BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 74(4), 2026, Article number: e158309
DOI: 10.24425/bpasts.2026.158309

Modified YOLO11 for improving car detection performance on the highway

SUTIKNO*, Aris SUGIHARTO, and Retno KUSUMANINGRUM

Diponegoro University, Faculty of Science and Mathematics, Department of Informatics, Semarang, Indonesia

Abstract. Vehicle detection in the road area serves as the basis for various other studies, including vehicle type identification, vehicle counting, and traffic violation detection. Two models that have been successfully implemented were YOLOv2 and YOLOv3, but both still have limitations in detecting small objects and require significant computational resources. YOLO11 displays enhanced accuracy and computational efficiency. Large variant models such as YOLO11m, YOLO11l, and YOLO11x achieve higher accuracy but often come with a substantial decrease in inference speed. Therefore, this study proposes modifying the YOLO11 architecture by removing two convolutional blocks and three C3k2 Blocks to increase detection speed without sacrificing accuracy. The proposed method is applied to two datasets: low-traffic and high-traffic highways. The test results showed that the suggested approach could reduce the inference time by 19% on low-traffic highways and 12% on high-traffic highways, respectively. In addition, the proposed method can increase the Average Precision (AP) by 0.004 on low-traffic highways and 0.003 on high-traffic highways, making it better suited for real-time car detection on highways.

Keywords: modified YOLO11; car detection; increased speed.

1. INTRODUCTION

The increasing number of cars on the road each year causes problems such as congestion and reduced traffic safety. Vehicle detection on the road plays a crucial role in improving the safety and efficiency of the transportation system. This detection technology also provided foundations for various other studies, such as vehicle type identification [1–3], vehicle counting [4–7], speed limit violation detection [8–10], detection of drivers who do not use seat belts [11–14], and detection of reckless drivers [15, 16]. The development of computer vision and image processing technology now offers promising solutions to face these challenges. One widely used algorithm for real-time object detection is YOLO (You Only Look Once), whose primary advantage lies in its fast and efficient computing.

Several previous studies have used the YOLO algorithm for vehicle detection. Sang *et al.* implemented the YOLOv2-vehicle network using the BIT-vehicle dataset. They reported that the algorithm improved precision and average IoU values compared to the YOLOv2 model and the Comp-model [17]. Xu *et al.* compared several YOLO variants – namely, YOLOv3, improved YOLOv3, and modified YOLOv3 – on the COCO dataset. The experimental results showed that the modified YOLOv3 achieved the highest average precision among the three variants [18]. Wang *et al.* also compared several YOLO models, namely YOLOv2, Tiny YOLOv2, Tiny YOLOv3, and SPPNet-YOLOv3. Based on the test results, SPPNet-YOLOv3 provided a higher mAP value than the other models [19]. Research by Jahan *et al.* compared the performance of YOLOv3, Improved YOLOv3, Faster R-CNN, and Modified YOLOv3. The results obtained showed that Modified YOLOv3 achieved the highest average precision among all tested methods [20]. Zuraimi *et al.* studied several YOLO variants, namely YOLOv4, YOLOv4-tiny, YOLOv3, and YOLOv3-tiny. The test results indicated that YOLOv4 achieved the best accuracy among the variants [21]. Furthermore, Song and Gu proposed using YOLOv5 for real-time vehicle detection. This model was compared with traditional detection methods and demonstrated lower false-detection rates [22]. Meanwhile, Rafi *et al.* developed a YOLOv5-based vehicle detection and tracking system by comparing three models, namely YOLOv5s, YOLOv5m, and YOLOv5l. According to the evaluation results, YOLOv5l achieved the best performance, with the highest mAP value among the three [23].

YOLOv2 and YOLOv3 are renowned for their ability to detect multiple objects in a single frame. However, both still have limitations in detecting small objects and require high computational resources, making them less than optimal for real-time applications [15, 16]. As an improvement over the previous version, YOLO11 offers higher accuracy and better computational efficiency. However, there is a trade-off between accuracy and inference time. Large variant models, such as YOLO11m, YOLO11x, and YOLO11l, have achieved higher accuracy and recall; however, their inference time also increased significantly. YOLO11 has been modified to improve accuracy or decrease inference time. Hua-Qin *et al.* modified YOLO11 by improving feature representation via guided and aggregated attention and by optimizing convolutional efficiency, resulting in a more accurate and lightweight model [24]. Meanwhile, Zhong *et al.* improved YOLO11 by integrating the Efficient Channel Attention (ECA) mechanism into the backbone to strengthen the repre-

*e-mail: sutikno@lecturer.undip.ac.id

Manuscript submitted 2025-10-31, revised 2026-02-10, initially accepted for publication 2026-03-23, published in July 2026.

sentation of directional crack features, and combining it with a multi-threshold segmentation strategy to produce accurate and complete road crack extraction [25]. This study modified the YOLO11 architecture by removing certain blocks to increase inference speed without significantly sacrificing accuracy.

