A Hybrid YOLO-NAS Technology in Transforming Autonomous Vehicle Manufacturing

Adi El-Dalahmeh[1,*], Layla Albdour[1], and Mohammad Alia[1]

[1] *Department of Cybersecurity, Faculty of Science and Information Technology, Al-Zaytoonah University of Jordan, Amman 11733, Jordan.*

[*] *Corresponding author: a.eldalahmeh@zuj.edu.jo ORCID: 0000-0001-8591-5987*

Received: 21 February 2026 **Accepted:** 30 July 2026 **Published:** 6 October 2026
DOI: 10.15849/ijasca..68 **License:** Open access under CC BY 4.0. Authors retain copyright.

Publisher: Al-Zaytoonah University of Jordan (ZUJ), Amman, Jordan.
Cite this article: *A.El-Dalahmeh , L.Albdour and M. Alia ,"A Hybrid YOLO-NAS Technology in Transforming Autonomous Vehicle Manufacturing," Int. J. Advance Soft Compu. Appl., vol. 18, no. 3, pp. 18–34, 2026, doi: https://doi.org/10.15849/ijasca..68*

ABSTRACT

Object detection, the core fundamental task of the environmental perception module for autonomous driving, still faces a bottleneck that is difficult to break through. Four types of interference, namely severe weather, low illumination, partial occlusion, and sensor noise, all reduce detection accuracy, which in turn weakens the reliability of core decision-making for safe driving. This study proposes a hybrid perception scheme based on YOLO-NAS that meets the indispensable requirement of real-time performance. By integrating the NAS-optimized architecture and connecting multi-step processing in sequence, the scheme improves the robustness of recognition in complex scenarios. The proposed framework was evaluated on the nuScenes autonomous driving benchmark using the official dataset split. Experimental evaluation under daytime, nighttime, and adverse weather conditions demonstrates that the proposed approach consistently achieves detection accuracies of 94–96%, outperforming YOLOv8 (84–89%) and YOLOv7 (79–81%). Overall, YOLO-NAS provides performance gains of approximately 5–10% over YOLOv8 and 13–25% over YOLOv7 while maintaining stable inference and improved robustness against partially occluded objects and challenging visual conditions. These results demonstrate that incorporating NAS into the YOLO detection pipeline improves perception reliability and detection consistency, offering a practical solution for enhancing the safety, robustness, and real-world deployment of autonomous vehicle perception systems.

Keywords: YOLO-NAS, Object Detection, Autonomous Vehicles, Computer Vision, Neural Architecture Search (NAS)

1 Introduction

In recent years, autonomous vehicles (AVs) have been hailed as a disruptive breakthrough in the transportation sector. Existing studies [1],[2] point out that their core values include reducing road deaths and injuries, alleviating traffic congestion, and reshaping public travel patterns. Moving past the previously widespread misconception that AVs rely on a single sensor technology, today's mainstream AVs adopt a multi-source sensor fusion framework: they collect data via LiDAR, radar, and high-definition cameras to construct a feature space of road conditions, and work with deep learning models to realize real-time perception and operation execution [3]. This developmental trajectory has driven the global automotive industry to shift its core focus from mechanical design to software and intelligent development [4]. As artificial intelligence (AI) and machine learning are integrated into vehicle design, vehicular network security has become a core standard, and AVs have evolved into mobile computing systems capable of continuously perceiving, analyzing, and adjusting their behavior [5].

Even as the global pace of technology iteration continues to accelerate, object detection technology for autonomous vehicles (AVs) remains a top-tier challenge in the field of AV research and development, as noted in reference [6]. AV systems must continuously identify five core categories of targets amid complex, dynamic and unpredictable road conditions: pedestrians, other vehicles, traffic lights, road signs, and random obstacles. Once either type of failure occurs —false alarms triggered by cross-category misjudgment, or missed target detection—it will not only spark an immediate safety crisis, but

also lead to eroded public trust as outlined in reference [7], as well as rising costs from additional system updates, vehicle recalls, and regulatory upgrades. Accordingly, the industry has put forward the urgent appeal documented in reference [8] for a high-precision, stable, and robust detection model adaptable to all scenarios.

Over the past few years, the YOLO series has become the mainstream solution for real-time object detection in the field of computer vision. Among all its iterative versions, YOLOv7 has attracted particular attention from practitioners. The efficient layer merging method and reparameterization technique it introduced achieved performance improvements: it runs faster while better retaining scaling characteristics [9],[10]. However, this model is not perfect. Its hard-coded architecture leads to insufficient flexibility, and the static attributes of its units make it highly prone to failure and unstable performance in complex driving environments such as low-light night conditions, congested traffic scenarios, severe weather, partially occluded objects, and multi-type road scenarios [11].

YOLOv8 was developed to address the inherent flaws of previous generations of YOLO series models, with core improvements to its robustness and generalization ability across diverse datasets. Its three core technical upgrades are a redesigned loss function [12], a streamlined network architecture [13], and enhanced architectural flexibility [14]. While retaining the iconic real-time detection advantage of the YOLO family, it has achieved improved detection accuracy and been widely applied in academic research and industrial scenarios. However, it still suffers from issues including detection errors in scenes with overlapping targets or partial occlusion, and performance degradation in harsh environments [15]. In high-risk scenarios such as autonomous driving, even minimal perception errors can threaten operational safety.

Existing research literature [16] points out that the fixed network architectures of YOLOv7 and YOLOv8 cannot adapt to the diverse dynamic operating conditions of autonomous driving scenarios. Drawing on the real-time inference requirements proposed in references [17],[18] including cross-environment stability, low false detection rates, and compatibility with safety decision-making—this study proposes integrating YOLO-series object detection with Neural Architecture Search (NAS). NAS can automatically optimize network topology to generate customized architectures, replacing the pure manual design approach [19].

This study focuses on the field of autonomous driving perception. Driven by the aforementioned research, it integrates the YOLO-NAS model into the manufacturing and deployment of autonomous vehicles, aiming to improve perception accuracy and reliability without sacrificing real-time performance; unlike traditional YOLO models that adopt a fixed network design, the NAS-powered YOLO-NAS can adapt to various driving scenarios, reduce false detections, and enhance operational safety and system trustworthiness.

