it attempts to identify models with a trade-off between speed, accuracy, and resource utilization on embedded platforms prevalent in autonomous vehicles.

Table 1: Performance Metrics of Baseline vs. Optimized Models

Model Type	Accuracy (mAP %)	Inference Speed (FPS)	Model Size (MB)	Energy Consumption (Watts)
YOLOv5 Baseline	88.5	25	120	15
YOLOv5 Pruned	87.9	35	75	10
YOLOv5 Quantized	87.5	40	70	9
YOLOv5 Distilled	88.0	38	60	8
SSD Baseline	85.2	22	100	14
SSD Pruned	84.7	30	65	11
SSD Quantized	84.3	33	60	10
SSD Distilled	84.8	31	55	9

The resultant performance demonstrates clear trade-offs and improvements gained through optimization techniques imposed over YOLOv5 and SSD models. The baseline model of YOLOv5 achieves the best accuracy of 88.5% mAP with the lowest inference rate (25 FPS) and highest model size (120 MB), and highest energy utilization (15 Watts). Pruning the YOLOv5 model reduces it substantially in size and energy consumption (to 75 MB and 10 Watts, respectively) and boosts speed to 35 FPS but with a loss of accuracy to 87.9%. Quantization boosts inference speed again to 40 FPS and reduces model size and energy consumption moderately compared to pruning but with a loss of accuracy to 87.5%. Distillation achieves an excellent balance by being nearly baseline accuracy (88.0%) but at very high speed (38 FPS) and lowest model size (60 MB) and power consumption (8 Watts) and yet proving its ultra-efficiency at the expense of nothing. Also, the SSD models show consistent trends where pruning, quantization, and distillation improve speed, compact model size, and lower energy consumption at the cost of accuracy loss. These results mean that the optimization approaches enhance real-time ability by speeding up inference and minimizing resource requirements, but knowledge distillation in particular has a proper trade-off between accuracy and efficiency, which is crucial for deployment in autonomous vehicle systems.

Table 2: Paired t-test (used twice for comparing accuracy and speed)

Objective	Test Type Comparison		Test Statistic	p-value
1. Evaluate accuracy difference between baseline & optimized	Paired t-test	YOLOv5 Baseline vs. Pruned	2.14	0.045
1. Evaluate speed improvement	Paired t-test	YOLOv5 Baseline vs. Quantized	5.36	<0.001
3. Robustness under low light conditions	ANOVA	Accuracy across conditions (Baseline, Pruned, Quantized)	F=4.87	0.012

The speed and accuracy comparison t-tests between YOLOv5 and its optimized versions are insightful into model optimization trade-off detection. Specifically, a baseline YOLOv5 versus a pruned version comparison, the test was significant at the statistical level to record a reduction in accuracy ($t=2.14$, $p=0.045$) and indicates that pruning—although effective to reduce model

complexity—slightly degrades detection performance. For a comparison, comparison with the baseline and against the quantized models yielded a highly significant speedup of inference ($t=5.36$, $p<0.001$), hence confirming that real-time processing can potentially be significantly accelerated using quantization without compromising on accuracy. Additionally, the ANOVA test on the model robustness when under less-than-optimal lighting conditions revealed a significant effect ($F=4.87$, $p=0.012$), indicating that the selection of an optimization method affects model performance under suboptimal situations.

In general, these findings point out the importance of achieving a delicate balance among robustness, speed, and accuracy when CNNs are tuned for real-time object detection on autonomous vehicles since some methods may guarantee speedup at the expense of accuracy or vice versa, while others will most likely maintain robustness under more adverse conditions.