2. PROPOSED METHOD

This study proposed a modified YOLO11 for road-car detection. This modification aims to reduce inference time (increase speed) without reducing accuracy.

2.1. YOLO11 architecture

The architecture of YOLO11 is illustrated in Fig. 1. YOLO11 employs a multi-scale prediction head to detect objects at multiple scales, including low, medium, and high. These detection boxes use feature maps generated by the backbone and neck. The C3K2 block is the primary component of the YOLO11 backbone, efficiently extracting features. This structure is developed from the CSP (Cross Stage Partial) bottleneck in the previous version by using a small convolution kernel (3×3) to speed up computation without reducing the ability to detect key features. CSP is a network architecture that divides and recombines features to reduce computation without degrading critical feature extraction capabilities. C3K2 uses a series of C3K blocks and convolutional layers at the beginning and end to optimize information flow and enrich feature representations. By dividing and recombining feature maps after several convolutional layers, this block maintains a balance between accuracy and inference time while reducing the number of parameters compared to previous structures, such as C2f in YOLOv8. The C3K2 block is illustrated in Fig. 2. Figure 3 shows a C3K block, one of the components within the C3K2 block.

2.2. Modified YOLO11 architecture (Proposed method)

The main contribution of this study is to propose a new YOLO architecture that modifies YOLO11 to improve the detection efficiency of cars on highways with different traffic characteristics, i.e., low traffic and high traffic. The YOLO11 modification involves reducing feature maps to support medium-scale prediction. This reduction is achieved by removing the blocks directly connected to the feature maps, because in car detection on the highway, especially in images with fixed camera