The mature YOLOv7 and YOLOv8 models have substantially advanced the development of real-time object detection technology, but they still suffer from unresolved environmental adaptation defects. YOLO-NAS combines the high detection efficiency of YOLO frameworks with the architecture optimization capabilities of Neural Architecture Search (NAS), adapts to complex driving scenarios, improves detection performance under extreme operating conditions, and supports the safe real-world deployment of autonomous driving.

- This paper proposes a hybrid object detection framework based on YOLO-NAS for autonomous vehicles.
- The driving detection framework proposed in this paper can improve detection accuracy and reliability in complex scenarios.
- The road environment detection model proposed in this study reduces inference time, cuts false detections, and improves user trust.
- YOLO-NAS integrates NAS-driven architecture optimization to enhance its perception capabilities, providing accurate and reliable decision support for autonomous vehicle navigation.

 ISSN 2710-1274 (Print) · 2074-8523 (Online) · Published by Al-Zaytoonah University of Jordan (ZUJ)

2 Related Work

2.1 Overview of the AV Industry

The autonomous vehicle (AV) industry combines advances in automotive engineering and intelligent software systems to create vehicles that can operate independently with little to no human intervention. It relies on multiple sensing technologies including LiDAR, radar, and cameras to collect data about the surrounding environment, and processes this data via machine learning algorithms to support safe navigation in complex urban scenarios. According to reference [20], the real-world deployment of AVs can deliver three core benefits: improving road safety, alleviating traffic congestion, and expanding mobility access for groups unable to drive. Figure 1 summarizes the core components and characteristics of the AV industry.

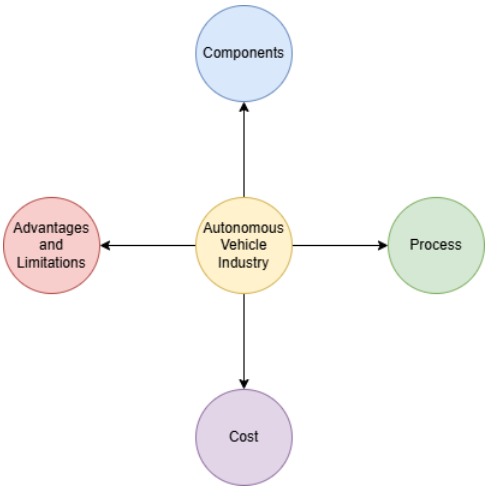


Fig. 1. Autonomous Vehicle Industry Aspects

1) Components [21]:

- Sensors: Cameras, radar, LiDAR, and ultrasonic sensors are used to detect and understand the surrounding environment (Figure 2).
- Actuators: Control acceleration, braking, and steering.
- Control Systems: Analyze sensor data, make decisions, and manage actuators.
- Computational Hardware: High-performance GPUs and CPUs process large data volumes.
- Connectivity: Integrates V2V and V2I communication technologies.
- Software: Includes AI-driven algorithms for perception, prediction, planning, and control.

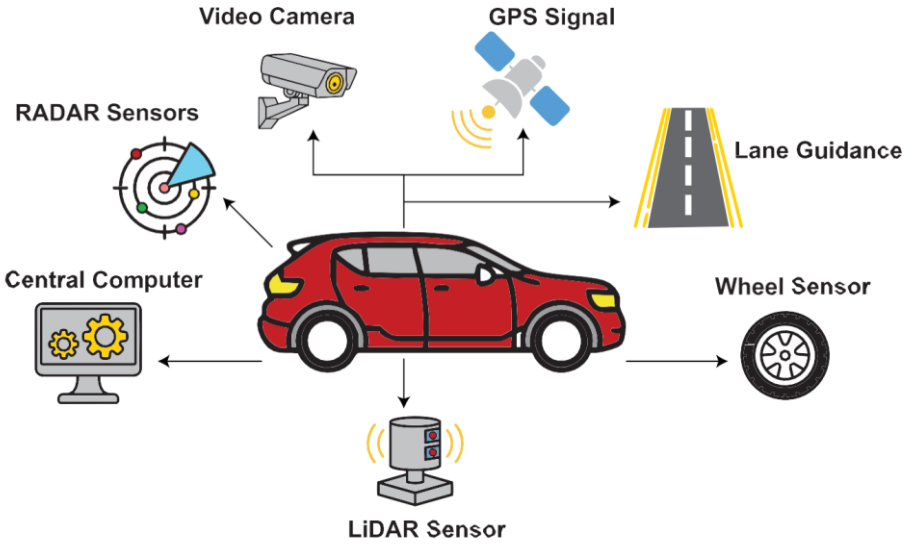


Fig. 2. Primary sensing technologies used in autonomous vehicles for real-time perception, localization, and navigation.

2) Processes [22]:

- Data Collection: Vehicles continuously collect environmental data.
- Data Processing: Algorithms analyze this data to interpret the surrounding environment.
- Decision Making: The system formulates responses based on insights gained from processed data.
- Actuation: The system implements these decisions through actions like steering or accelerating.
- Feedback Loop: Continuous algorithm refinement based on operational feedback.

3) Cost [23]:

- Initial Investment: High due to expensive components like LiDAR sensors.
- Future Reductions: Costs expected to decrease with economies of scale and technological advances.
- Long-Term Savings: AVs may reduce operational costs in taxi and delivery services.

4) Advantages [24]:

- Safety: Reduces accidents caused by human errors, improving road safety.
- Efficiency: Optimized driving reduces congestion and idle time.
- Accessibility: Supports individuals unable to drive, such as the elderly and disabled.
- Productivity: Passengers can work or relax during transit, increasing efficiency.

5) Limitations [25]:

- Technical Challenges: Difficulties in predicting human behavior and environmental factors
- Security Concerns: Vulnerable to hacking and cyber- attacks, risking passenger safety and data security.

2.2 Related Work

Over the past 10 years, there has been significant interest in autonomous driving from both automotive manufacturers and technology firms. A number of leading firms have begun getting away from traditional vehicle production and sinking heavily into self-driving platforms. A prime example of this trend is the rollout of autonomous ride-hailing services in select major cities, demonstrating that AV technology is beginning to transition from controlled research settings into real-world operation.

