Indonesian Journal of Data and Science

Volume 7 Issue 1
ISSN 2715-9936
https://doi.org/10.56705/ijodas.v7i1.371

Research Article

Vehicle Detection Using YOLOv8 on Low-Resolution Images

Nifal [1,*]; Farniwati Fattah [2]; Andi Widya Mufila Gaffar [3]
[1] Universitas Muslim Indonesia, Makassar, Indonesia, 13020220039@student.umi.ac.id
[2] Universitas Muslim Indonesia, Makassar, Indonesia, farniwati.fattah@umi.ac.id
[3] Universitas Muslim Indonesia, Makassar, Indonesia, widya.mufila@umi.ac.id

Correspondence should be addressed to Nifal; 13020220039@student.umi.ac.id

Received 03 Jan 2026; Accepted 02 Februari 2026; Published 30 March 2026

Abstract:
Vehicle detection in low-resolution images remains a significant challenge in computer vision, particularly for embedded devices such as ESP32-CAM with limited computational resources and simple image resolution. This study evaluates the performance of YOLOv8 on low-resolution QVGA (320 × 240 pixels) images for vehicle detection and classification. The dataset was independently collected in a controlled laboratory environment using miniature vehicles, covering four vehicle classes (motorcycle, car, bus, and truck) with a total of 4,000 images and a 70:20:10 data split. A pretrained YOLOv8 model was fine tuned for 100 epochs and tested on an ESP32-CAM prototype. The evaluation results demonstrate excellent performance, achieving precision of 0.999, recall of 1.000, mAP@0.5 of 0.995, and mAP@0.5-0.95 of 0.995 on the validation data, as well as real-time detection accuracy of 97% for motorcycles and cars, and 99% for buses and trucks. These findings indicate that YOLOv8 can deliver reliable vehicle detection performance on low-resolution images and is suitable for implementation in embedded device-based systems.

Keywords: Yolov8, Computer Vision, Vehicle Detection, Low-Resolution Images, ESP32-CAM.

1. Introduction

Real-time vehicle detection on low-resolution images is an important challenge in computer vision, particularly for embedded devices such as ESP32-CAM, which are widely used in IoT applications with limited computational resources [1], [2]. Conventional approaches often result in low detection accuracy under low-resolution conditions due to the loss of visual feature details, thereby reducing the efficiency of accurate vehicle object identification [3], [4]. Therefore, an object detection method capable of performing accurate and real-time vehicle estimation under limited resolution conditions is required.

Traditional computer vision approaches that combine image pre-processing techniques, such as segmentation, with classification algorithms have been widely used for visual object recognition and evaluated using accuracy, precision, and recall metrics [5], In recent years, deep learning–based computer vision has been extensively applied to vehicle detection due to its ability to automatically extract visual features from images [6]. The You Only Look Once (YOLO) algorithm is one of the most popular object detection methods because it enables real-time multi-class detection with high accuracy [7]. Deep learning–based computer vision approaches have also been implemented in various systems for real-time object recognition [8]. The latest version, YOLOv8, offers significant improvements in terms of architectural efficiency, training stability, and inference speed, making it well suited for vehicle detection scenarios with limited image resolution and computational resources [9], [10]. Several recent studies have shown that the integration of YOLOv8 can improve detection accuracy under low-resolution conditions [11], although further optimization is often required [11], [12], [13], [14].

Previous studies have commonly used public traffic datasets with high-resolution images and diverse environmental conditions [15]. However, performance evaluation of object detection models on low-resolution images (e.g., QVGA 320 × 240 pixels) and controlled environments remains relatively limited, particularly for vehicle count estimation scenarios [16]. Low resolution poses serious challenges in vehicle object detection, especially for small-sized classes such as motorcycles, due to the loss of feature details and reduced accuracy in small object detection [17], [18]. Recent studies indicate that YOLOv8 remains promising for applications under limited-resolution conditions when optimized through enhancement techniques or architectural modifications, although its baseline performance tends to degrade for small objects and low-resolution images [19], [20].

Based on this background, this study evaluates the performance of the YOLOv8 algorithm in detecting and classifying vehicles using QVGA-resolution images independently collected in a controlled laboratory environment using ESP32-CAM. The dataset includes four vehicle classes: motorcycle, car, bus, and truck. The model is trained and evaluated using standard object detection metrics (such as mAP, Precision, and Recall) and further tested on a prototype system as a limited implementation study. The primary focus of this research is the performance analysis of YOLOv8-based vehicle detection under low-resolution conditions, while system and hardware aspects are utilized solely as acquisition and testing media.

