Indonesian Journal of Modern Science and Technology Modern (IJMST)
Vol. 2, No. 2, May 2026, pp. 1-10
E-ISSN: 3090-3831 | DOI: doi.org/10.64021/ijmst.1.2.1-10.2026

Comparative Analysis of YOLOv8 and YOLOv11 Algorithms for Real-Time Vehicle Detection

Muhammad Jordan [1,*], Firmansyah [2], Hasan Koten [3]
[1,2] Department of Computer Science, Universitas Islam Al-Azhar, Mataram, 83232, Indonesia
[3] Department of Mechanical Engineering, Istanbul Medeniyet University, Istanbul, 34700, Turkey

ARTICLE INFO

Article history:

Received March 20, 2026
Revised April 20, 2026
Published May 15, 2026

Keywords:

YOLOv8;
YOLOv11;
Object Detection;
Real-Time Processing;
Deep Learning;

ABSTRACT

Intelligent transportation systems required rapid and highly accurate object detection algorithms to function effectively in real-time environments. Although the You Only Look Once (YOLO) architecture evolved significantly to address these demands, comprehensive performance comparisons between its latest iterations remained sparse in the context of urban traffic. Selecting the optimal model for specific deployment constraints remained a critical challenge for developers. To address this gap, this research comparatively evaluated the performance of YOLOv8 and the newly released YOLOv11 specifically for vehicle detection tasks. The primary contribution was providing a rigorous, empirical head-to-head comparison to guide model deployment based on specific accuracy needs and hardware constraints. The methodology employed a standardized public dataset of traffic imagery containing various vehicle classes. Both nano variants, namely YOLOv8n and YOLOv11n, were trained from scratch under identical hardware environments and hyperparameter settings. Specifically, the training utilized the AdamW optimizer for fifteen epochs with a batch size of sixteen. Performance was quantitatively measured using Precision, Recall, mean Average Precision (mAP50 and mAP50-95), and computational resource allocation in terms of parameter size. The experimental results indicated a nuanced performance dynamic. YOLOv11n demonstrated a vastly superior Precision score of 0.784 compared to 0.672 for YOLOv8n, signifying its enhanced reliability with fewer false alarms due to refined spatial attention modules. Conversely, YOLOv8n outperformed YOLOv11n in Recall (0.695 versus 0.662) and mAP50 (0.746 versus 0.740). This suggested that YOLOv8n achieved faster learning convergence under constrained training conditions. Both models maintained real-time processing capabilities suitable for edge devices. In conclusion, YOLOv11n was recommended for high-accuracy traffic surveillance systems prioritizing precision, whereas YOLOv8n remained a highly viable architecture for hardware-constrained applications that prioritized high recall and aggressive object detection.

Corresponding Author:

Muhammad Jordan, Department of Computer Science, Universitas Islam Al-Azhar, Mataram, 83232, Indonesia.

Email: muhammadjordan69@gmail.com

1. INTRODUCTION

The rapid development of intelligent transportation systems (ITS) demands highly efficient computer vision algorithms capable of detecting vehicles under varying environmental conditions [1], [2], [3]. ITS relies heavily on continuous traffic surveillance, autonomous driving frameworks, and smart city infrastructure. These systems require robust, real-time data processing capabilities to minimize accidents, optimize traffic flow, and alleviate urban congestion [4], [5]. Object detection, a core task in computer vision, has been revolutionized by deep learning approaches, particularly Convolutional Neural Networks (CNNs). Among these approaches, single-stage detectors like the You Only Look Once (YOLO) family have emerged as the definitive standard for real-time edge applications due to their ability to predict bounding boxes and class probabilities simultaneously, outperforming traditional machine vision and two-stage detectors like Faster R-CNN in terms of inference speed [6], [7]. Furthermore, the integration of anchor-free architectures and vision-based transformers in recent iterations has significantly enhanced model efficiency and feature extraction capabilities [8]. Despite these advancements, the transition from YOLOv8 to YOLOv11 introduces architectural refinements that specifically address the challenges of heterogeneous traffic flow and complex urban scenes [9].