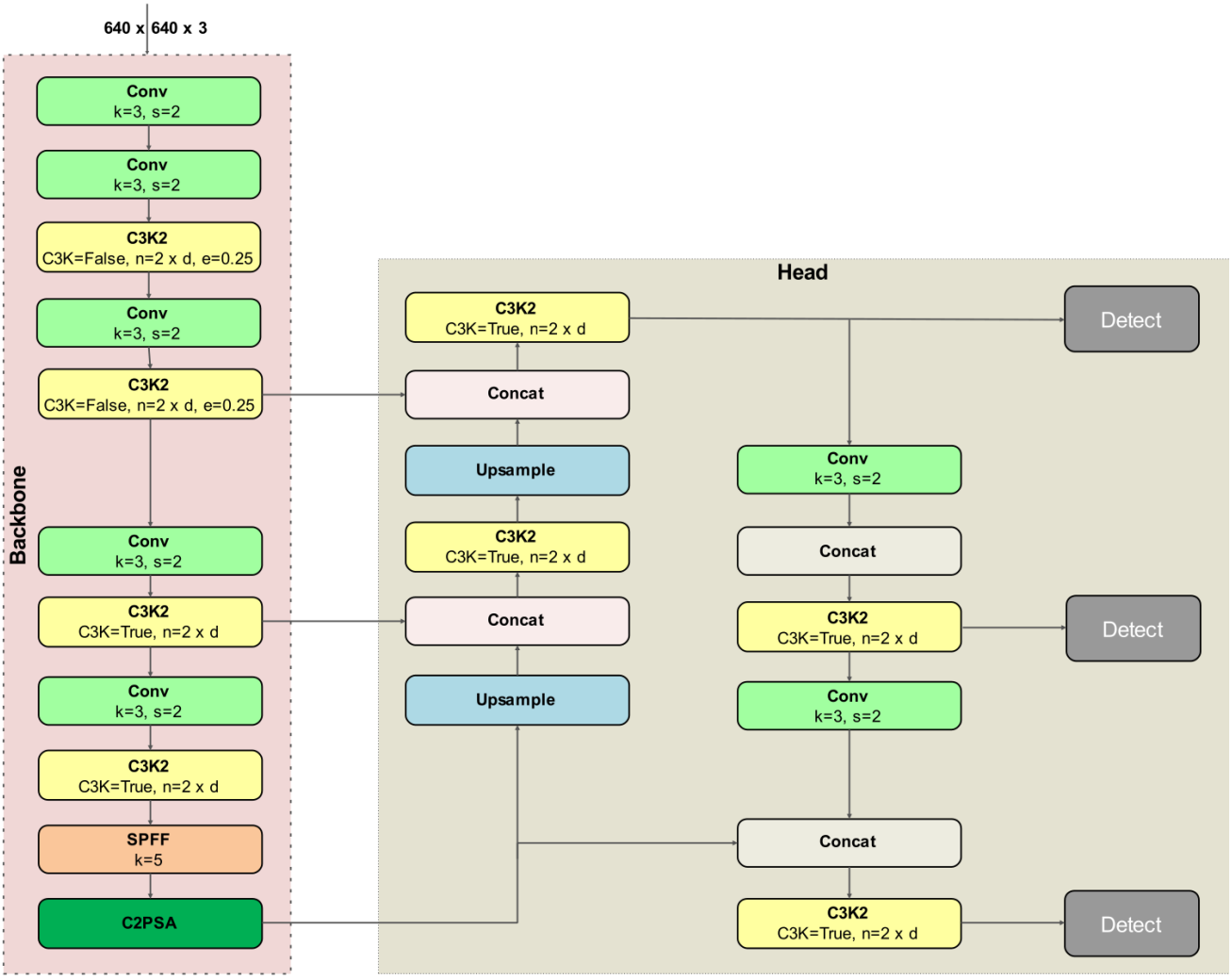


Fig. 1. YOLO11 architecture

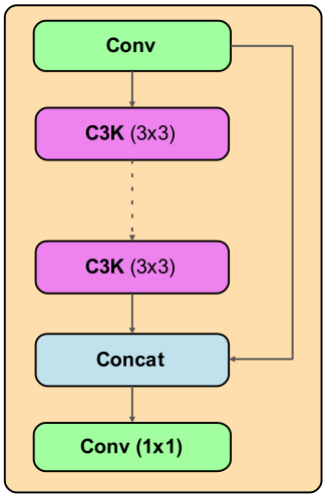


Fig. 2. C3K2 block

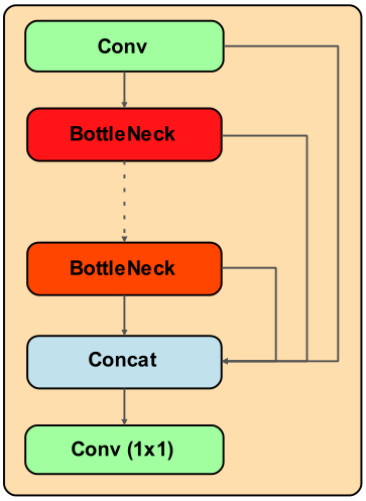


Fig. 3. C3K block

angles, the medium-scale contribution to accuracy is not always significant.

Specifically, the removed block consists of two convolutional blocks (Conv blocks) and three C3K2 blocks. This reduction in the number of blocks aims to lower inference time and reduce memory usage, making the model lighter and better suited for real-time applications. By eliminating these blocks, the modified YOLO11 architecture is expected to maintain accurate car detection performance in both low and high traffic conditions, without sacrificing significant accuracy, while reducing inference time. The modified YOLO11 architecture is shown in Fig. 4.

2.3. Evaluation metric

The proposed model is evaluated using three metrics: precision, recall, and Average Precision (AP). Additionally, we assess inference time (T_{inf}) and training times (T_{tra}). In object detection, precision measures how correctly the model detects the objects, and it is the ratio of correct detections to all predicted detec-