2. Method

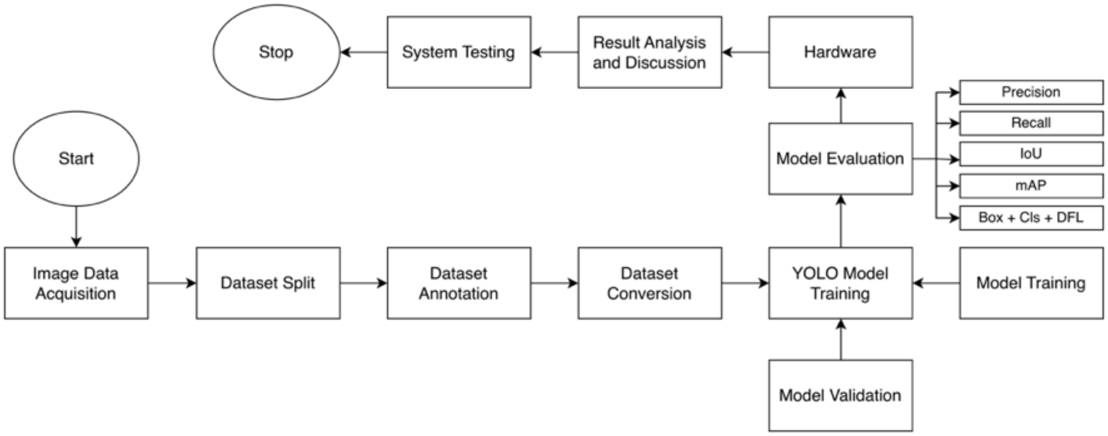


Figure 1. Research Design

This study employs an experimental research design to evaluate the performance of the YOLOv8 algorithm in detecting [21], and classifying vehicles on low-resolution images acquired using ESP32-CAM [22]. Image data were captured using an ESP32-CAM module with QVGA resolution (320 × 240 pixels) in a controlled laboratory environment and include four vehicle classes: motorcycle, car, bus, and truck [23]. The research stages consist of image data acquisition, dataset splitting, annotation and conversion to YOLO format, YOLOv8 model training, and model performance evaluation using predefined metrics [24], [25].

Image Data Acquisition

The dataset was collected in an indoor laboratory environment with a fixed camera angle and controlled lighting conditions using miniature vehicles to simulate two-way traffic flow [26], [27]. Image acquisition was independently performed using an ESP32-CAM module with QVGA resolution (320 × 240 pixels) installed at a distance of approximately 20 cm from the simulation area. Data were captured via an HTTP endpoint and sequentially stored using a Python-based program without additional pre-processing. The dataset consists of 4,000 images, evenly distributed across four simulated vehicle categories (car, motorcycle, bus, and truck; 1,000 images each) and does not originate from any public dataset. The controlled environment, low image resolution, and use of miniature vehicles make this dataset specifically focused on controlled evaluation of vehicle count estimation system performance [28].

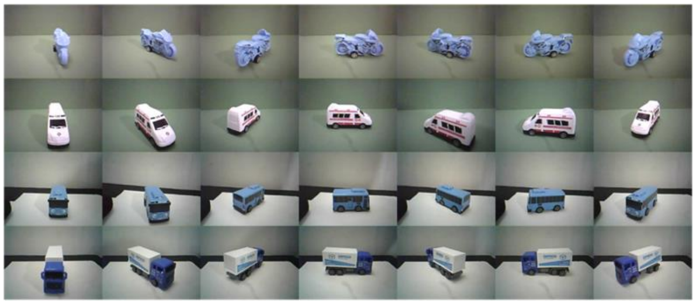


Figure 2. Miniature Vehicle Dataset

Dataset Split

The dataset was divided into three subsets: training data (70%), validation data (20%), and testing data (10%). The split was performed randomly to ensure balanced distribution and to prevent data leakage, where the training data were used for model training, the validation data for performance monitoring during training, and the testing data for final evaluation of model generalization [29].

Table 1. Dataset Split

Dataset Type	Percentage	Number of Images	Function
Training	70%	2.800	YOLO model training
Validation	20%	800	Performance monitoring
Testing	10%	400	Final evaluation
Total	100%	4.000	Entire dataset

Dataset Annotation and YOLO Format Conversion

After dataset splitting, each image was annotated with bounding boxes corresponding to vehicle objects (car, motorcycle, bus, and truck) as labels for object detection training. The annotations were converted into YOLO format (.txt) containing class indices and bounding box coordinates normalized to image dimensions, as required for YOLOv8 training. The dataset was organized following the standard YOLO directory structure for the modeling stage [30].