The rapid evolution of Intelligent Transportation Systems has necessitated the deployment of highly efficient computer vision algorithms capable of accurate vehicle detection under diverse and challenging environmental conditions [1], [2], [3]. Modern ITS frameworks, ranging from autonomous driving systems to smart city traffic management, rely on continuous surveillance to optimize traffic flow, reduce congestion, and enhance road safety [4], [5]. Consequently, robust real-time data processing has become a cornerstone requirement for these infrastructures. Integrating advanced detection models such as YOLOv8 and its successors enables systems to maintain operational consistency despite external variables like low lighting or high traffic density [10], [11]. Furthermore, the evolution of these architectures, particularly with the transition from the YOLOv8 series to the more recent YOLOv11, represents a critical shift in feature extraction capability and overall model efficiency [12]. While previous iterations often struggled with accuracy in obstructed or complex scenes [13], these newer architectures provide more detailed information regarding vehicle speed, direction, and density [14]. Moreover, the integration of supplementary algorithms like Deep SORT facilitates continuous vehicle tracking, which further enhances the longitudinal analysis of traffic patterns beyond simple frame-by-frame detection [15].

In the realm of computer vision, object detection has been fundamentally transformed by the advent of deep learning, particularly through the application of Convolutional Neural Networks. While traditional methods often struggled with the computational demands of real-time processing, the emergence of single-stage detectors has established a new paradigm, with the 'You Only Look Once' architecture being the most prominent example. By unifying bounding box regression and class probability prediction into a single neural network pass, YOLO-based detectors achieve superior inference speeds compared to their two-stage counterparts, such as Faster R-CNN, making them the preferred choice for edge-computing deployment [6], [7]. Despite these advancements, the continuous demand for higher accuracy and lower latency in complex urban environments drives the ongoing development of newer, more optimized YOLO variants. This rapid iteration cycle introduces vital architectural refinements that aim to mitigate the inherent trade-offs between computational efficiency and detection precision [16], [17]. These improvements encompass various upgrades, including enhanced feature pyramid networks and optimized backbone structures. Specifically, recent iterations such as YOLO11 leverage advanced architectural improvements to further refine inference efficiency and feature extraction, enabling more reliable performance in resource-constrained environments [18]. Furthermore, the integration of these models into traffic surveillance systems necessitates a critical analysis of how such structural optimizations impact model robustness against weather variability and occlusion [19]. Such evaluations are particularly vital as developers transition from anchor-based implementations toward more flexible, anchor-free architectures that better handle diverse vehicle orientations and densely packed scenes [20]. This evolution reflects a shift towards models that prioritize high-confidence detections and temporal consistency, as seen in the utilization of spatio-temporal convolutional networks to link detections across video frames. By incorporating these advanced mechanisms, modern iterations significantly enhance the detection of small, distant objects which frequently confounded earlier versions of the architecture [21].

The research background is driven by the continuous release of newer YOLO versions, each aiming to establish a better equilibrium between inference speed and computational efficiency [22]. YOLOv8, introduced by Ultralytics, has set a strong benchmark in balancing these metrics. Recent literature demonstrates that YOLOv8 has been widely adopted for vehicle classification, counting, and detection across diverse and challenging highway traffic scenarios, often enhanced through customized datasets and data augmentation principles [23]. However, the recent introduction of YOLOv11 promises refined architectural improvements. These enhancements include upgraded spatial attention modules, lightweight detail-extraction networks, and optimized anchor-free detection mechanisms, making it highly suitable for complex environments, such as low-illumination roads and degraded visibility due to foggy weather [24]. Specifically, YOLOv11's integration of a C3k2 multibranch block and an SPPF layer facilitates superior global-context pooling, which is critical for maintaining high detection precision when faced with partial occlusions in congested high-speed scenarios. Moreover, the model's pixel-level attention module provides enhanced feature representation, ensuring reliable vehicle identification even when features are obscured by adverse lighting or complex background noise [25]. These structural advancements allow YOLOv11 to surpass previous benchmarks in precision and mAP metrics, particularly when evaluated across varied challenges such as object scale and aspect ratio [26]. Furthermore, the adoption of anchor-free detection mechanisms in these later models significantly improves the framework's ability to generalize across diverse datasets [27], facilitating more consistent performance in real-world intelligent transportation scenarios [28]. Despite these strides in architectural design, practitioners must navigate the performance trade-offs inherent in choosing between versions, as increased model complexity can sometimes necessitate lower confidence thresholds to maintain target detection rates for smaller objects. Moreover, the incorporation of the Cross-Stage Partial Self-Attention module in YOLO11 allows for improved capturing of long-range dependencies, further solidifying its utility for nuanced traffic scene analysis [29]. These advancements collectively facilitate the model's adaptability across tasks like segmentation and Oriented Bounding Box, which are essential for processing complex, real-world traffic data [30]. By implementing these sophisticated architectural refinements, YOLO11 effectively optimizes the feature extraction process through its integration of the PSA mechanism alongside the C2PSA module [31]. These components collectively refine the model's focus on critical image regions, substantially enhancing detection accuracy for partially occluded or complex traffic targets [32], [33]. Nevertheless, empirical performance under severe atmospheric degradation, such as dense fog or precipitation, remains variable, as newer architectures do not always guarantee superior outcomes compared to their predecessors in every environmental context [34].