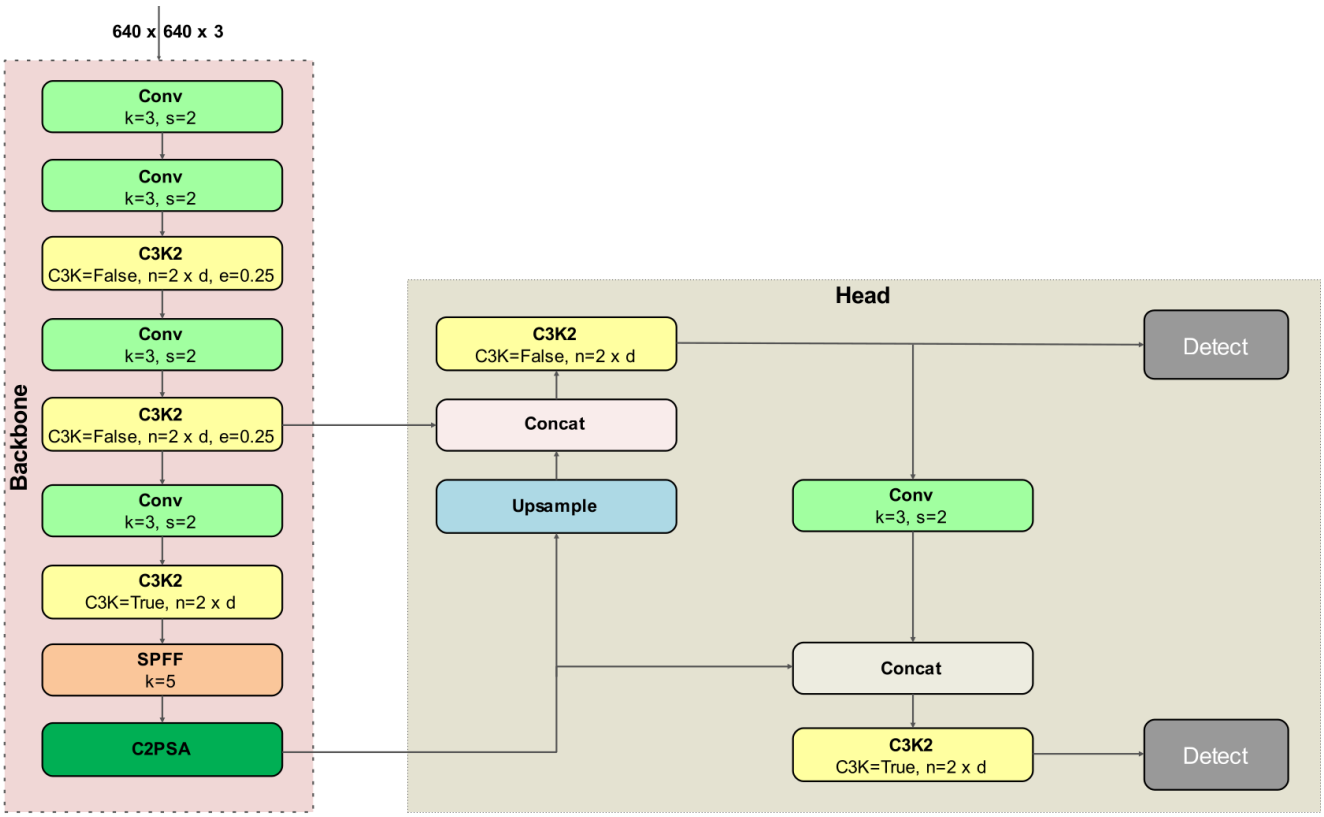


Fig. 4. Modified YOLO11 architecture (Proposed method)

tions. Recall measures how well the model finds all objects that are actually present in the image. Meanwhile, AP represents the overall performance of the model for a single class of objects by calculating the area under the precision–recall curve.

Precision, recall, and AP are calculated using equations (1), (2), and (3), respectively [26]. TP is the correct detection of the ground-truth bounding box, FP is the incorrect detection of the existing or nonexistent object, and FN is the ground-truth not detected. $P_i(R_{n+1})$ is calculated using equation (4), where $P(\overline{R})$ is the precision measured at recall $\overline{R}$.

$$P = \frac{TP}{TP+FP}, \tag{1}$$

$$R = \frac{TP}{TP+FN}, \tag{2}$$

$$AP = \sum_{n} (R_{n+1} - R_n)\, P_i\,(R_{n+1}), \tag{3}$$

$$P_i\,(R_{n+1}) = \max_{\overline{R}:\ \overline{R} \geq R_{n+1}} P\left(\overline{R}\right). \tag{4}$$

3. EXPERIMENTS AND RESULTS

3.1. Dataset

This study used two datasets. Both are the results of recordings taken on the highway in Semarang, Indonesia. An example of a video frame from this dataset is shown in Fig. 5. Figure 5a shows the first dataset, SMG-LowTraffic (SLT), collected on low-traffic roads. Figure 5b shows the second dataset, SMG-HighTraffic (SHT), collected on roads and toll roads with heavy traffic. The SLT dataset comprises 2680 frames at 960×540 px. Meanwhile, the SHT dataset comprises 2 116 frames at 3840×2160 px. These datasets are divided into two parts: the first 80% is used for training, and the remaining 20% for testing. Furthermore, each frame is annotated in the Roboflow platform with bounding boxes for each car. Annotations are performed manually to ensure the accuracy of the object location and size. Roboflow is also used to standardize YOLO11 annotation formats and share datasets.

3.2. Experiments

Testing was conducted using an NVIDIA A100 GPU (40 GB).

All tests used an input image size of 640×640 px, a batch size of 16, were trained for 100 epochs with a learning rate of 0.01, and utilized pretrained weights with an automatic optimizer. Data augmentation techniques such as mosaic, RandAugment, and HSV color transformations were applied to enhance car detection performance. Testing began with several proposed models: modified YOLO11n, YOLO11s, YOLO11m, YOLO11l, and YOLO11x. These models were then compared with the unmodified YOLO11 model. Furthermore, the proposed models were compared with previous research.

The results of the proposed method testing are shown in Table 1. The test results on the SLT dataset showed that the high-

(a) (b)

Fig. 5. Example of a video frame: (a) SLT dataset, (b) SHT dataset

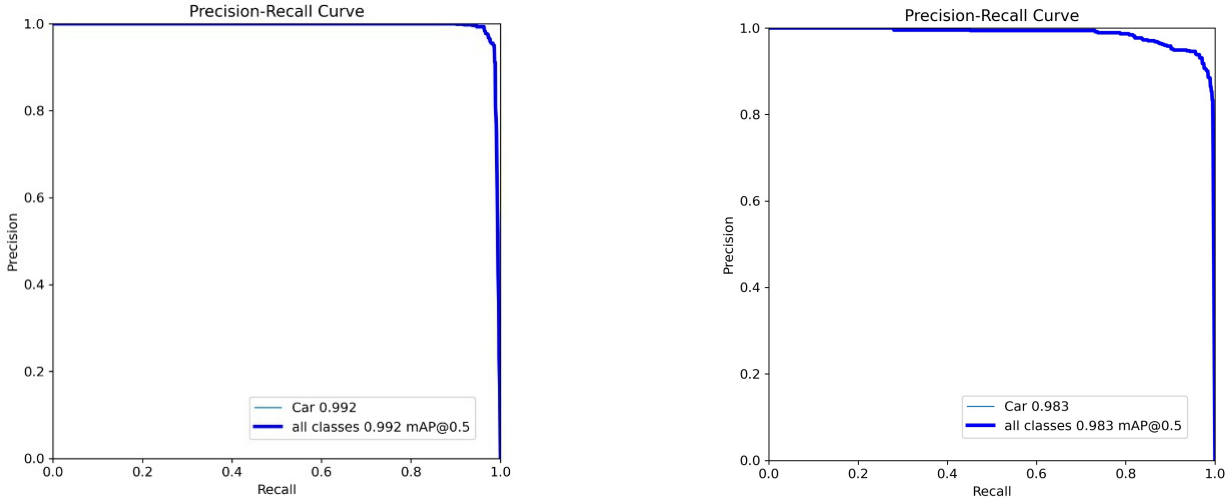
Table 1
Test results of the proposed method on five models