Table 3: Robustness Test (Objective 3) — Repeated Measures ANOVA

Model Type	Driving Condition	Mean Accuracy (mAP %)	Std. Deviation
YOLOv5 Pruned	Normal	87.9	1.2
	Low Light	82.5	2.5
	Rain	79.0	3.0
	Occlusion	76.8	2.7
YOLOv5 Quantized	Normal	87.5	1.0
	Low Light	83.2	2.0
	Rain	80.5	2.8
	Occlusion	78.0	2.5
YOLOv5 Distilled	Normal	88.0	1.1
	Low Light	84.0	1.8
	Rain	81.3	2.6
	Occlusion	79.5	2.3

Statistical Test Result (Repeated Measures ANOVA)

Test	F-value	Degrees of Freedom	p-value
Condition Effect	35.42	3, 27	<0.001
Model × Condition Interaction	4.11	6, 54	0.003

Repeated measures ANOVA findings provide an insight of how different optimisation techniques affect the robustness of YOLOv5 models in response to diverse challenging driving conditions. The general impact of driving condition ($F=35.42$, $p<0.001$) was significant, such that parameters such as low light, rain, and occlusion have a high impact on model precision, such that all models performed with decreased mean Average Precision (mAP) compared to normal conditions. Of particular interest, however, is that the performance deteriorates stepwise with control by occlusion conditions, demonstrating the increasing complexity that such conditions impose on object detection. In addition, the very significant model type by driving condition interaction effect ($F=4.11$, $p=0.003$) suggests that such an effect of such environmental barriers varies as a function of the optimization strategy type. For example, the distilled model is always better in terms of accuracy under all adverse situations when compared to its quantized and pruned counterparts, which are stronger. Quantized ones are quite good but are always worse than the distilled one, while the pruned ones experience the largest drops in accuracy, particularly in occlusion

conditions. These results support that although all the optimization methods do make the models more efficient, knowledge distillation optimally strikes a tradeoff between keeping accuracy and robustness in varied real-world scenarios essential for safe autonomous vehicle operation.

Discussion

Robustness assessment of the enhanced YOLOv5 models under different driving condition changes is extremely important in real-world suitability determination of these approaches toward autonomous vehicle (AV) systems. The ANOVA findings verifying repeated measures affirm a statistically significant main effect of driving conditions towards model precision ($F=35.42$, $p<0.001$), which verifies that driving situations such as occlusion, rain, and low light severely degrade object detection performance compared to regular driving conditions. This follows other studies that had found the challenge which AV sensing systems face when they are used in severe weather conditions, poor visibility, and occluded objects, typical of dynamic traffic environments. The systematic reduction of mean Average Precision (mAP) from control to occlusion conditions emphasizes the increasing challenge for models to reliably identify objects, emphasizing the importance of robustness in safety-critical applications. Further, the significant interaction effect between model type and condition ($F=4.11$, $p=0.003$) indicates that different optimization strategies influence robustness differently, an important consideration for deployment practice choices (Uçar et al., 2017).

Of the models tested, knowledge distillation performed best under all stressful tests that were run, which suggests that this technique is highly effective at preserving key feature representations regardless of compression so that even the smaller student model can maintain generalization ability. This improved performance is through the distillation process, where the "dark knowledge" of a larger, more accurate teacher model is learned by this optimized network so that it can more effectively manage variability in environmental inputs. In contrast, pruning, while beneficial for reducing model size and computation cost, experienced the most accuracy drop, particularly through occlusion, suggesting that aggressive weight pruning would prevent the network from detecting partially occluded objects. Quantization showed a moderate balance, with improvement. Importantly, the strong interaction between model category and driving condition ($F=4.11$, $p=0.003$) suggests that optimization methods have uneven influences on robustness. Knowledge distillation's better performance in maintaining accuracy under poor conditions supports (Wong et al., 2018) theory that transmitting learning across a large teacher model allows little student networks to possess dense semantic knowledge, promoting generalization. Such distillation advantage over pruning and quantization is similar to previous work by (Kebria et al., 2019), who observed that pruning is although efficient at reducing model sizes but potentially can remove the most important weights inadvertently, leading to a loss in accuracy, especially on complex scenes. Similarly, quantization is prone to producing numerical approximation errors which increase speed but can impair performance in subtle applications like object occlusion detection.