Table 2. Class Index Definition

Class Index	Vehicle Type
0	Motorcycle
1	Car
2	Bus
3	Truck

YOLO-Based Detection Model

This study employs YOLOv8 as the vehicle detection algorithm due to its lightweight and efficient architecture, which is suitable for real-time object detection on low-computational-resource devices. The model was trained using images resized to 640 × 640 pixels to detect four vehicle classes: motorcycle, car, bus, and truck. As illustrated in **Figure 3**, YOLOv8 consists of a Backbone (Conv Block, C2f, and SPPF) for feature extraction, a Neck (UpSample, Concat, and C2f) for multi-scale feature fusion, and a Detection Head based on a decoupled head that separates classification and bounding box regression processes to improve detection accuracy and inference speed [31].

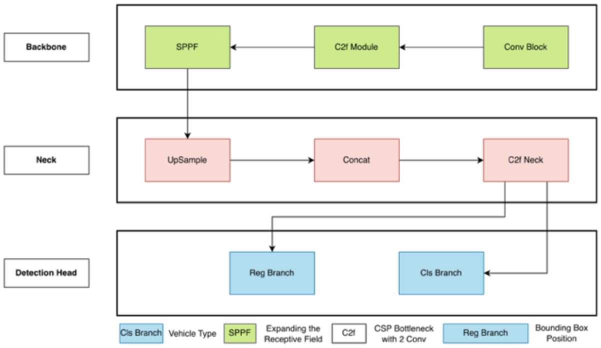


Figure 3. YOLOv8 Architecture

Model Training Configuration

Model training was conducted using YOLOv8 with pretrained initial weights and fine-tuned using the vehicle dataset in YOLO format. The training process was performed for 100 epochs with an input image size of 320 × 320 pixels, adjusted to match the QVGA resolution characteristics of ESP32-CAM, using a batch size of 16 and eight workers. All training was executed on an NVIDIA RTX 4060 GPU using default YOLOv8 parameters to ensure experimental consistency and reproducibility.

Model Validation Strategy

The validation strategy utilized 20% of the dataset that was not included in the training process to monitor the model's generalization capability and detect potential overfitting. Evaluation was periodically performed during training by the YOLOv8 framework by comparing prediction results with ground truth annotations from the validation dataset.

Model Evaluation Metrics

The performance of the YOLOv8 model was evaluated using standard object detection metrics, including Precision, Recall, Mean Average Precision (mAP@50), and mAP@50–95, along with monitoring of training and validation loss. All metrics were computed on the validation dataset to assess the model's generalization performance.

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

$$mAP@50 = \frac{1}{N}\sum_{i=1}^{N} AP_i$$

$$mAP@50 - 90 = \frac{1}{10}\sum_{k=1}^{N} mAP_{(0.50\ +\ 0.05k)}$$

$$\mathcal{L}_{total} = \mathcal{L}_{box} + \mathcal{L}_{cls} + \mathcal{L}_{DFL}$$

IoT Traffic Light Prototype

The trained model was implemented on an IoT-based adaptive traffic light prototype simulating a two-way intersection. The system employs two ESP32-CAM modules as visual sensors and one ESP32 as a central controller, along with a TM1637 display to present countdown timing. In addition to visual detection, the system is equipped with a sound sensor as a supporting feature for detecting priority vehicles; however, this aspect is not the main focus of the study. Vehicle count estimation is performed using the YOLOv8 model and evaluated in a controlled laboratory environment using a physical prototype and intersection simulation [32].

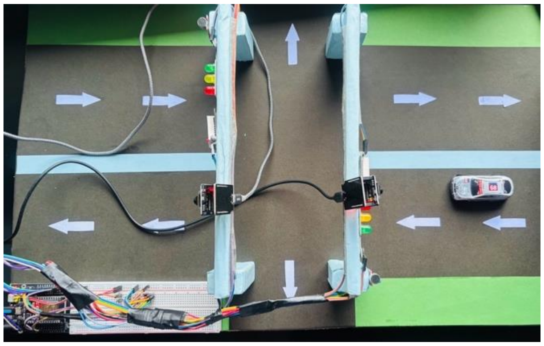


Figure 4. Adaptive Traffic Light System

3. Results and Discussion

Results

The evaluation results of the YOLOv8 model training over 100 epochs are presented in **Figure 5**, where the training and validation loss curves—consisting of box loss, classification loss, and distribution focal loss (DFL)—show a consistent decrease with similar trends across both data subsets. In addition, performance metrics such as precision, recall, and mean average precision (mAP) at various Intersection over Union (IoU) thresholds exhibit a stable improvement throughout the training process. The decreasing loss patterns and increasing metric values indicate that the model successfully achieved convergence and effectively improved vehicle detection performance under controlled testing conditions.