Model	SLT dataset			SHT dataset		
	Precision	Recall	AP	Precision	Recall	AP
Modified YOLO11n	0.985	0.964	0.991	0.941	0.969	**0.983**
Modified YOLO11s	0.975	**0.969**	0.989	0.937	0.967	0.981
Modified YOLO11m	**0.993**	0.964	**0.992**	0.933	**0.978**	0.982
Modified YOLO11l	0.989	0.963	0.990	**0.945**	0.956	**0.983**
Modified YOLO11x	0.989	0.963	0.990	**0.945**	0.956	**0.983**

est recall with the modified YOLO11s model was 0.969, while the highest precision and AP were achieved with the modified YOLO11m at 0.993 and 0.992, respectively. In the SHT dataset, the highest recall was achieved with the modified YOLO11m model, while the highest precision and AP were achieved with the modified YOLO11l and YOLO11x models. The modified YOLO11n also achieves the highest AP, although the precision value is lower than that of these two models. For this reason, the best model of the proposed method that will be compared with other models is modified YOLO11m for the SLT dataset and YOLO11l for the SHT dataset.

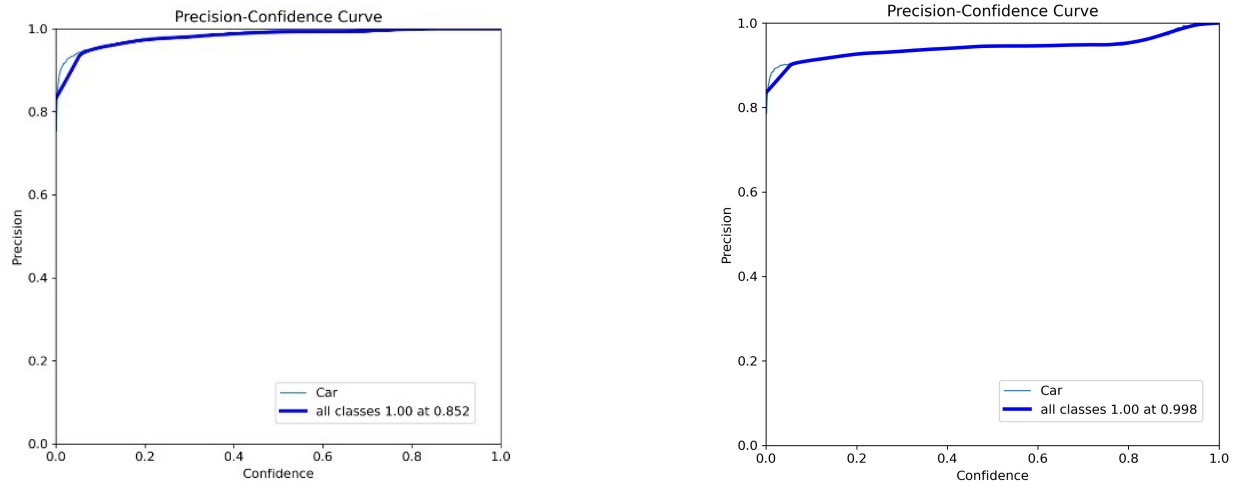
In the SLT dataset, the standard YOLO11m has 20 053 779 parameters and 68.2 GFLOPs. In comparison, the proposed method (modified YOLO11m) uses 10 125 266 parameters and 60.3 GFLOPs only, resulting in approximately 49.5% parameter reduction and a decrease in computational complexity. In the SHT dataset, the standard YOLO11l has 25 311 251 parameters and 87.3 GFLOPs, while the proposed method (modified YOLO11l) reduces the number of parameters to 12 820 690 and GFLOPs to 76.4, resulting in a decrease of approximately 49.3%.

The results of training and testing the proposed method can be displayed as curves. Figure 6 is a view of the precision-recall curve of the proposed method. Figure 6a for the SLT dataset and Fig. 6b for the SHT dataset. Figure 7 is a view of the Precision-confidence curve, where Fig. 7a is for the SLT dataset, and Fig. 7b is for the SHT dataset. Figure 8 shows the Recall-confidence curve for the method, with Fig. 8a for the SLT dataset and Fig. 8b for the SHT dataset. The training and testing performance of the proposed method is shown in Fig. 9. Figure 9a shows the results for the SLT dataset, and Fig. 9b shows the results for the SHT dataset.