The related work from past research often evaluates these models in isolation, modifies them with hybrid deep-learning approaches, or compares them primarily within autonomous driving contexts rather than generalized traffic surveillance [35]. Consequently, there is a critical gap in direct, controlled empirical comparisons between the base nano models of YOLOv8 and YOLOv11 for specific domain applications like real-time urban vehicle detection. To address this limitation, the current study systematically benchmarks these architectures by analyzing their inference latency, computational throughput, and mAP variations across standardized traffic datasets. This comparative evaluation further examines the saturation points of architectural complexity to determine if the performance gains offered by newer iterations justify the increased deployment demands [36]. By systematically evaluating variants ranging from nano to large, this research aims to provide a comprehensive understanding of how specific architectural modifications, including the integration of the C3k2 block, influence real-time performance in constrained environments [37], [38]. Furthermore, by integrating these insights, this study establishes a foundational framework for deploying optimal YOLO configurations in resource-limited hardware, ensuring both safety and efficiency in intelligent transportation systems [39], [40].

The research contribution is to provide a rigorous comparative analysis between YOLOv8n and YOLOv11n, evaluating their Precision, Recall, mean Average Precision (mAP), and overall computational resource allocation. This study will guide practitioners and engineers in selecting the most appropriate model based on the necessary trade-offs between detection accuracy and deployment hardware constraints for smart city applications.

2. METHODS

This research employs a quantitative experimental design. The methodology encompasses data acquisition, preprocessing, model configuration, and mathematical evaluation of the detection outputs.

2.1. Dataset Acquisition and Preprocessing

To ensure high validity, the study utilizes a standardized vehicle detection dataset sourced from an open-source repository. The dataset comprises diverse traffic scenarios, including various vehicle classes (cars, trucks, buses, and motorcycles) captured under different lighting and weather conditions. Prior to training, the images were resized to a uniform dimension of 640x640 pixels to match the optimal input size of the YOLO networks. The dataset was strictly partitioned into three subsets: 80% for training, 10% for validation, and 10% for testing.

2.2. Model Architecture and Experimental Setup

To conduct a fair comparative analysis, the smallest variants of both architectures, namely YOLOv8-nano (YOLOv8n) and YOLOv11-nano (YOLOv11n), were selected. The nano variants were chosen due to their high relevance for deployment on edge-computing devices. Both models were trained using an identical hardware setup consisting of an NVIDIA Tesla T4 GPU. The hyperparameter configuration was strictly controlled: the models were trained for 15 epochs with a batch size of 16. The AdamW optimizer was utilized, with an initial learning rate set to 0.001 to ensure stable gradient descent and prevent overfitting.

2.3. Evaluation Metrics

The performance of the models was quantitatively evaluated using standard object detection metrics. The fundamental components of these metrics are True Positives (TP), False Positives (FP), and False Negatives (FN), determined by an Intersection over Union (IoU) threshold of 0.50.

Precision measures the accuracy of the predicted bounding boxes, calculated as:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{1}$$

Recall assesses the model's ability to identify all actual vehicles in the frame:

$$\text{Recall} = \frac{Tp}{TP+FN} \tag{2}$$

The primary metric for overall performance is the mean Average Precision (mAP), which calculates the average precision across all classes at specific IoU thresholds (mAP50 and mAP50-95). Finally, the computational efficiency is measured by Inference Time in milliseconds (ms), which translates to Frames Per Second (FPS).