Sundaresan et al. [26] investigated multi-class vehicle detection using YOLOv8 and YOLOv10, including cars, buses, motorcycles, bicycles, trucks. They trained both models on a well annotated dataset and used extensive data augmentation, then they evaluated performance with common detection metrics like precision, recall, F1-score and confusion matrices along with mean average precision (mAP). YOLOv8 uses an anchor-free design along with a C2f module and new loss functions in their setup, while YOLOv10 introduced several changes which include NMS-free pipeline, dual label assignment, partial self-attention. YOLOv10, on the other hand, achieved superior performance overall (especially for bicycles and trucks), while YOLOv8 slightly outperformed in car detection. Yet the study tested only in static conditions, without changing lighting or weather dynamics, nor did they approach real-time deployment efficiency in autonomous driving scenarios.

Sharma et al. [27] investigated the application of YOLOv7 for real-time vehicle detection to assist intelligent transportation systems and public safety. We trained the model on examples of all kinds of traffic, in different light conditions and rural and urban environments. They then employed transfer learning in an effort to enhance results, optimizing critical hyperparameters and gauging the output against standard performance metrics, including precision, recall and F1-score as well as intersection over union, or IoU. Their findings show that YOLOv7 out-performed in normal driving cases during

day and still achieved reasonable performance under challenging night-time conditions. But the performance began to lag during low-light conditions, suggesting that having additional improvements — like sensor fusion or better image preprocessing — may be needed for detection in dark settings.

Nikarika et al. [28] explored YOLOv7 as well, but modified the model to improve performance for public security vehicle detection applications. They tested their experiments on Kaggle Vehicle Detection dataset: This dataset has 8779 images with about 26000 labelled objects in eight classes. Testing accuracy and a reasonable loss value of the model are achieved by optimizing the parameters. This was further validated by precision–recall curves and confusion matrix analysis of collected data. The benefits of results are impressive, and authors acknowledged though they may be over-estimating their performance based on feature engineering from the data used in combination with its architecture - as useful to validate outside this specific data/context.

With respect to another impracticality area, YOLOv8 was also criticized by [29] in terms of performance set against autotooth detection using a dual-image dental imaging dataset captured orthogonally. The evaluation looked not just at accuracy, but also speed of inference and general computational efficiency. With an mAP@50 = 92.5%, YOLOv8 achieved the best detection performance, and YOLOv6 the fastest runtime of ∼160 FPS. They hypothesized that YOLOv7 might be a nice compromise wherein it maintains an acceptable tradeoff of speed and accuracy to justify use in time-sensitive clinical contexts. They also cited limitations of the dataset in terms of a factor that supported less generalization of results to broader settings.

Azis et al. [30] employs YOLO models in the detection of aerial vehicles, from YOLOv5-L to YOLO10-L versions, specially focusing on small and/or partially occluded objects. They performed their experiments on the high-resolution COWC-M dataset of more than 11,000 aerial images across several urban regions. They used data augmentation techniques, such as Mosaic, to increase the robustness of the model. They found YOLOv10-L to be the best all-around performer (mAP50 = 0.9890). But did continue to drop when vehicles were small or heavily occluded in complex, cluttered scenes. That means that specific advances (like hybrid-model fine-tuning, and evaluation on more challenging datasets) still needed.

Lin et al. [31] conducted another comparative analysis by evaluating YOLO models (v5–v8) for real-time vehicle detection using traffic monitoring. As for datasets, the authors needed to combine Kaggle images with dashcam videos taken from Taiwan highway driving videos, and also manually label their data using LabelImg. Under the conditions of standardizing the training procedures on each model, YOLOv6l obtained 99.5% mAP achieving a great performance in detection, and compared with YOLOv5x and YOLOv7x they showed up a more uniform precision recall curves. Although YOLOv8 achieved similar performance with a shorter training time, it should be pointed out by the authors that certain parameters, such as model depth and dataset size, may have strong correlation on accuracy and computation cost.

Liu et al. [32] relying on the improved YOLOX-based stage, [32] proposed V-YOLO for better detection performance in dense traffic scenarios with occlusions and small-object problems. In the framework, therefore, they designed a wavelet-enhanced convolution module, SimAM attention and a new Varifocal IoU loss function. And indeed they claimed measurable improvements over a baseline YOLOX on the VisDrone dataset using mAP50 and mAP75. However, moderate traffic or bad weather still caused performance degradation - indicating room for further robustness improvement.

Finally, Yusof et al. [33] investigated road-defect detection that includes potholes and cracking patterns using YOLOv5,YOLOv6 and YOLOv7 Their dataset includes 3,328 annotated images captured from a GoPro camera mounted on a moving vehicle and augmented. The detection performance (mAP@0. 5 = 79.0%) with the highest inference speed as well. The authors recognized some limitations (of which they believe all are surmountable in the next iteration), such as size of dataset, resolution and number of training epochs used, that would limit generalization to a larger range of road environments.

Alam et al. [34] This approach leveraged YOLOv5-based models recently employed in detecting vehicles from images coming from fisheye cameras positioned on the roads of urban intersections [34].

Using fisheye images causes real-world problems, such as significant lens distortion and glare, including changing shadows and large variances in objects sizes when they are seen in different locations within the frame. At first, we make a few changes on the general detection pipeline proposed in this paper in order to adapt it for these issues. Significant additions included a lightweight day–night CNN classifier which divides images into classes based on lighting condition so the system can use different detection models for daytime/nighttime scenes. They also sought to improve detection results on hard cases — such as partially-occluded or heavily-distorted vehicles — by upsampling the difficult samples during training. In addition, it utilized a stage-wise transfer learning strategy by sequentially training on COCO and BDD100k and subsequently fine-tuned on fisheye images from custom datasets. Authors also applied a selective ensemble of the best classifiers discovered to reduce false positives and increase robustness of the learned prediction rules. The proposed approach achieved an mAP@0. With respect to a baseline YOLOv5 model, our trained models gain mAP 50 (mean Average Precision at IoU threshold 0.50) improvement that exceeds up to +13.7%, over the whole course of real-time inference speed, with batch-size = 1. But performance was still affected by inconsistent annotations and remains especially sensitive in extremely challenging nighttime conditions, they wrote.

In study [35], the authors experimented with weights from YOLOv5 Nano, being lighter up to these heavy-duty weights of YOLOv5 X-Large, in order to measure their performance of intercepting different vehicles' categories in actual traffic. Their dataset was 1,321 images and to ensure that the models didn't just memorize photos but in fact learned what a car (and a bike and a truck) looked like, they used heavy data augmentation (more on that here: e.g. blur, rotation and "mosaic" mixing). While the pint-sized YOLOv5n proved to be a workhorse for dependable all-around car detection, the meatier Large and X-Large variants really shone for isolating those buses and bicycles. But even the most powerful models get thrown for a loop trying to determine motorcycles from bicycles — which goes to show that despite their abilities, there exist some things that a complex visual environment combined with small dataset can use to topple these AI tools.