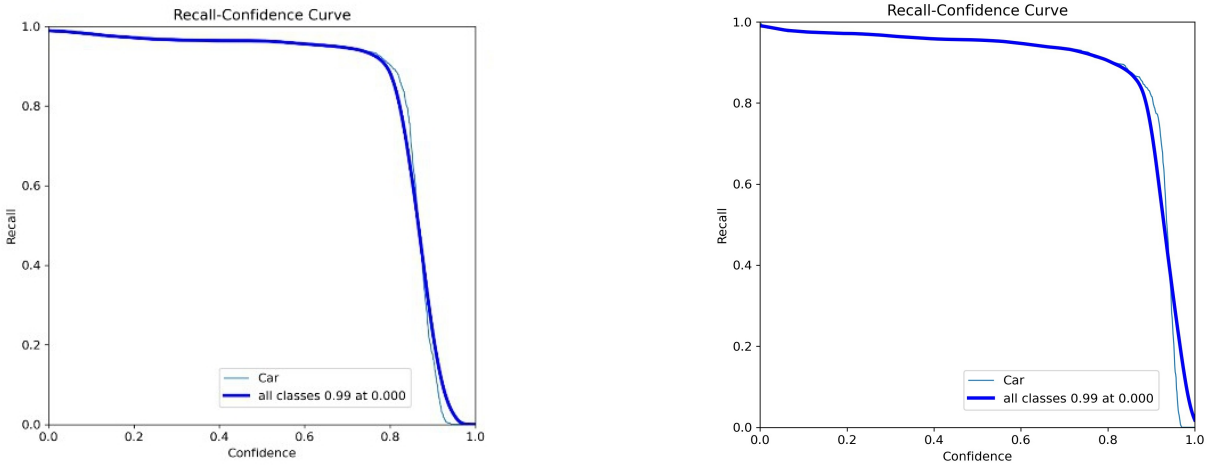
(a) (b)

Fig. 6. Precision-recall curve: (a) SLT dataset, (b) SHT dataset



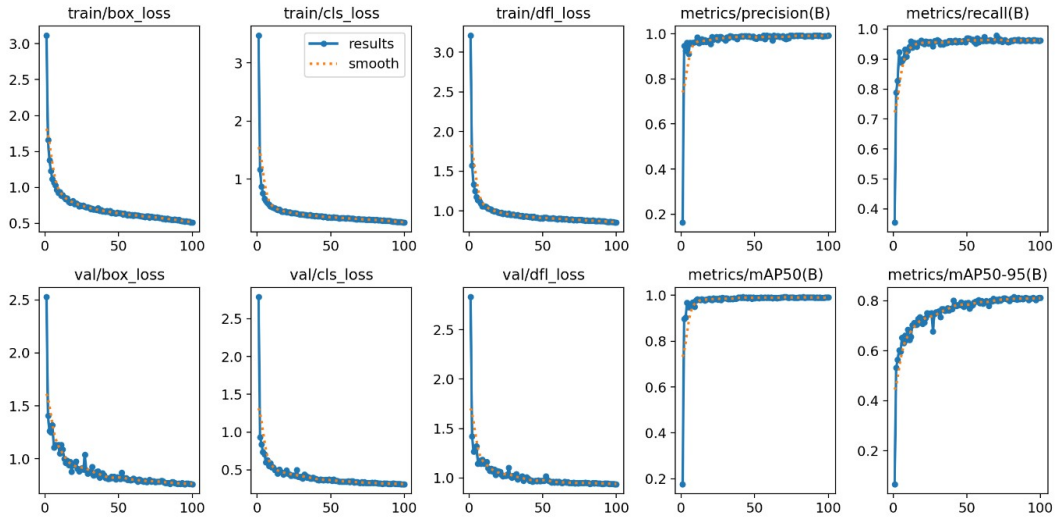
(a) (b)