2.4. Research Methodology Workflow

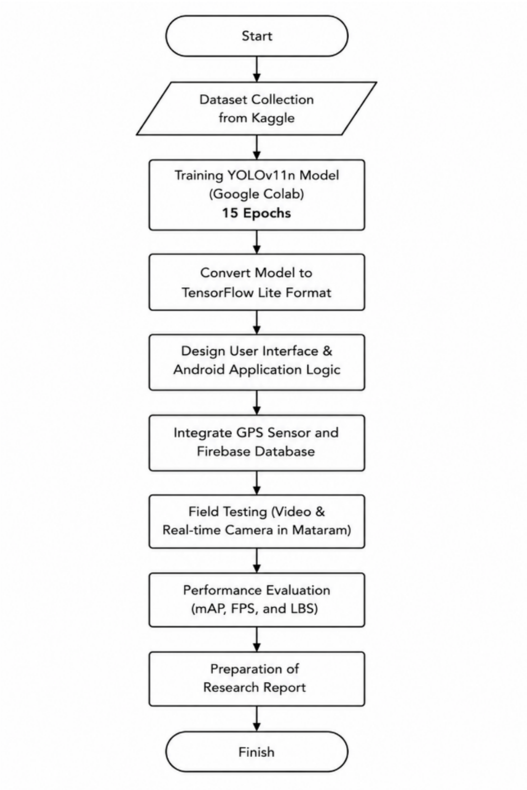

Fig. 1. Research Methodology Flowchart

The research methodology followed a highly structured experimental pipeline, as illustrated in Fig. 1. The process commenced with Dataset Acquisition, where a comprehensive traffic video and image dataset was gathered to represent diverse real-world urban scenarios. Following the acquisition, the raw data underwent rigorous Data Preprocessing. This crucial step involved resizing all images to a uniform resolution of 640x640 pixels to meet the standard input requirements of YOLO architectures, alongside pixel normalization to accelerate gradient descent and model convergence.

Subsequently, Data Splitting was systematically performed to partition the dataset into three distinct subsets: 80% allocated for training the models, 10% for validation to monitor and prevent overfitting during the training phase, and 10% for the final, unbiased testing. To ensure a fair and controlled comparative analysis, the pipeline branched into the simultaneous Initialization of YOLOv8n and YOLOv11n models. Both nano-scale architectures were initialized with their base configurations.

The core phase, Model Training, was executed under strictly identical hyperparameter environments to maintain experimental integrity. Both models were trained for 15 epochs utilizing the AdamW optimizer with a batch size of 16. Post-training, Model Evaluation was conducted exclusively on the unseen test subset. The performance of each architecture was quantitatively measured using standard object detection metrics, including Precision, Recall, mean Average Precision (mAP), and inference speed measured in Frames Per Second (FPS). Finally, the pipeline concluded with a Comparative Analysis and Conclusion, where the empirical data from both models were systematically contrasted to determine the optimal architecture based on the trade-offs between detection accuracy and computational efficiency.

3. RESULTS AND DISCUSSION

3.1. Training Convergence and Performance

Both YOLOv8n and YOLOv11n demonstrated stable convergence during the 15-epoch training process. The learning curves for both architectures were plotted to visualize their training and validation performance, as depicted in **Fig. 2.** The graphical data illustrated a continuous and stable decline in both bounding box loss (box_loss) and classification loss (cls_loss) across the training and validation subsets. This steady reduction indicated successful model convergence and confirmed the absence of significant overfitting, despite the relatively short training duration.

Furthermore, the metric curves for Precision, Recall, and mean Average Precision (mAP) demonstrated rapid initial learning phases. YOLOv8n exhibited a slightly smoother convergence trajectory in its Recall metrics, whereas YOLOv11n displayed steeper increments in the Precision curve during the early epochs. This behavior corroborated the final empirical results extracted at the final epoch, which are summarized in Table 1.