3 Proposed Work

3.1 Overall System

To self drive, an AV relies on subsystems. The perception system interprets this data for object detection, tracking, and localization; the cameras, LiDAR, radar, and GPS are all sensors that capture environmental data in real time. The control system tells many actuators in charge of steering, acceleration and braking what to do, while decision-making algorithms use this data to build a plan for where to drive or when to stop. This also includes managing the data Processing, high performance hardware, and software, which require vehicle-to-vehicle (V2V) and Vehicle-to-infrastructure (V2I), for better coordination. Integrated safety mechanisms ensure reliability. In recent years, AVs have made remarkable progress in navigating dynamic environments, driven by advances in mechanical engineering, electrical engineering, computer science, and artificial intelligence. Modern AV architectures integrate these disciplines into a unified system to enable safe and reliable autonomous operation.

1) YOLO-NAS: YOLO is a popular RTOD (Real Time Object Detection algorithm), to detect and classify objects in the image at really high speed. YOLO-NAS takes this a step further with the use of Neural Architecture Search (NAS) because it allows to adjust the architecture of the network more effectively for finding solutions on specific tasks. This dynamic adjustment allows YOLO-NAS to customize its structure per each individual object detection task, thereby enhancing accuracy and performance for more complex datasets.

Extremely helpful in applications where speed and precision are crucial—such as in self-driving car systems or sophisticated surveillance devices—YOLO-NAS continuously refines itself by absorbing

more datasets, adjusting to varying surroundings. Its ability to boost the detection accuracy and their speed in processing enables it to be a highly potentiated real time algorithm.

2) Convolutional Neural Network (CNN): A convolutional neural network (CNN) is a deep learning model designed to process structured data like images, as well as sequential input. CNNs are extensively used in computer vision tasks, including image classification, object detection, and image segmentation. Their primary strength lies in their ability to automatically learn hierarchical feature representations. from raw input data, enabling the extraction of increasingly complex visual patterns across successive network layers.

The architecture of a CNN consists of several essential layers:

- Input Layer: Receives raw data, such as images, and prepares it for further processing.
- Convolutional Layers: Apply learnable filters to extract local patterns and features.
- Activation Layer: Uses functions such as ReLU (Rectified Linear Unit) to introduce non-linearity and enhance learning.
- Pooling Layer: Reduces spatial dimensions by summarizing feature information, improving computational efficiency.
- Fully Connected Layer: Links every neuron from the previous layer to the next, facilitating high-level feature extraction and classification.
- Output Layer: Produces the final prediction, which varies depending on the specific task, such as classification, regression, or segmentation.

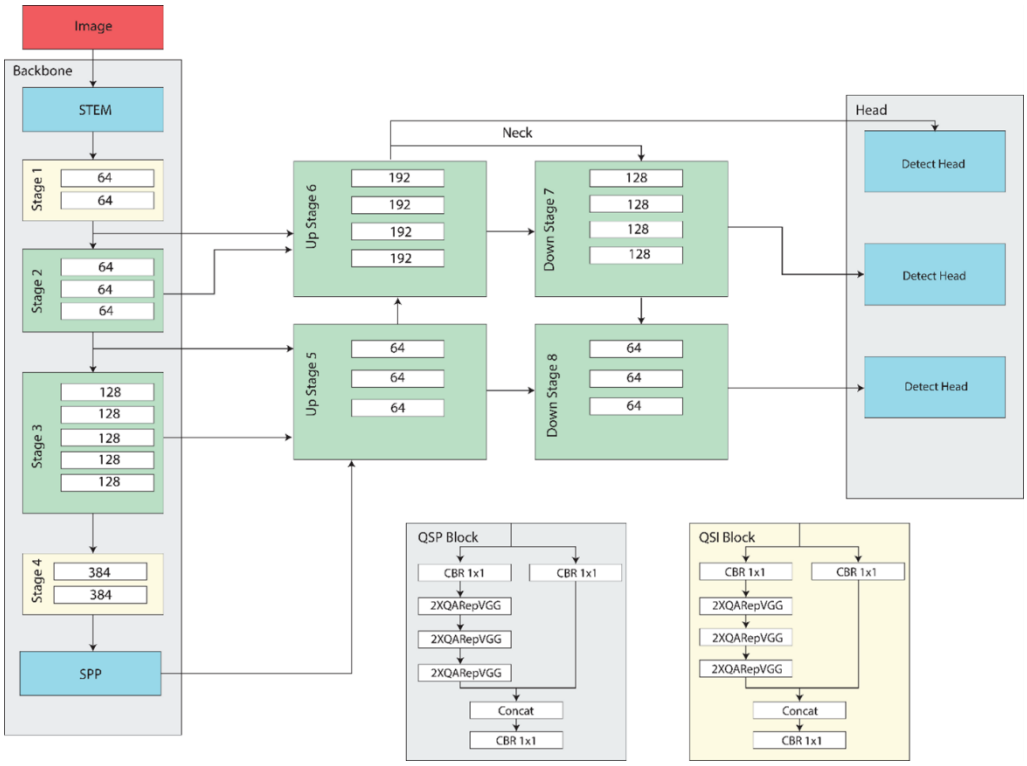


Fig. 3. Architecture of the proposed YOLO-NAS model, illustrating the backbone, neck, and detection heads optimized using Neural Architecture Search (NAS).

3.2 Proposed Work

Our research aims to improve environmental perception in AVs by incorporating a pre-trained YOLO-NAS model, which utilizes a CNN for object detection. Figure 4 illustrates the flowchart of the proposed algorithm. The system consists of five key components designed to enhance detection accuracy and real-time processing.

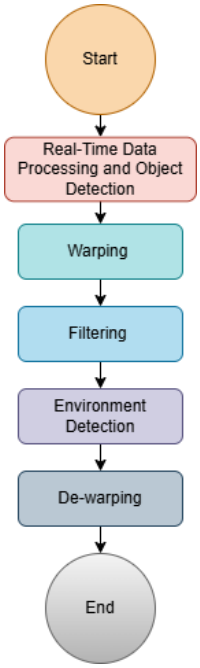


Fig. 4. Flowchart of the proposed YOLO-NAS framework for environment detection in AVs.