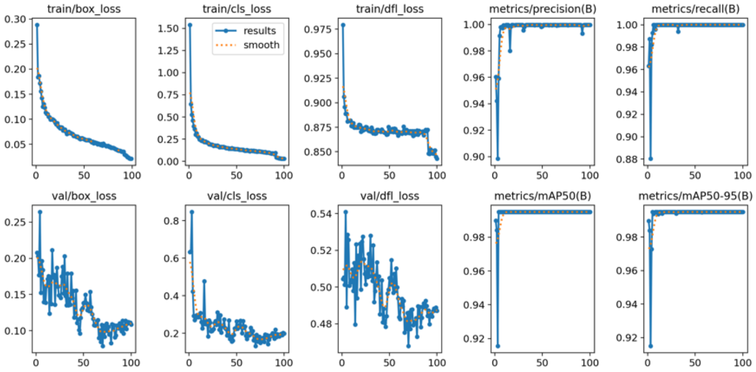
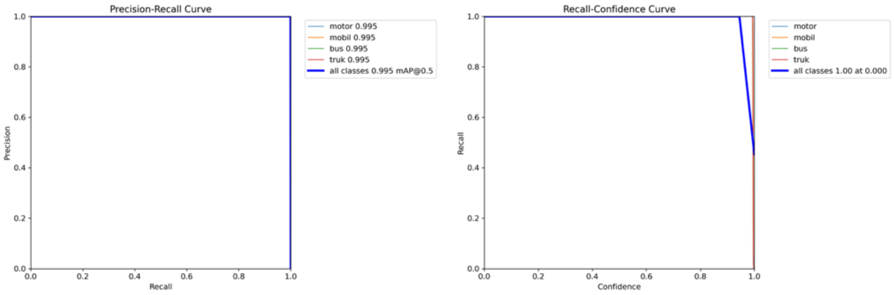


Figure 5. Training and Evaluation Results

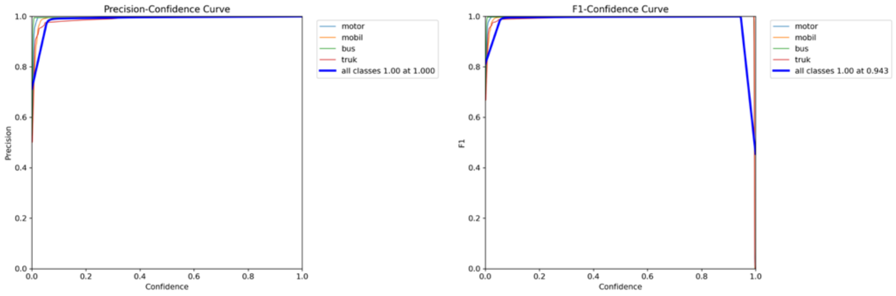
Figure 6 presents the Precision–Recall and Recall–Confidence curves for each vehicle class. The Precision–Recall curves illustrate the relationship between prediction accuracy and the model's ability to detect all vehicle objects, while the Recall–Confidence curves describe the model's confidence level in its detection results. These curves are used to evaluate the multi-class vehicle detection performance of YOLOv8.



(a) (b)

Figure 6. (a) Precision-Recall Curve and (b) Recall-Confidence Curve

Figure 7 shows the F1–Confidence and Precision–Confidence curves for each vehicle class. The F1–Confidence curves represent the balance between precision and recall, whereas the Precision–Confidence curves illustrate the effect of confidence thresholds on detection accuracy. These curves are utilized to analyze the performance of YOLOv8 and to determine the optimal confidence threshold.



(a) (b)

Figure 7. (a) F1-Confidence Curve and (b) Precision-Confidence Curve

Based on **Figure 8**, the model successfully classifies vehicles with a high level of accuracy, with only two images misclassified as background. This result indicates a relatively low classification error and demonstrates that the model is capable of effectively distinguishing between vehicle classes.