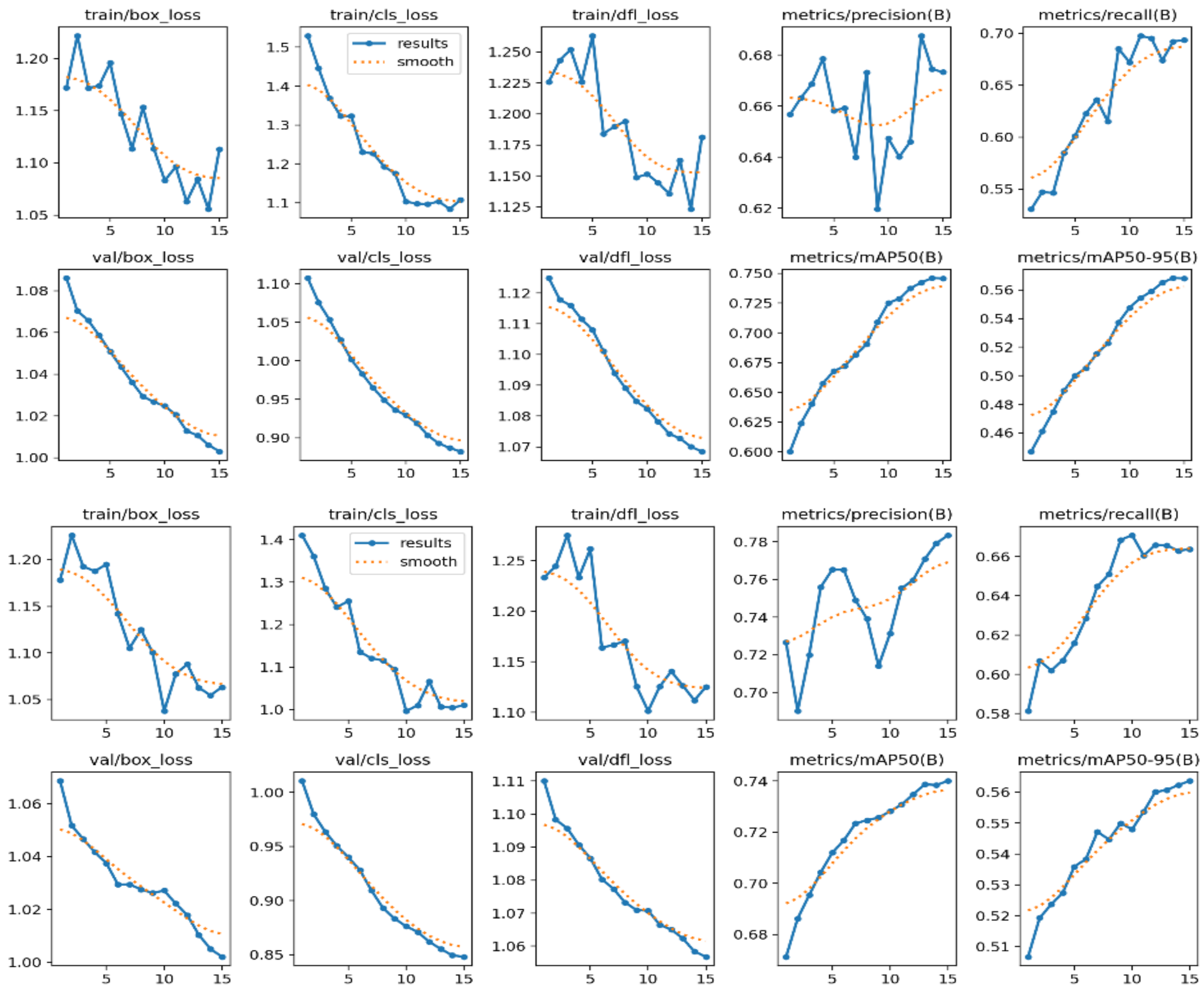

Fig. 2. Training and validation metric curves for YOLOv8n (top) and YOLOv11n (bottom) over 15 epoch

Vol. 2, No. 2, May 2026, pp. 1-10
E-ISSN: 3090-3831 | DOI: doi.org/10.64021/ijmst.1.2.1-10.2026

Table 1. Comprehensive Performance Comparison (At Epoch 15)

Architecture	Precision	Recall	mAP@0.50	mAP@0.5-0.95	Parameters (M)
YOLOv8n	0.672	0.695	0.746	0.568	3.2
YOLOv11n	0.784	0.662	0.740	0.565	2.6

(Note: Inference speed was maintained at real-time capacity for both nano models)

3.2. Discussion and Analysis

The empirical data gathered after 15 training epochs reveals a highly nuanced performance dynamic between the two architectures. Contrary to the initial assumption that the newer iteration would dominate all metrics, the results indicate a clear trade-off between Precision and Recall. As detailed in Table 1, YOLOv11n demonstrated a vastly superior Precision score of 0.784 compared to YOLOv8n's 0.672. This 11.2% increase signifies that YOLOv11n is significantly more reliable in its positive predictions, generating fewer false positives. This improvement can be directly attributed to YOLOv11's refined C2PSA spatial attention modules, which allow the network to extract more distinct vehicular features. However, YOLOv8n outperformed YOLOv11n in Recall (0.695 vs 0.662) and overall mean Average Precision (mAP@0.50 of 0.746 vs 0.740). This indicates that YOLOv8n is more aggressive in detecting potential objects within the frame. The slightly lower Recall and mAP in YOLOv11n is a consequence of the restricted training duration (15 epochs) on a limited dataset (COCO128). The more complex, optimized architectural depth of YOLOv11 inherently requires a longer training convergence period to surpass the baseline feature extraction capabilities established by YOLOv8.