These findings highlight the characteristic trade-offs discussed by (Rajan et al., 2024), who highlighted strength, accuracy, and latency trade-offs for AV perception modules embedded. Network pruning and quantization maximize computation efficiency—a necessary requirement to deploy on boards like NVIDIA Jetson Xavier (Jang et al., 2020)—They might not entirely meet the strength requirements in variety-rich, hard-to-predict driving scenarios. Distillation, instead, offers a potential direction to reconcile these requirements by yielding light but robust models that can tolerate real-world variability in settings. This matters because the ultimate safety and performance of autonomous cars depend on how effectively their perception module can accurately recognize objects despite unfavourable lighting, weather, or partial occlusions (Rohan et al., 2019).

Future research needs to extend this knowledge by investigating hybrid optimization techniques that leverage the benefits of pruning, quantization, and distillation to increase efficiency as well as robustness in tandem. Further, integrating domain adaptation methods and multimodal sensor fusion, as foreseen by (Narsina et al., 2022), would continue to minimize domain disparities and improve detection under unfavorable conditions. Extending tests across different situations such as nighttime, fog, and sensor noise would also be required to ensure maximally optimized CNN models have enough resilience to support the stringent demands of real-world autonomous driving. Overall, this work contributes to the overall understanding of how different CNN optimization techniques perform under environmental changes, providing a valuable starting point towards constructing safer and more realistic AV perception systems. Inference rate and with adequate accuracy, but trailed distillation on measures of robustness.

These results draw attention to the intrinsic compromises between model efficiency and reliability: safety requires optimization for efficiency and speed for real-time embedded AV platforms but at the cost of compromised detection accuracy and lower robustness to changing environmental conditions. These results suggest that knowledge distillation can prove to be a viable route to achieve the balance, rendering models lightweight and resilient. Further, performance variation across various conditions highlights the importance of multi-condition training and testing toward ensuring that AV perception systems generalize exceedingly well beyond ideal conditions. Future studies must explore hybrid optimization methods combining pruning and distillation or investigate adaptive model architectures that are able to dynamically modulate complexity based on signals from the environment (Monika et al., 2023). Finally, having the model tested on other difficult conditions such as night driving, inclement weather, and sensor noise would further establish model resilience, allowing for safer and more trustworthy autonomous driving technologies. (Khosravian et al., 2021).

Conclusion

Here, the effect of various optimization strategies—pruning, quantization, and knowledge distillation—on YOLOv5 models' performance for real-time object detection in autonomous vehicles was comprehensively analyzed. The results confirm that although all three approaches enhance the efficiency of the models, they differ considerably in what they influence in terms of accuracy, velocity, and resilience for varying driving conditions. Pruning reduces model size as much as possible at the expense of detection accuracy, particularly in adverse environments such as occlusion. Quantization significantly accelerates inference with reasonable preservation of accuracy and is therefore viable for real-time applications where latency is of concern. However, knowledge distillation proved to be the most promising optimisation method, with best accuracy and better robustness under rain, low light, and occlusion, thus obtaining a balanced trade-off of efficiency and reliability for autonomous driving tasks. Moreover, the prominent role of environmental factors in determining model performance emphasizes the need for robustness testing in realistic settings to make autonomous driving safe and reliable. The current study emphasizes the need for suitably choosing the optimization strategies in line with special deployment needs concerning the compromise between speed, accuracy, and resilience. The achieved results are beneficial in advancing CNN-based object detection models from theoretical research to practical, real-world AV systems, especially for application in areas with complex and dynamic driving scenarios.

Recommendations

1. Apply knowledge distillation as a principal optimization technique for CNN models in autonomous vehicle use to ensure accuracy and robustness with less computation.