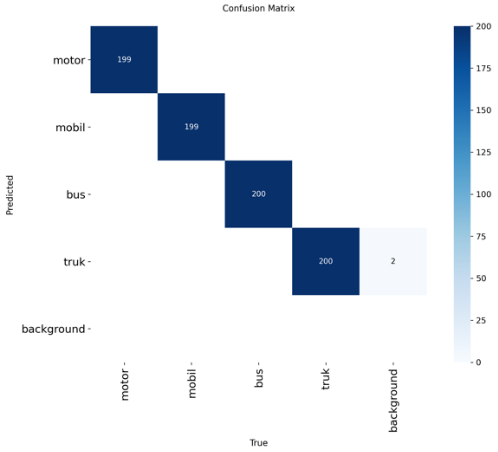


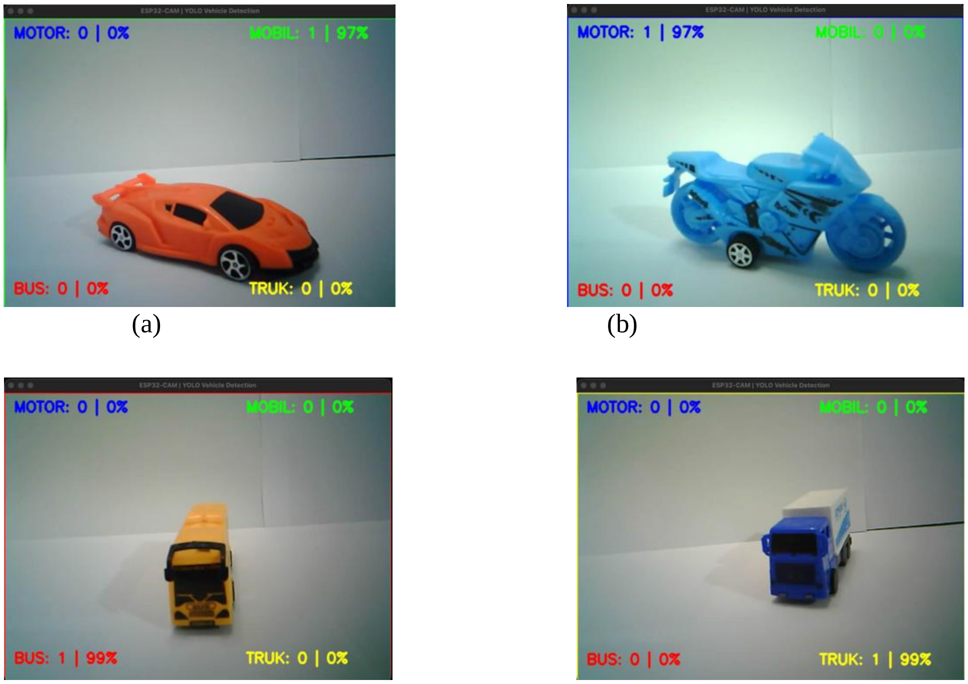
Figure 8. Confusion Matrix

As presented in **Figures 5**, **6**, and **7**, and summarized in **Table 3**, the vehicle detection system achieves excellent performance, with an overall precision (P) of 0.99975, recall (R) of 1.000, mAP@50 of 0.995, and mAP@50–95 of 0.995. These values confirm that the model exhibits very high accuracy in vehicle identification, where its ability to detect all target objects (100% recall) and minimize detection errors (precision close to 100%) remains stable up to the 100th epoch. This is further supported by the continuous reduction in validation box loss and classification loss, indicating that the model has reached an optimal convergence point without experiencing overfitting.

Table 3. Validation Results

Epoch	Box Loss (T)	Cls Loss (T)	Precision (B)	Recall (B)	mAP50 (B)	mAP50-95 (B)	Box Loss (V)	Cls Loss (V)
1	0.288	1.540	0.960	0.963	0.989	0.989	0.207	0.632
5	0.119	0.264	0.999	1.000	0.995	0.995	0.170	0.449
15	0.081	0.187	0.999	1.000	0.995	0.995	0.140	0.312
30	0.061	0.133	0.999	1.000	0.995	0.995	0.133	0.231
45	0.052	0.122	0.999	1.000	0.995	0.995	0.132	0.229
60	0.046	0.110	0.999	1.000	0.995	0.995	0.131	0.221
75	0.040	0.098	0.999	1.000	0.995	0.995	0.129	0.215
85	0.038	0.091	0.999	1.000	0.995	0.995	0.116	0.195
90	0.035	0.082	0.999	1.000	0.995	0.995	0.099	0.171
91	0.035	0.096	0.999	1.000	0.995	0.995	0.099	0.170
96	0.034	0.080	0.999	1.000	0.995	0.995	0.097	0.162
100	0.034	0.074	0.999	1.000	0.995	0.995	0.096	0.157