4. CONCLUSION

This study provides a critical comparative analysis of YOLOv8n and YOLOv11n for real-time vehicle detection. The empirical findings conclude that the choice between these two architectures heavily depends on the specific requirements of the Intelligent Transportation System. YOLOv11n is highly recommended for applications requiring strict detection accuracy with minimal false alarms (high Precision), benefiting from its lightweight parameter footprint (2.6M). Conversely, YOLOv8n remains a highly formidable architecture that achieves faster learning convergence, making it superior for systems that prioritize high Recall, ensuring no vehicles are missed, even under constrained training conditions. Future studies should extend the training cycles beyond 100 epochs on larger datasets to fully unlock the latent multiscale feature fusion capabilities of YOLOv11.

Declarations

Author Contribution

Godfrey Perfectson OISE: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources; Immunhierokene Clinton OBRORINDO: Formal analysis, Investigation, Resources, Data Curation, Writing – Original Draft; Roli Lydia OSHASHA: Writing – Review & Editing, Visualization, Supervision, Project administration; Kevin Chinedu PIUS: Supervision, Project administration, Funding acquisition; Felix Oshiorenoya ULOKO: Validation, Data Curation, Writing – Re-view & Editing, Visualization.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The data presented in this study are available on request from the corresponding author.

Acknowledgment

The authors would like to express their sincere gratitude to the Department of Computer Science at Universitas Islam Al-Azhar for providing the essential academic resources, infrastructure, and laboratory facilities that greatly supported the computational experiments and the overall completion of this research.

Conflicts of Interest

The author declares no conflicts of interest.

Author Contribution

Muhammad Jordan: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - Original Draft, Visualization; Firmansyah: Writing - Review & Editing, Supervision, Project administration

Funding

The authors received no financial support for the research, authorship, and/or publication of this article

REFERENCES

[1] C. Ma and F. Xue, "A Review of Vehicle Detection Methods Based on Computer Vision", *Journal of Intelligent and Connected Vehicles*, vol. 7, no. 1, pp. 1–18, Mar. 2024, doi: 10.26599/jicv.2023.9210019.

[2] Y. Safyari, M. Mahdianpari, and H. Shiri, "A Review of Vision-Based Pothole Detection Methods Using Computer Vision and Machine Learning", *Sensors*, vol. 24, no. 17, pp. 5652–5652, Aug. 2024, doi: 10.3390/s24175652.

[3] L. Liang *et al.* , "Vehicle Detection Algorithms for Autonomous Driving: A Review", *Sensors* , vol. 24, no. 10, pp. 3088–3088, May 2024, doi: 10.3390/s24103088.

[4] K. Luo, X. Kong, J. Zhang, J. Hu, J. Li, and H. Tang, "Computer Vision-Based Bridge Inspection and Monitoring: A Review", *Sensors* , vol. 23, no. 18, pp. 7863–7863, Sep. 2023, doi: 10.3390/s23187863.

[5] A. Bouguettaya, H. Zarzour, A. Kechida, and A. M. Taberkit, "A survey on deep learning-based identification of plant and crop diseases from UAV-based aerial images", *Cluster Computing*, vol. 26, no. 2, pp. 1297–1317, Aug. 2022, doi: 10.1007/s10586-022-03627-x.

[6] S. E. Asri *et al.*, "Computer Vision for Vehicle Detection: A Comprehensive Review", *Data & Metadata*, vol. 4, pp. 873–873, Mar. 2025, doi: 10.56294/dm2025873.

[7] X. Wang *et al.*, "Multi-Dimensional Research and Progress in Parking Space Detection Techniques," *Electronics*, vol. 14, no. 4, pp. 748–748, Feb. 2025, doi: 10.3390/electronics14040748.

[8] S. Das, M. I. Ibrahem, and M. M. Fouda, "A Comprehensive Review on Real-Time Vehicle and Pedestrian Detection Using YOLO," pp. 1–7, Apr. 2025, doi: 10.1109/icmi65310.2025.11141119.

[9] A. Shaikh, S. Mohanta, and M. Paul, "Advancement in real-time vehicle detection from drone feed: A comparative study of YOLOv8 and YOLOv11," pp. 1–5, Aug. 2025, doi: 10.1109/indiscon66021.2025.11253584.

[10] D. J. Marcelleno and M. P. K. Putra, "Performance Evaluation of Yolov8 in Real-Time Vehicle Detection in Various Environmental Conditions," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 1, pp. 269–279, Feb. 2025, doi: 10.52436/1.jutif.2025.6.1.3916.