1) Real-time Data Processing and Object Detection: The pretrained YOLO-NAS model detects objects in each frame captured by the camera by predicting the class and bounding box of each object.

2) Warping: Using the region of interest (RoI) identified by the YOLO-NAS model, the perspective of the input images is transformed via cv2.warpPerspective. The transformation matrix M is defined by Equations (1) and (2).

$$div(x,y) = M_{31}(x) + M_{32}(y) + M_{33} \quad (1)$$

$$dst(x,y) = \left[\frac{M_{11}(x) + M_{12}(y) + M_{13}}{div(x,y)}, \frac{M_{21}(x) + M_{22}(y) + M_{23}}{div(x,y)}\right] \quad (2)$$

3) Filtering: To isolate the lane color, a filter is applied to the warped image that captures the range of white and yellow colors using LUV and Lab image formats.

4) Environment Detection: Environment points are detected by extracting the nonzero values from the filtered image. These points are segmented into sub images, analyzed for peak nonzero values, and fitted using polynomial regression:

$$P(x,y) = p_2x^2 + p_1y + p_0 \quad (3)$$

The error E from the fitted lanes is calculated as

$$E = \sum (P(x,y) - y)^2 \quad (4)$$

5) De-warping: Finally, the warped images are reverted to the original perspective using the inverse transformation matrix M−1, applying the same formulations as in Equations eq (1) and (2).

The output of the proposed framework is the original perspective of the processed frame, accurately representing the vehicle's environment. These processing steps including object detection, warping,

filtering, environment detection, and de-warping collectively enhance AV perception and decision making.

3.3 Dataset Description

To evaluate the proposed YOLO-NAS framework, experiments were conducted using the nuScenes dataset, a large-scale public benchmark developed specifically for autonomous driving research. The dataset was selected because it provides diverse urban driving scenarios collected under a wide range of environmental conditions, including daytime, nighttime, rain, varying traffic densities, and complex road layouts. These characteristics make it well suited for assessing the robustness and generalization capability of real-time object detection models intended for autonomous vehicle perception.

The nuScenes dataset contains 1,000 driving scenes, comprising approximately 1.4 million camera images together with synchronized sensor measurements from six cameras, five radars, and one LiDAR sensor. The dataset provides approximately 1.4 million annotated 3D bounding boxes covering 23 semantic object categories, including cars, trucks, buses, trailers, construction vehicles, pedestrians, bicycles, motorcycles, traffic cones, and barriers. Official annotations supplied by the dataset developers were used throughout this study.

For the object detection experiments, only the RGB camera images were utilized. Images were resized to match the input resolution required by the YOLO-NAS architecture and normalized before training. Standard image preprocessing and augmentation techniques, including random horizontal flipping, scaling, color jittering, and mosaic augmentation, were applied during training to improve model robustness and reduce overfitting.

The official training, validation, and test partitions of the nuScenes dataset were adopted to ensure a fair and reproducible evaluation protocol. Model development and hyperparameter tuning were performed using the training and validation subsets, whereas all quantitative performance metrics reported in this paper were obtained exclusively from the test set.

The nuScenes dataset was selected because it represents realistic autonomous driving environments with significant variations in illumination, weather, object scale, occlusion, and traffic complexity. These characteristics provide a rigorous benchmark for evaluating the effectiveness of the proposed YOLO-NAS framework under challenging real-world driving conditions.

4 Results and Discussion

4.1 Environment Setup

The proposed autonomous vehicle perception framework was implemented using the YOLO-NAS object detection architecture provided by the SuperGradients 3.7.1 deep learning library, with PyTorch 2.2.2 serving as the primary framework for model training, validation, and inference. Unlike conventional YOLO implementations based on manually designed network architectures, YOLO-NAS employs NAS to automatically optimize the network topology, providing an improved balance between detection accuracy and computational efficiency.

All experiments were conducted on a workstation running Microsoft Windows 11 Pro (Version 23H2). The implementation was developed using Python 3.10.13, PyTorch 2.2.2, and SuperGradients 3.7.1. GPU acceleration was enabled through CUDA 12.1 and cuDNN 9.0, allowing efficient model training and real-time inference. The computer vision pipeline also utilized OpenCV 4.10.0 for image preprocessing, perspective transformation, filtering, visualization, and post-processing operations. Model optimization was performed using the AdamW optimizer with mixed-precision training to improve computational efficiency.

The experimental workflow consisted of four stages. First, images from the nuScenes dataset were preprocessed through resizing, normalization, and data augmentation before being provided to the network. Second, the YOLO-NAS model was trained using the official training partition, while hyperparameter tuning and model selection were performed using the validation partition. Third, the optimized model was evaluated on the independent test partition to ensure an unbiased

assessment of detection performance. Finally, quantitative evaluation and qualitative visualization were performed using the trained detector, and all results reported in this paper, including the detection examples presented in Figures 5–7, were generated from the nuScenes test set.

The complete software environment was configured to ensure compatibility between the operating system, GPU drivers, CUDA toolkit, and deep learning libraries, providing a stable and reproducible platform for training and evaluating the proposed YOLO-NAS framework under diverse autonomous driving scenarios.

4.2 Results

The test results of the YOLOv7 model collected by the authors of this paper are visualized in Figure 5. Across all test scenarios, this model registers a detection accuracy ranging from 79% to 81%. It delivers qualified performance on clear, sunny days, but its reliability declines in nighttime and rainy conditions. When targets are occluded or have insufficient contrast with the background, false detections and missed detections occur far more frequently. These adaptation flaws of such fixed architecture models will disrupt their application in autonomous driving scenarios that require high safety standards.



(a)



(b)



(c)

Fig. 5. YOLOv7 detection results under representative driving conditions, illustrating its performance in daytime, nighttime, and challenging environments.

 ISSN 2710-1274 (Print) · 2074-8523 (Online) · Published by Al-Zaytoonah University of Jordan (ZUJ)

The detection performance of YOLOv8 measured in the experiments of this paper is presented in Figure 6. Compared with YOLOv7, the prior generation model of the same series, YOLOv8 is equipped with an improved loss function and a more flexible network architecture. Its tested accuracy ranges from 84% to 89%, it achieves more stable recognition of pedestrians and vehicles in complex driving scenarios, and it retains the real-time performance essential for autonomous driving.