In the ESP32-CAM–based system testing, the YOLOv8 model demonstrates detection accuracies of 97% for motorcycles, 97% for cars, 99% for buses, and 99% for trucks, as shown in Figure 9. These results confirm that the model is capable of reliably detecting each vehicle class in real-device implementation under controlled testing conditions.



(a) (b)

(c) (d)

Figure 9. Real-time ESP32-CAM vehicle detection: (a) car, (b) motorcycle, (c) bus, (d) truck.

Discussion

The YOLOv8-based vehicle detection system implemented on ESP32-CAM demonstrates higher accuracy compared to conventional approaches such as loop detectors, ultrasonic sensors, or other simple methods, as it is capable of directly estimating vehicle counts from low-resolution images. By employing a lightweight YOLOv8 model on QVGA resolution, the system achieves competitive performance with a significantly lower computational burden than previous methods that rely on larger models or higher image resolutions. Buses and trucks are detected more accurately than motorcycles and cars, which is consistent with the influence of object size and low resolution on detection performance.

The evaluation was conducted using a laboratory dataset with miniature vehicles, QVGA resolution, and a fixed camera angle, which limits the system's ability to optimally handle occlusion, distance variations, extreme lighting changes, and real-world traffic dynamics. Further research is required to evaluate the system at real intersections with heterogeneous traffic, higher image resolutions, and multi-sensor integration to improve robustness and generalization.

4. Conclusion

This study successfully demonstrates that a YOLOv8 model integrated with an ESP32-CAM module can be effectively used to detect and classify vehicles in low-resolution images under controlled testing environments. Using QVGA-resolution images and four vehicle classes, the YOLOv8 model achieves very high detection performance, indicated by precision and recall values close to 1.0, as well as mAP@50 and mAP@50–95 of 0.995. Testing on real devices shows detection accuracies of 97% for motorcycles and cars, and 99% for buses and trucks, confirming that the lightweight YOLOv8 architecture can operate reliably on IoT devices with limited computational resources.

Nevertheless, the system performance was evaluated using a dataset collected in a controlled laboratory environment with miniature vehicles, low image resolution, and a fixed camera angle. Future work may focus on evaluation at real-world intersections with variations in lighting conditions, camera viewpoints, and more complex traffic densities, as well as exploring the integration of additional sensors to enhance the robustness and scalability of computer vision–based vehicle detection systems.

References:

[1] S. Ma, H. Lu, J. Liu, Y. Zhu, And P. Sang, "Layn: Lightweight Multi-Scale Attention Yolov8 Network For Small Object Detection," Ieee Access, Vol. 12, Pp. 29294–29307, 2024, Doi: 10.1109/Access.2024.3368848.

[2] X. Chai, M. Zhao, J. Li, And J. Li, "Image Small Target Detection In Complex Traffic Scenes Based On Yolov8 Multiscale Feature Fusion," Alexandria Engineering Journal, Vol. 126, Pp. 578–590, Jul. 2025, Doi: 10.1016/J.Aej.2025.04.105.

[3] M. Oussouaddi, O. Bouazizi, A. El Mourabit, Z. El A. Alaoui Ismaili, Y. Attaoui, And M. Chentouf, "Dsr-Yolo: A Lightweight And Efficient Yolov8 Model For Enhanced Pedestrian Detection," Cognitive Robotics, Vol. 5, Pp. 152–165, 2025, Doi: 10.1016/J.Cogr.2025.04.001.

[4] F. Feng, Y. Hu, W. Li, And F. Yang, "Improved Yolov8 Algorithms For Small Object Detection In Aerial Imagery," Journal Of King Saud University - Computer And Information Sciences, Vol. 36, No. 6, P. 102113, Jul. 2024, Doi: 10.1016/J.Jksuci.2024.102113.

[5] H. Azis, Nirmala, L. Syafie, Herman, F. Fattah, And T. Hasanuddin, "Unveiling Algorithm Classification Excellence: Exploring Calendula And Coreopsis Flower Datasets With Varied Segmentation Techniques," In 2024 18th International Conference On Ubiquitous Information Management And Communication (Imcom), Ieee, Jan. 2024, Pp. 1–7. Doi: 10.1109/Imcom60618.2024.10418246.