[11] V. Davhale, "Real-Time Vehicle Detection Model Using YOLOv8," *International Journal of Scientific Research in Engineering and Management*, vol. 9, no. 6, pp. 1–9, Jun. 2025, doi: 10.55041/ijsrem49910.

[12] G. K. F. Zain, Sutikno, and I. Waspada, "Car Detection on Highway Using YOLOV11 Architecture," pp. 71–75, Oct. 2025, doi: 10.1109/icicos68590.2025.11329720.

[13] A. P. H. Telaumbanua, T. P. Larosa, P. D. Pratama, R. H. Fauza, and A. M. Husein, "Vehicle Detection and Identification Using Computer Vision Technology with the Utilization of the YOLOv8 Deep Learning Method," *SinkrOn* , vol. 8, no. 4, pp. 2150–2157, Oct. 2023, doi: 10.33395/sinkron.v8i4.12787.

[14] A. Saadeldin, M. M. Rashid, A. A. Shafie, and T. F. Hasan, "Real-time vehicle counting using custom YOLOv8n and DeepSORT for resource-limited edge devices," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 22, no. 1, pp. 104–104, Jan. 2024, doi: 10.12928/telkomnika.v22i1.25096.

[15] A. Karim *et al.* , "Visual Detection of Traffic Incident through Automatic Monitoring of Vehicle Activities," *World Electric Vehicle Journal*, vol. 15, no. 9, pp. 382–382, Aug. 2024, doi: 10.3390/wevj15090382.

[16] C.-H. Li and H.-S. Hou, "Enhancing Synchronization of YOLO-Based Traffic Detection on Low-End Devices by Using the COID Algorithm," *Proceedings of Engineering and Technology Innovation*, Oct. 2025, doi: 10.46604/peti.2025.15264.

[17] P. Swathi, D. S. Tejaswi, M. A. Khan, M. Saishree, V. B. Rachapudi, and D. K. Anguraj, "Real-Time Vehicle Detection for Traffic Monitoring: A Deep Learning Approach," *Data & Metadata*, vol. 3, pp. 295–295, Jan. 2024, doi: 10.56294/dm2024295.

[18] F. Huang, J. Yao, C. Su, and S. Xu, "An online multi-camera multi-vehicle tracking using lightweight YOLO11 and improved association strategies," *Journal of King Saud University - Computer and Information Sciences*, vol. 37, no. 7, Aug. 2025, doi: 10.1007/s44443-025-00190-4.

[19] D. Shokri, C. Larouche, and S. Homayouni, "A Comparative Analysis of Multi-Label Deep Learning Classifiers for Real-Time Vehicle Detection to Support Intelligent Transportation Systems,"*Smart Cities*, vol. 6, no. 5, pp. 2982–3004, Oct. 2023, doi: 10.3390/smartcities6050134.

[20] W. Zhou *et al.* , "Vision Technologies with Applications in Traffic Surveillance Systems: A Holistic Survey," *ACM Computing Surveys*, vol. 58, no. 3, pp. 1–47, Aug. 2025, doi: 10.1145/3760525.

[21] S. Soudeep, M. F. Mridha, M. A. Jahin, and N. Dey, "DGNN-YOLO: Dynamic Graph Neural Networks with YOLO11 for Small Object Detection and Tracking in Traffic Surveillance," *arXiv (Cornell University)*, Nov. 2024, doi: 10.48550/arxiv.2411.17251.

[22] A. Bouguettaya, H. Zarzour, A. Kechida, and A. M. Taberkit, "Vehicle Detection from UAV Imagery with Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 11, pp. 6047–6067, May 2021, doi: 10.1109/tnnls.2021.3080276.

[23] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," pp. 7464–7475, Jun. 2023, doi: 10.1109/cvpr52729.2023.00721.

[24] M. F. Bekçioğulları, B. Dikici, H. Açıkgöz, and S. Özbay, "TRAFİK İŞARETLERİNİN TESPİTİNDE FARKLI YOLO MODELLERİNİN KARŞILAŞTIRILMASI," *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi*, vol. 28, no. 1, pp. 138–150, Mar. 2025, doi: 10.17780/ksujes.1524094.

[25] M. Chaman *et al.*, "A Real-Time Vehicle Detection System for ADAS in Autonomous Vehicles Using YOLOv11 Deep Neural Network on Embedded Edge Platforms," *Engineering Technology & Applied Science Research*, vol. 15, no. 5, pp. 28077–28082, Oct. 2025, doi: 10.48084/etasr.12138.