Although the YOLO series has completed multiple rounds of iterative performance improvements in prior research, the analysis conducted by the authors of this paper points out that YOLOv8 still has critical deployment flaws: it struggles to detect overlapping and small scale objects, and is prone to failure when encountering glare or heavy rainfall. Even though it outperforms older versions of the model, its reliability still declines significantly in real world high dynamic driving scenarios, which are far more complex than ideal scenarios.



(a)



(b)



(c)

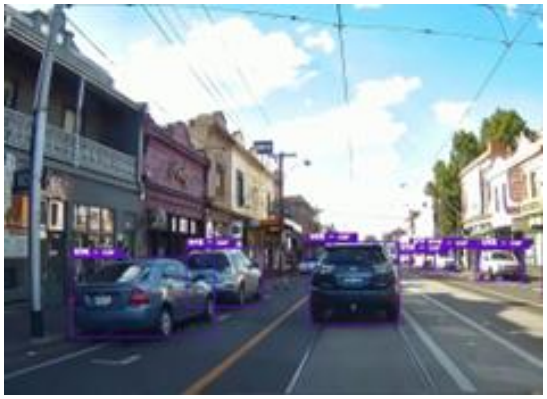
Fig. 6. YOLOv8 detection results under representative driving conditions, illustrating its performance in daytime, nighttime, and challenging environments

The detection performance of the YOLO-NAS model proposed in this paper is presented in Figure 7. This model introduces neural architecture search (NAS) to achieve automatic optimization of its network architecture. After testing across driving scenarios including nighttime settings, severe weather, and scenarios with partially occluded targets, the model is capable of identifying three core target types: vehicles, pedestrians, and traffic signs. Its full scenario detection accuracy reaches 94%–96%, and it

maintains a stable inference speed, which makes it suitable for safety critical applications such as autonomous driving. Even small improvements in perception accuracy can help boost the operational safety of vehicles.



(a)



(b)



(c)

Fig. 7. Detection results of the proposed YOLO-NAS model, showing improved robustness and accuracy across diverse driving scenarios.

Table 1 shows the comparison of detection rates for YOLOv7, YOLOv8 and YOLO-NAS. Models are tested under different conditions. The results validate YOLO-NAS as the superior and most friendly framework, gaining 5–10% improvement over YOLOv8 and 13–25% improvement over YOLOv7. The results demonstrate that incorporating dynamically optimized network architectures significantly enhances the perception capabilities of autonomous vehicles.

Table 1. Comparison of YOLO versions under varying conditions

YOLO-Version	(a)	(b)	(c)
YOLOv7	80%	81%	81%
YOLOv8	89%	84%	86%
YOLO-NAS	96%	94%	95%

ISSN 2710-1274 (Print) · 2074-8523 (Online) · Published by Al-Zaytoonah University of Jordan (ZUJ)

Table 2. Comparison of Object Detection Performance for YOLOv7, YOLOv8, and YOLO-NAS on the nuScenes

YOLO-Version	Recall	F1-Score	**mAp@50**	**mAp@50-95**
YOLOv7	88-92%	89-92%	90-93%	58-68%
YOLOv8	92-94%	90-93%	92-94%	63-72%
YOLO-NAS	93-96%	92-95%	93-96%	66-75%

Table 2 summarizes the quantitative comparison among the three evaluated detectors. YOLO-NAS consistently achieves the highest Precision, Recall, F1-score, mAP@50, and mAP@50–95 values, confirming its superior ability to detect and localize objects under diverse environmental conditions. The progressive improvements observed from YOLOv7 to YOLOv8 and finally to YOLO-NAS demonstrate the effectiveness of architecture optimization in improving perception robustness for autonomous vehicle systems.

4.3 Discussion

The YOLO-NAS model proposed in this study outperforms YOLOv7 and YOLOv8 in terms of performance across all evaluation scenarios. It achieves an accuracy of 94–96% under adverse weather conditions such as nighttime and rainy days. Integrating NAS into the YOLO framework enables architectural optimization and breaks through the inherent limitations of fixed network design.

The advantages of YOLO-NAS analyzed in this paper extend far beyond higher detection accuracy. This model can support the core functions of the autonomous driving perception layer, which are trajectory planning, obstacle avoidance, and speed adjustment. It can avoid the risks of false detection and missed detection. With improved detection capabilities for occluded targets and low-visibility targets, it can output stable perception results, enabling smoother and more reliable vehicle operation.

The core advantage of YOLO-NAS is its exceptional adaptability. Unlike same-series models YOLOv7 and YOLOv8, which adopt predefined network architectures, YOLO-NAS is optimized via automated architecture search. It can adapt to scenarios such as low light, dense traffic flow, and sensor noise, and improves detection performance in real-world autonomous driving scenarios.

The authors of this paper propose that more stable object detection for the perception systems of autonomous vehicles can reduce software recalibration and post-deployment corrections, simplify system validation across multiple operating conditions, enhance trust in the technology, and advance regulatory approval and commercial implementation.

First, we align the relevant conclusions of previous studies on YOLOv7 and YOLOv8, and sort out the performance degradation issues of the two models in occluded, nighttime, and severe weather scenarios. This study finds that YOLO-NAS can alleviate most of these problems, meets the standard inference speed requirement, and is well-suited to the real-time application demands of autonomous driving.

Although the YOLO-NAS proposed in this paper has prominent advantages over other lightweight YOLO variants, it still has two core limitations. First, it has relatively high resource requirements, making it difficult to adapt to embedded deployment scenarios with limited resources. Optimizations can be developed along directions such as hardware-software co-design, as outlined in reference [33]. Second, the range of validation scenarios for the model is limited, and additional testing in complex urban intersections and multi-sensor environments must be supplemented.

5 Conclusion

In order to improve autonomous driving systems' perceptual capabilities, this paper presents an object identification framework based on YOLO-NAS. The suggested framework allows the detector to maintain high processing speed while maintaining dependable performance under a variety of driving conditions by combining the real-time inference efficiency associated with the YOLO family with the architecture optimization offered by Neural Architecture Search (NAS). With findings ranging from 94% to 96%, experimental examination showed consistent detection performance across all investigated conditions. The suggested framework performed more robustly and reliably than YOLOv7 and YOLOv8, especially in situations with clutter, partial occlusion, or more complicated scenes. This work's methodological foundation is based on the strategy outlined in [5].