[6] L. Syafie, H. Azis, And F. T. Admojo, “Optimizing Javanese Numeral Recognition Using Yolov8 Technology: An Approach For Digital Preservation Of Cultural Heritage,” Indonesian Journal Of Data And Science, Vol. 6, No. 1, Pp. 113–122, Mar. 2025, Doi: 10.56705/Ijodas.V6i1.239.

[7] 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, P. 131, Jun. 2023, Doi: 10.3390/Jimaging9070131.

[8] F. Fathrurahman, Dolly Indra, Tasrif Hasanuddin, Herdianti Darwis, And Tanaka Kazuaki, “Smart Waste Bin Prototype For University Waste Management,” Indonesian Journal Of Data And Science, Vol. 6, No. 3, Pp. 378–392, Dec. 2025, Doi: 10.56705/Ijodas.V6i3.324.

[9] M. Bakirci, “Utilizing Yolov8 For Enhanced Traffic Monitoring In Intelligent Transportation Systems (Its) Applications,” Digit. Signal Process., Vol. 152, P. 104594, Sep. 2024, Doi: 10.1016/J.Dsp.2024.104594.

[10] Q.-A. N. Duc Et Al., “Optimizing Traffic Light Control Using Yolov8 For Real-Time Vehicle Detection And Traffic Density,” In 2024 9th International Conference On Integrated Circuits, Design, And Verification (Icdv), Ieee, Jun. 2024, Pp. 119–124. Doi: 10.1109/Icdv61346.2024.10616901.

[11] H. Azis, Nasrullah, M. Abdullah, S. Ismail, Purnawansyah, And L. Syafie, “A Comparative Study Of Yolo Models For Enhanced Vehicle Detection In Complex Aerial Scenarios,” In 2025 19th International Conference On Ubiquitous Information Management And Communication (Imcom), Ieee, Jan. 2025, Pp. 1–8. Doi: 10.1109/Imcom64595.2025.10857527.

[12] S. Saklani, M. Manchanda, R. Sharma, And D. Singh, “Real-Time Traffic Management System Using Yolov8 And Cnn: A Deep Learning Approach With Iot Integration,” In 2025 First International Conference On Advances In Computer Science, Electrical, Electronics, And Communication Technologies (Ce2ct), Ieee, Feb. 2025, Pp. 645–651. Doi: 10.1109/Ce2ct64011.2025.10939584.

[13] A. Bhivgade And C. Puri, “Vision-Based Adaptive Traffic Signal System Using Yolov8 For Smart City Infrastructure,” In 2025 6th International Conference On Electronics And Sustainable Communication Systems (Icesc), Ieee, Sep. 2025, Pp. 1604–1610. Doi: 10.1109/Icesc65114.2025.11212240.

[14] M. Raza Et Al., “An Edge-Deployed Real-Time Adaptive Traffic Light Control System Using Yolo-Based Vehicle Detection And Pce-Aware Density Estimation,” Ieee Access, Vol. 13, Pp. 153586–153613, 2025, Doi: 10.1109/Access.2025.3602844.

[15] T. Azfar, K. Huang, A. Tracy, S. Misiewicz, C. Liu, And R. Ke, “Traffic Co-Simulation Framework Empowered By Infrastructure Camera Sensing And Reinforcement Learning,” Sep. 2025, Doi: 10.48550/Arxiv.2412.03925.

[16] D. Stiller Et Al., “Open Webcam Data For Traffic Monitoring: Yolov8 Detection Of Road Users Before And During Covid-19,” Transp. Res. Interdiscip. Perspect., Vol. 36, P. 101774, Mar. 2026, Doi: 10.1016/J.Trip.2025.101774.

[17] Y. Cheng, Z. Ning, Z. Deng, K. Li, And J. Yin, “Psf3-Yolov8:Low Resolution Vehicle Detection Algorithm Based On Improved Yolov8,” In 2024 12th International Conference On Information Systems And Computing Technology (Isctech), Ieee, Nov. 2024, Pp. 1–6. Doi: 10.1109/Isctech63666.2024.10845721.

[18] A. Sundaresan 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, Aug. 2024, Doi: 10.3390/Vehicles6030065.

[19] B. Khalili And A. W. Smyth, “Sod-Yolov8—Enhancing Yolov8 For Small Object Detection In Aerial Imagery And Traffic Scenes,” Sensors, Vol. 24, No. 19, P. 6209, Sep. 2024, Doi: 10.3390/S24196209.