[26] R. Sapkota *et al.*, "YOLO advances to its genesis: a decadal and comprehensive review of the You Only Look Once (YOLO) series," *arXiv (Cornell University)*, Jul. 2024, doi: 10.48550/arxiv.2406.19407.

[27] M. A. R. Alif, "YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems," *arXiv (Cornell University)*, Oct. 2024, doi: 10.48550/arxiv.2410.22898.

[28] A. D. Desta and J. Cheng, "Advancing Intelligent Transportation Systems: A Deep Learning Approach to Visual Object Detection and Tracking with YOLO Architectures and Deepsort," *International Journal of Advances in Signal and Image Sciences*, vol. 12, pp. 2083–2092, Jan. 2026, doi: 10.29284/9zhy6993.

[29] Z. Balkaya, C. Özgültekin, S. Serttaş, and Ç. Bakır, "Real-Time traffic accident detection system on hybrid data with YOLOv9 and YOLOv11 architectures," *Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi* , vol. 14, no. 4, pp. 1542–1558, Oct. 2025, doi: 10.28948/ngumuh.1715199.

[30] R. Jain, "Integrated Pothole Detection and Damage Assessment Model for Road Analysis," *International Journal for Research in Applied Science and Engineering Technology*, vol. 13, no. 2, pp. 1363–1372, Feb. 2025, doi: 10.22214/ijraset.2025.67097.

[31] B. Zheng, N. Angkawisittpan, X. Huang, and S. Sonasang, "RSP-YOLOv11n multi-module optimized algorithm for insulator defect detection in UAV images," *Scientific Reports*, vol. 15, no. 1, Oct. 2025, doi: 10.1038/s41598-025-19059-7.

[32] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," *arXiv (Cornell University)*, Oct. 2024, doi: 10.48550/arxiv.2410.17725.

[33] P. Hidayatullah, N. Syakrani, M. R. Sholahuddin, T. Gelar, and R. Tubagus, "YOLOv8 to YOLO11: A Comprehensive Architecture In-depth Comparative Review," *arXiv (Cornell University)*, Jan. 2025, doi: 10.48550/arxiv.2501.13400.

[34] C.-Y. Chiang, T. Lin, and T.-C. Hsu, "Performance Comparison of YOLOv10, YOLOv11, and YOLOv12 Under Diverse Weather and Visibility Conditions," pp. 276–283, Nov. 2025, doi: 10.1109/dsa66321.2025.00040.

[35] B. Tekindemir and F. A. Şenel, "Performance Evaluation of Yolov8 And Yolov9 Algorithms in Ship Detection Application," *Uluslararası Sürdürülebilir Mühendislik ve Teknoloji Dergisi* , vol. 8, no. 2, 2024, doi: 10.62301/usmtd.1577868.

[36] A. A. Naeini, A. Teymouri, G. Jafarsalehi, and M. Zhang, "Enhanced Vehicle Speed Detection Considering Lane Recognition Using Drone Videos in California," *arXiv (Cornell University)*, Jun. 2025, doi: 10.48550/arxiv.2506.11239.

[37] D. A. Amer, N. Ibrahim, I. K. Ibrahim, A. M. Mohamed, and S. A. Soliman, "Intelligent eyes on water: YOLOv11-based real-time drowning detection system," *The Journal of Supercomputing,* vol. 81, no. 12, Aug. 2025, doi: 10.1007/s11227-025-07732-7.

[38] S. Teboulbi, S. Messaoud, M. A. Hajjaji, M. Atri, and A. Mtibaa, "Performance Evaluation of Advanced YOLOv10 and YOLOv11 Architectures for Object Detection in Autonomous Driving Scenarios," *IEEE Transactions on Computers,* pp. 1–21, Jan. 2026, doi: 10.1109/tc.2026.3666452.

[39] M. Mahala, M. Pattanaik, and G. Pandey, "Advancing Real-Time Object Detection for Autonomous Vehicles with YOLOv11," pp. 1–6, Mar. 2025, doi: 10.1109/iatmsi64286.2025.10984727.

[40] H. M. Z. Hossain *et al.*, "Evaluating YOLO Architectures: Implications for Real-Time Vehicle Detection in Urban Environments of Bangladesh," *arXiv (Cornell University)*, Sep. 2025, doi: 10.48550/arxiv.2509.05652.