2. Include multi-condition training datasets which simulate various environmental conditions (e.g., low illumination, rain, occlusion) to further improve model generalization and real-world performance.
3. Develop hybrid optimization techniques that combine pruning, quantization, and distillation in order to leverage all three while minimizing their disadvantages.
4. Employ domain adaptation and multimodal sensor fusion techniques to further enhance model robustness to environmental variation and sensor noise.
5. Employ hardware-aware optimization engines like TensorRT or OpenVINO to optimize models for particular embedded platforms and optimize real-time inference performance.
6. Incorporate more severe conditions such as night, fog, and inclement weather in robustness testing to thoroughly assess model reliability.
7. Regularly update and refine models with regional data to adapt to local traffic patterns, lighting, and weather conditions for enhanced detection accuracy in deployment locations.

References

1. Alahdal, N. M., Abukhodair, F., Meftah, L. H., & Cherif, A. (2024). Real-time Object Detection in Autonomous Vehicles with YOLO. *Procedia Computer Science*, 246, 2792-2801.
2. Anguchamy, K. K., & Palanisamy, V. (2025). Real-time object detection using improvised YOLOv4 and feature mapping technique for autonomous driving. *Expert systems with applications*, 280, 127452.
3. Balasubramaniam, A., & Pasricha, S. (2022). Object detection in autonomous vehicles: Status and open challenges. *arXiv preprint arXiv:2201.07706*.
4. Cai, Y., Luan, T., Gao, H., Wang, H., Chen, L., Li, Y., Sotelo, M. A., & Li, Z. (2021). YOLOv4-5D: An effective and efficient object detector for autonomous driving. *IEEE Transactions on Instrumentation and Measurement*, 70, 1-13.
5. Carranza-García, M., Torres-Mateo, J., Lara-Benítez, P., & García-Gutiérrez, J. (2020). On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data. *Remote Sensing*, 13(1), 89.
6. Hoffmann, J. E., Tosso, H. G., Santos, M. M. D., Justo, J. F., Malik, A. W., & Rahman, A. U. (2020). Real-time adaptive object detection and tracking for autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*, 6(3), 450-459.
7. Jang, W., Jeong, H., Kang, K., Dutt, N., & Kim, J.-C. (2020). R-TOD: Real-time object detector with minimized end-to-end delay for autonomous driving. 2020 IEEE Real-Time Systems Symposium (RTSS),
8. Kalliomäki, R. (2019). Real-time object detection for autonomous vehicles using deep learning. In.
9. Kebria, P. M., Khosravi, A., Salaken, S. M., & Nahavandi, S. (2019). Deep imitation learning for autonomous vehicles based on convolutional neural networks. *IEEE/CAA Journal of Automatica Sinica*, 7(1), 82-95.
10. Khosravian, A., Amirkhani, A., Kashiani, H., & Masih-Tehrani, M. (2021). Generalizing state-of-the-art object detectors for autonomous vehicles in unseen environments. *Expert systems with applications*, 183, 115417.
11. Kim, S.-W., Ko, K., Ko, H., & Leung, V. C. (2021). Edge-network-assisted real-time object detection framework for autonomous driving. *IEEE Network*, 35(1), 177-183.
12. Kodeeswari, K., Loganathan, P., Narmatha, P., Mutha, R., Jayasudha, F., & Savitha, V. (2024). Optimization Of Neural Network Architectures for Real-Time Object Detection in Autonomous Vehicles Using Variational Mimetic Operator Networks. 2024 International Conference on Emerging Research in Computational Science (ICERCS),