[20] Y. Liu, Z. Huang, Q. Song, And K. Bai, “Pv-Yolo: A Lightweight Pedestrian And Vehicle Detection Model Based On Improved Yolov8,” Digit. Signal Process., Vol. 156, P. 104857, Jan. 2025, Doi: 10.1016/J.Dsp.2024.104857.

[21] J. Qiu, W. Zhang, S. Xu, And H. Zhou, “Dp-Yolo: A Lightweight Traffic Sign Detection Model For Small Object Detection,” Digit. Signal Process., Vol. 165, P. 105311, Oct. 2025, Doi: 10.1016/J.Dsp.2025.105311.

[22] A. R. Manga’, H. Herawati, And P. Purnawansyah, “Utilization Of Deep Learning Yolo V9 For Identification And Classification Of Toraja Buffalo Breeds,” Ilkom Jurnal Ilmiah, Vol. 17, No. 1, Pp. 12–19, Apr. 2025, Doi: 10.33096/Ilkom.V17i1.2349.12-19.

[23] Q. Shi, F. Zhong, B. Li, Z. Xu, J. Hong, And R. Li, “Fast Vehicle Detection Algorithm Based On Lightweight Yolo7-Tiny,” In Fifth International Conference On Computer Vision And Data Mining (Iccvdm 2024), X. Zhang And M. Yin, Eds., Spie, Oct. 2024, P. 113. Doi: 10.1117/12.3048399.

[24] J. Li, J. Zhang, Y. Shao, And F. Liu, “Sre-Yolov8: An Improved Uav Object Detection Model Utilizing Swin Transformer And Re-Fpn,” Sensors, Vol. 24, No. 12, P. 3918, Jun. 2024, Doi: 10.3390/S24123918.

[25] W. Zhang, Y. Li, J. Hu, And N. Wang, “A Study On Front Vehicle Collision Warning Method Based On Lightweight Yolov8 And Deepsort,” Egyptian Informatics Journal, Vol. 33, P. 100861, Mar. 2026, Doi: 10.1016/J.Eij.2025.100861.

[26] R. Ayachi, Y. Said, M. Afif, A. Alshammari, M. Hleili, And A. Ben 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.

[27] H. Bae Et Al., “Miniature Testbed For Validating Multi-Agent Cooperative Autonomous Driving,” Nov. 2025, Doi: 10.48550/Arxiv.2511.11022.

[28] D. Li, P. Auerbach, And O. Okhrin, “Autonomous Driving Small-Scale Cars: A Survey Of Recent Development,” Ieee Transactions On Intelligent Transportation Systems, Vol. 26, No. 10, Pp. 14591–14614, Oct. 2025, Doi: 10.1109/Tits.2025.3572394.

[29] J. Pohlmann, M. Matthe, T. Kronauer, P. Auerbach, And G. Fettweis, “Ros2-Based Small-Scale Development Platform For Ccam Research Demonstrators,” In 2022 Ieee 95th Vehicular Technology Conference: (Vtc2022-Spring), Ieee, Jun. 2022, Pp. 1–6. Doi: 10.1109/Vtc2022-Spring54318.2022.9860981.

[30] M. Sivakumar, S. Parthasarathy, And T. Padmapriya, “Trade-Off Between Training And Testing Ratio In Machine Learning For Medical Image Processing,” Peerj Comput. Sci., Vol. 10, P. E2245, Sep. 2024, Doi: 10.7717/Peerj-Cs.2245.

[31] M. A. Salem And A. H. Rabia, “Real-Time On-The-Go Annotation Framework Using Yolo For Automated Dataset Generation,” In 2025 Interdisciplinary Conference On Electrics And Computer (Intcec), Ieee, Sep. 2025, Pp. 1–7. Doi: 10.1109/Intcec65580.2025.11256048.

[32] Y. Liu And S. Shen, “Vehicle Detection And Tracking Based On Improved Yolov8,” Ieee Access, Vol. 13, Pp. 24793–24803, 2025, Doi: 10.1109/Access.2025.3538556.

[33] M. Raza Et Al., “An Edge-Deployed Real-Time Adaptive Traffic Light Control System Using Yolo-Based Vehicle Detection And Pce-Aware Density Estimation,” Ieee Access, Vol. 13, Pp. 153586–153613, 2025, Doi: 10.1109/Access.2025.3602844.