Fig. 7. Precision-confidence curve: (a) SLT dataset, (b) SHT dataset

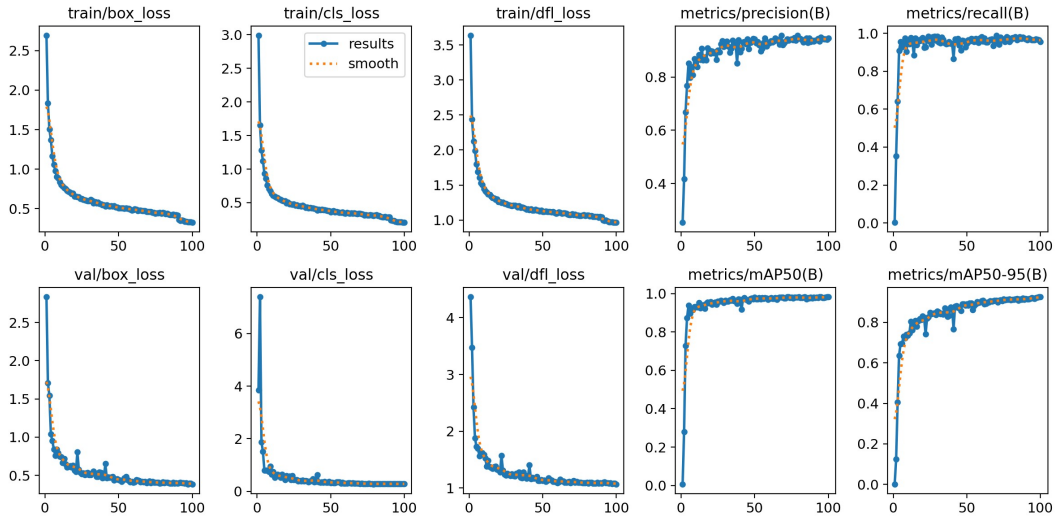


(a) (b)

Fig. 8. Recall-confidence curve: (a) SLT dataset, (b) SHT dataset



(a)



(b)

Fig. 9. Performance of training and testing: (a) SLT dataset, (b) SHT dataset

Figure 10 shows an example of car detection. Figure 10a is for the SLT dataset, and Fig. 10b is for the SHT dataset. The proposed method can detect cars and those partially blocked by other objects, such as trees, in low-traffic areas. However, in heavy traffic, the proposed method still cannot detect partially cut cars or car objects that are too large (i.e., exceeding the frame height).

The best proposed method was compared with five YOLO11 models (before modification). The results of this test are shown in Table 2. The proposed method yields a higher AP than others in both datasets. In addition, the proposed method is superior to other methods in terms of precision on the SLT dataset, although it is slightly inferior to YOLO11n on the SHT dataset. Compared to YOLO11n on inference time, the proposed method is about twice as high on the SLT dataset and three times as high on the SHT dataset. However, this comparison is not entirely fair as it would be more appropriate to compare the proposed method with the baseline architecture. The results of this comparison are shown in Table 3. The best model of the proposed method is modified YOLO11m for the SLT dataset and modified YOLO11l for the SHT dataset. For this reason, we compared YOLO11m on the SLT dataset and YOLO11l on the SHT dataset. The proposed method consistently improves AP across both datasets. The magnitude of the AP increase was 0.004 in the SLT dataset and 0.003 in the SHT dataset. In addition, the proposed method can reduce inference time and training time. The decrease in inference time was 0.3 ms (19%) for the SLT dataset and 0.2 ms (12%) for the SHT dataset. Meanwhile, the training reduction

(a) (b)

Fig. 10. Example of car detection: (a) SLT dataset, (b) SHT dataset

Table 2
Comparison of the proposed method with 5 YOLO11 models

Model	SLT dataset					SHT dataset				
	Precision	Recall	AP	T_{inf} (ms)	T_{tra} (hours)	Precision	Recall	AP	T_{inf} (ms)	T_{tra} (hours)
YOLO11n	0.991	0.958	0.986	**0.8**	**0.449**	**0.948**	0.963	0.977	**0.5**	**0.535**
YOLO11s	0.974	**0.969**	0.990	0.9	0.471	0.941	0.985	0.981	0.8	0.558
YOLO11m	0.987	0.961	0.988	1.6	0.609	0.941	0.974	0.979	1.2	0.633
YOLO11l	0.987	0.958	0.987	1.9	0.800	0.926	**0.986**	0.980	1.7	0.736
YOLO11x	0.987	0.963	0.989	2.3	1.034	0.919	0.976	0.969	2.4	0.913
Proposed	**0.993**	0.964	**0.992**	1.3	0.590	0.945	0.956	**0.983**	1.5	0.673

Table 3
Comparison of the proposed method and baseline YOLO11 model

Parameter	SLT dataset			SHT dataset		
	Yolo11m	Proposed model	Difference	Yolo11l	Proposed model	Difference
AP	0.988	**0.992**	0.004	0.980	**0.983**	0.003
T_{inf} (ms)	1.6	**1.3**	0.3	1.7	**1.5**	0.2
T_{tra} (hours)	0.609	**0.590**	0.019	0.736	**0.673**	0.063

Table 4
Comparison of the proposed method and previous research

Model	SLT dataset					SHT dataset				
	Precision	Recall	AP	T_{inf} (ms)	T_{tra} (hours)	Precision	Recall	AP	T_{inf} (ms)	T_{tra} (hours)
YOLOv8x [27]	0.990	0.961	0.990	2.1	1.059	**0.945**	**0.969**	0.977	2.3	1.046
YOLOv8m	0.989	0.959	0.990	1.8	0.768	0.936	0.965	0.980	**1.4**	0.751
Proposed	**0.993**	**0.964**	**0.992**	**1.3**	**0.590**	**0.945**	0.956	**0.983**	1.5	**0.673**

time was 0.019 hours for the SLT dataset and 0.063 hours for the SHT dataset. Therefore, it can be concluded that the proposed method, which reduces two Conv blocks and three C3k2 blocks, can increase AP while decreasing inference and training time.