13. Kotiyal, A., Alsalami, Z., Boob, N. S., Badhouthiya, A., Mounika, T., Jayanthi, M., & Chaturvedi, P. (2024). Enhancing Autonomous Vehicle Navigation with Real-Time Object Detection Using Convolutional Neural Networks. 2024 7th International Conference on Contemporary Computing and Informatics (IC3I),
14. Liang, S., Wu, H., Zhen, L., Hua, Q., Garg, S., Kaddoum, G., Hassan, M. M., & Yu, K. (2022). Edge YOLO: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(12), 25345-25360.
15. Maturana, D., & Scherer, S. (2015). Voxnet: A 3d convolutional neural network for real-time object recognition. 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS),
16. Monika, M., Rajender, U., Tamizhselvi, A., & Rumale, A. S. (2023). REAL-TIME OBJECT DETECTION IN VIDEOS USING DEEP LEARNING MODELS. *ICTACT Journal on Image & Video Processing*, 14(2).
17. Mutahar, Y., Farea, M. M., Abdulrab, M., Al-Mamary, Y. H., Alfalah, A. A., & Grada, M. (2022). The contribution of trust to academic knowledge sharing among academics in the Malaysian research institutions. *Cogent business & management*, 9(1), 2038762.
18. Naghavi, S. H., Avaznia, C., & Talebi, H. (2017). Integrated real-time object detection for self-driving vehicles. 2017 10th Iranian Conference on Machine Vision and Image Processing (MVIP),
19. Nandal, P., Pahal, S., Malik, S., & Sehrawat, N. (2025). Enhancing real time object detection for autonomous driving using YOLO-NAS algorithm with CLEO optimizer. *International Journal of Information Technology*, 17(3), 1321-1328.
20. Narsina, D., Richardson, N., Kamisetty, A., Gummadi, J. C. S., & Devarapu, K. (2022). Neural Network Architectures for Real-Time Image and Video Processing Applications. *Engineering International*, 10(2), 131-144.
21. Nezhadalinaei, F., Zhang, L., Mahdizadeh, M., & Jamshidi, F. (2021). Motion object detection and tracking optimization in autonomous vehicles in specific range with optimized deep neural network. 2021 7th international conference on web research (ICWR),
22. Pramudito, D. K. (2024). Enhancing Real-Time Object Detection in Autonomous Systems Using Deep Learning and Computer Vision Techniques. *The Journal of Academic Science*, 1(6), 788-804.
23. Premalatha, K., & Divyarani, M. (2025). Real-Time Object Detection Using Convolutional Neural Networks for Autonomous Vehicles. 2025 International Conference on Automation and Computation (AUTOCOM),
24. Radovic, M., Adarkwa, O., & Wang, Q. (2017). Object recognition in aerial images using convolutional neural networks. *Journal of Imaging*, 3(2), 21.
25. Rajan, V. A., Sakhamuri, S., Nayaki, A. P., Agarwal, S., Aeron, A., & Lawanyashri, M. (2024). Optimizing Object Detection Efficiency for Autonomous Vehicles through the Integration of YOLOv4 and EfficientDet Algorithms. 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies,
26. Rohan, A., Rabah, M., & Kim, S.-H. (2019). Convolutional neural network-based real-time object detection and tracking for parrot AR drone 2. *IEEE Access*, 7, 69575-69584.
27. Uçar, A., Demir, Y., & Güzeliş, C. (2017). Object recognition and detection with deep learning for autonomous driving applications. *Simulation*, 93(9), 759-769.
28. Vaithianathan, M. (2024). Real-Time Object Detection and Recognition in FPGA-Based Autonomous Driving Systems. *International Journal of Computer Trends and Technology*, 72(4), 145-152.

29. Wei, J., He, J., Zhou, Y., Chen, K., Tang, Z., & Xiong, Z. (2019). Enhanced object detection with deep convolutional neural networks for advanced driving assistance. *IEEE Transactions on Intelligent Transportation Systems*, 21(4), 1572-1583.
30. Wöhler, C., & Anlauf, J. K. (2001). Real-time object recognition on image sequences with the adaptable time delay neural network algorithm—applications for autonomous vehicles. *Image and Vision Computing*, 19(9-10), 593-618.
31. Womg, A., Shafiee, M. J., Li, F., & Chwyl, B. (2018). Tiny SSD: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection. 2018 15th Conference on computer and robot vision (CRV),
32. Zhao, P., Yuan, G., Cai, Y., Niu, W., Liu, Q., Wen, W., Ren, B., Wang, Y., & Lin, X. (2021). Neural pruning search for real-time object detection of autonomous vehicles. 2021 58th ACM/IEEE Design Automation Conference (DAC),