Reliable perception remains one of the fundamental requirements for autonomous driving because the quality of object detection directly influences subsequent planning and control decisions. Consistent real-time detection reduces the need for extensive software recalibration after deployment and simplifies system validation under practical operating conditions. Despite these advantages, the computational requirements of YOLO-NAS remain higher than those of lightweight YOLO variants, which may limit deployment on embedded platforms with constrained hardware resources. Addressing this limitation represents an important direction for future research. Model compression and related optimization strategies, such as those discussed in [34], offer a practical means of reducing computational overhead while preserving detection performance. In addition, evaluating the proposed framework on larger and more diverse datasets would provide a more comprehensive assessment of its ability to generalize across different driving environments and operational conditions.

DECLARATIONS

Funding. This research was funded by Al-Zaytoonah University of Jordan through the research project "Lightweight Post-Quantum Cryptography Framework for Securing Smart Traffic and National IoT-Based Critical Infrastructure", Grant No. 33/09/2025–2026.

Conflicts of interest. The authors declare that they have no competing interests.

Ethics approval. Ethics approval was not required for this study because it did not involve human participants, human data, animals, or biological materials.

Informed consent. Not applicable. This study did not involve human participants; therefore, informed consent was not required.

Data availability. The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Author contributions. Adi El-Dalahmeh conceived the study, designed the research methodology, developed the proposed framework, conducted the experiments, analyzed and interpreted the results, and prepared the original manuscript draft. Layla Albdour contributed to the methodology, validation, data interpretation, and critical revision of the manuscript. Mohammad Alia contributed to the study design, supervised the research, provided technical guidance and resources, and critically reviewed and revised the manuscript. All authors read, reviewed, and approved the final version of the manuscript prior to submission.

Use of artificial intelligence. The authors declare that no artificial intelligence (AI) or AI-assisted technologies were used in the conception, analysis, interpretation, writing, or preparation of this manuscript.

Acknowledgements. The authors would like to express their sincere gratitude to Al-Zaytoonah University of Jordan for supporting the research project entitled "Lightweight Post-Quantum Cryptography Framework for Securing Smart Traffic and National IoT-Based Critical Infrastructure" under Grant No. 33/09/2025–2026, and for providing the resources necessary to conduct this research.

References

[1] A. A. A. Alkhatib, K. Abu Maria, S. AlZu'bi, and E. Abu Maria, “Smart traffic scheduling for crowded cities road networks,” *Egyptian Informatics Journal*, vol. 23, no. 4, pp. 163–176, Dec. 2022, doi: 10.1016/j.eij.2022.10.002.

[2] A. El-Dalahmeh, M. El-Dalahmeh, and J. Li, “Enhanced vehicle detection through multi-sensor fusion utilizing YOLO-NAS and Faster R-CNN,” in Proc. 2024 Int. Joint Conf. Neural Networks (IJCNN), Yokohama, Japan, Jun.–Jul. 2024, pp. 1–6, doi: 10.1109/IJCNN60899.2024.10651024.

[3] .H. Jazyah, "5G Security, Challenges, Solutions, and Authentication," International Journal of Advances in Soft Computing &Its Applications, vol. 15, no. 3, pp. 54–68, Nov. 2023, doi: 10.15849/IJASCA.231130.04.

[4] A. El-Dalahmeh, M. El-Dalahmeh, and J. Li, "Integrating Elliptic Curve Encryption and Post-Quantum Cryptography for Secure VANETs Using Fog Computing and SDN," *IET Intelligent Transport Systems*, vol. 20, no. 1, p. e70214, 2026, doi: 10.1049/itr2.70214.

[5] R. Shadid, W. Shadid, and Y. Khawaja, "Recognizing Traffic Lights of Interest for Vehicle Driver Without Prior Information," International Journal of Advances in Soft Computing and its Applications, vol. 15, no. 2, pp. 236–251, Jul. 2023, doi: 10.15849/IJASCA.230720.14.

[6] A. Vijayakumar and S. Vairavasundaram, "YOLO-based object detection models: A review and its applications," *Multimedia Tools and Applications*, vol. 83, no. 35, pp. 83535–83574, 2024, doi: 10.1007/s11042-024-18872-y.

[7] J. E. Gallagher and E. J. Oughton, "Surveying You Only Look Once (YOLO) Multispectral Object Detection Advancements, Applications, and Challenges," IEEE Access, vol. 13, pp. 7366–7395, 2025, doi: 10.1109/ACCESS.2025.3526458.

[8] R. Ayachi, Y. Said, M. Afif, A. Alshammari, M. Hleili, and A. B. Abdelali, "Assessing YOLO models for real-time object detection in urban environments for advanced driver-assistance systems (ADAS)," Alexandria Engineering Journal, vol. 123, pp. 530–549, Jun. 2025, doi: 10.1016/j.aej.2025.03.077.

[9] V. K. Dileep, P. S. Athira, P. Jayakumar, and S. Praveena, "A Comprehensive Review of Smart Emergency Vehicle Detection and Response Systems," in *Proc. 2025 3rd Int. Conf. Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, Coimbatore, India, Apr. 2025, pp. 1–6, doi: 10.1109/ICAECA63854.2025.11012205.

[10] U. Dwivedi, K. Joshi, S. K. Shukla, and A. S. Rajawat, "An Overview of Moving Object Detection Using YOLO Deep Learning Models," in Proc. 2024 2nd Int. Conf. Disruptive Technologies (ICDT), Greater Noida, India, Mar. 2024, pp. 1014–1020, doi: 10.1109/ICDT61202.2024.10489800.

[11] S. Al Mokdad, M. A. Talib, S. Z. Ghebremeskel, F. Lamghari, S. B. Mirza, and A. B. Nassif, "An Overview of YOLO-Based Helmet Detection: Modifications and Advancements," in *Proc. 2024 8th Int. Conf. Imaging, Signal Processing and Communications (ICISPC)*, Kuala Lumpur, Malaysia, Jul. 2024, pp. 6–14, doi: 10.1109/ICISPC63824.2024.00009.

[12] A. R. Al-Ghamdi, "Revolutionizing Collision Avoidance Using Smart Vehicle Networks," International Journal of Advances in Soft Computing &Its Applications, vol. 15, no. 3, pp. 254–273, Nov. 2023, doi: 10.15849/IJASCA.231130.17.

[13] M. A. R. Alif and M. Hussain, "YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain," arXiv preprint arXiv:2406.10139, 2024.

[14] M. Mao and M. Hong, "YOLO Object Detection for Real-Time Fabric Defect Inspection in the Textile Industry: A Review of YOLOv1 to YOLOv11," Sensors, vol. 25, no. 7, Art. no. 2270, Apr. 2025, doi: 10.3390/s25072270.

[15] A. R. Khekan, H. S. Aghdasi, and P. Salehpour, "The Impact of YOLO Algorithms Within Fall Detection Application: A Review," IEEE Access, vol. 13, pp. 6793–6809, 2025, doi: 10.1109/ACCESS.2024.3496823.

[16] S. Kang, Z. Hu, L. Liu, K. Zhang, and Z. Cao, "Object Detection YOLO Algorithms and Their Industrial Applications: Overview and Comparative Analysis," Electronics, vol. 14, no. 6, Art. no. 1104, Mar. 2025, doi: 10.3390/electronics14061104.

[17] P. Thakor and B. Dixit, "Advancements in Pedestrian and Vehicle Detection: A Comprehensive Review of Deep Learning Techniques," in Proc. 2025 IEEE Int. Conf. Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), vol. 3, Mar. 2025, pp. 1–6, doi: 10.1109/IATMSI64286.2025.10985560.

[18] L. Liang, H. Ma, L. Zhao, X. Xie, C. Hua, M. Zhang, and Y. Zhang, "Vehicle detection algorithms for autonomous driving: A review," Sensors, vol. 24, no. 10, p. 3088, 2024.

[19] M. Flores-Calero, C. A. Astudillo, D. Guevara, J. Maza, B. S. Lita, B. Defaz, J. S. Ante, D. Zabala-Blanco, and J. M. Armingol Moreno, "Traffic Sign Detection and Recognition Using YOLO Object Detection Algorithm: A Systematic Review," Mathematics, vol. 12, no. 2, Art. no. 297, Jan. 2024, doi: 10.3390/math12020297.

[20] M. Majumder and C. Wilmot, "Automated Vehicle Counting from Pre-Recorded Video Using You Only Look Once (YOLO) Object Detection Model," J. Imaging, vol. 9, no. 7, Art. no. 131, Jul. 2023, doi: 10.3390/jimaging9070131.

[21] A. Benjumea, I. Teeti, F. Cuzzolin, and A. Bradley, "YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles," arXiv preprint arXiv:2112.11798, 2021.

[22] M. A. Rouf, Q. Wu, X. Yu, Y. Iwahori, H. Wu, and A. Wang, "Real-Time Vehicle Detection, Tracking and Counting System Based on YOLOv7," Embedded Selforganising Systems, vol. 10, no. 7, pp. 4–8, Jul. 2023, doi: 10.14464/ess.v10i7.598.

[23] M. Bie, Y. Liu, G. Li, J. Hong, and J. Li, "Real-Time Vehicle Detection Algorithm Based on a Lightweight You-Only-Look-Once (YOLOv5n-L) Approach," Expert Syst. Appl., vol. 213, Art. no. 119108, Mar. 2023, doi: 10.1016/j.eswa.2022.119108.

[24] M. Al Rawajbeh and H. Al-Zoubi, “Implementing the concept of smart cities in Amman: Reality and aspirations review,” in Proceedings of the 2024 IEEE/ACS 21st International Conference on Computer Systems and Applications (AICCSA), Oct. 2024, pp. 1–7. IEEE.

[25] A. A. A. Alkhatib, A. A. Hnaif, and T. Sawalha, “A New System for Road Traffic Optimisation Using the Virtual Traffic Light Technology,” Comput. Syst. Sci. Eng., vol. 47, no. 1, pp. 637–656, 2023, doi: 10.32604/CSSE.2023.037345.

[26] A. S. Geetha, M. A. R. Alif, M. Hussain, and P. Allen, “Comparative analysis of YOLOv8 and YOLOv10 in vehicle detection: Performance metrics and model efficacy,” Vehicles, vol. 6, no. 3, pp. 1364–1382, 2024.

[27] R. Sharma, P. Sharma, S. Hariharan, and S. Mahajan, “Improving public security: Application of YOLOv7 for vehicle detection,” in Proceedings of the 2024 International Conference on Computational Intelligence and Computing Applications (ICCICA), vol. 1, May 2024, pp. 43–46. IEEE.

[28] Nikarika ,V. Kukreja, N. Thapliyal, M. Aeri, and R. Sharma, “An advanced YOLOv7 model for vehicle detection to enhance the security of public places,” in Proceedings of the 2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS), Jun. 2024, pp. 1–5. IEEE.

[29] J. B. G. SV, J. G. Ponsam, and D. B. PS, “A study on fast detection and evaluating YOLOv5, YOLOv6, YOLOv7 and YOLOv8,” in Proceedings of the 2025 International Conference on Computing and Communication Technologies (ICCCT), Apr. 2025, pp. 1–5. IEEE.

[30] H. Azis, M. Abdullah, S. Ismail, and L. Syafie, “A comparative study of YOLO models for enhanced vehicle detection in complex aerial scenarios,” in Proceedings of the 2025 19th International Conference on Ubiquitous Information Management and Communication (IMCOM), Jan. 2025, pp. 1–8. IEEE.

[31] C. J. Lin and C. M. Lee, “Real-time vehicles detection with YOLOv8,” in Proceedings of the 2024 International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan), Jul. 2024, pp. 805–806. IEEE.

[32] Z. Liu and Z. Zhang, “The research on an improved YOLOX-based algorithm for small-object road vehicle detection,” Electronics, vol. 14, no. 11, p. 2179, 2025.

[33] N. I. M. Yusof, A. Sophian, H. F. M. Zaki, A. A. Bawono, A. H. Embong, and A. Ashraf, “Assessing the performance of YOLOv5, YOLOv6, and YOLOv7 in road defect detection and classification: a comparative study,” Bulletin of Electrical Engineering and Informatics, vol. 13, no. 1, pp. 350–360, 2024.

[34] M. J. Alam, M. Z. Hasan, M. M. Rahman, M. A. Rahman, N. H. Sarker, S. Azad, T. N. Islam, B. Paul, T. Anjum, and B. Halder, “An optimized YOLOv5 based approach for real-time vehicle detection at road intersections using fisheye cameras,” arXiv preprint arXiv:2502.04566, 2025.

[35] A. S. Geetha, “Comparing YOLOv5 variants for vehicle detection: A performance analysis,” arXiv preprint arXiv:2408.12550, 2024.