The proposed method was then compared with previous research. We tested it on the same dataset, as in Table 4. In reference [27], YOLOv8x was used with a batch size of 8 and 100 epochs. Additionally, we compared it with YOLOv8m. The results of the comparison show that the proposed method is superior to YOLOv8x and YOLOv8m in terms of precision, AP, and training time in both datasets. In addition, the proposed method is superior to others in terms of recall and inference time on the SLT dataset, although it falls slightly behind on the SHT dataset. For this reason, it can be concluded that the proposed method achieves higher precision and AP compared to previous research.

4. CONCLUSIONS

This study proposed a modified version of YOLO11 architecture, referred to as the modified YOLO11. This model is used for detecting cars on highways. Modifications are made by reducing the feature maps for prediction at a medium scale, specifically using two Convolutional Blocks and three C3K2 Blocks. The test results show that the proposed method consistently reduces inference time and increases AP across both datasets. The decrease in inference time was 19% in datasets with low traffic and 12% in datasets with heavy traffic. The magnitude of the AP increase is 0.004 in low-traffic conditions and 0.003 in high-traffic conditions. In addition, the proposed method can reduce training time. The suggested approach can also detect small cars and partially blocked cars, such as those partially blocked by other objects, in low-traffic areas. However, in heavy traffic, the proposed method still cannot detect partially cut cars. Therefore, it is better suited to real-time conditions than the YOLO11 model for detecting cars on highways with light and heavy traffic.

REFERENCES

[1] S. Naseer, S.M.A. Shah, S. Aziz, M.U. Khan, and K. Iqtidar, "Vehicle make and model recognition using deep transfer learning and support vector machines," in *Proc. 2020 23rd IEEE International Multi-Topic Conference, INMIC*, 2020, pp. 4–9, doi: 10.1109/INMIC50486.2020.9318063.

[2] M.W. Nafi'I, E.M. Yuniarno, and A. Affandi, "Vehicle brands and types detection using Mask R-CNN," in *Proc. 2019 International Seminar on Intelligent Technology and Its Application*, 2019, pp. 422–427, doi: 10.1109/ISITIA.2019.8937278.

[3] I.V. Pustokhina *et al.*, "Automatic vehicle license plate recognition using optimal K-Means with Convolutional Neural Network for intelligent transportation systems," *IEEE Access*, vol. 8, pp. 92907–92917, 2020, doi: 10.1109/ACCESS.2020.2993008.

[4] M.A. Abdelwahab, "Accurate vehicle counting approach based on deep neural networks," in *Proc. 2019 International Conference on Innovative Trends in Computer Engineering*, 2019, pp. 1–5, doi: 10.1109/ITCE.2019.8646549.

[5] Q. Chen, N. Huang, J. Zhou, and Z. Tan, "An SSD algorithm based on vehicle counting method," in *Chinese Control Conference, CCC*, 2018, pp. 7673–7677, doi: 10.23919/ChiCC.2018.8483037.

[6] C.M. Tsai, F.Y. Shih, and J.W. Hsieh, "Real-time vehicle counting by deep-learning networks," in *Proc. International Conference on Machine Learning and Cybernetics*, 2022, pp. 175–181, doi: 10.1109/ICMLC56445.2022.9941299.

[7] J.M. Anil, L. Mathews, R. Renji, R.M. Jose, and S. Thomas, "Vehicle counting based on convolution neural network," in *Proc. 7th International Conference on Intelligent Computing and Control Systems*, 2023, pp. 695–699, doi: 10.1109/ICICCS56967.2023.10142302.

[8] J. Timofejevs, A. Potapovs, and M. Gorobetz, "Algorithms for computer vision based vehicle speed estimation sensor," in *63rd Annual International Scientific Conference on Power and Electrical Engineering of Riga Technical University*, 2022, pp. 1–6, doi: 10.1109/RTUCON56726.2022.9978802.

[9] W. Jianping, L. Zhaobin, L. Jinxiang, G. Caidong, S. Maoxin, and T. Fangyong, "An algorithm for automatic vehicle speed detection using video camera," in *Proc. 2009 4th International Conference on Computer Science and Education*, 2009, pp. 193–196, doi: 10.1109/ICCSE.2009.5228496.

[10] A. Lad, P. Kanaujia, Soumya, and Y. Solanki, "Computer vision enabled adaptive speed limit control for vehicle safety," in *Proc. 2021 1st IEEE International Conference on Artificial Intelligence and Machine Vision*, 2021, pp. 4–8, doi: 10.1109/AIMV53313.2021.9670944.

[11] A. Upadhyay, B. Sutrave, and A. Singh, "Real time seatbelt detection using YOLO deep learning model," in *2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, 2023, pp. 1–6, doi: 10.1109/SCEECS57921.2023.10063114.

[12] J. Albesa and M. Gasulla, "Seat occupancy and belt detection in removable seats via inductive coupling," in *IEEE 74th Ve-*

hicular Technology Conference, 2011, pp. 1–5, doi: 10.1109/VETECF.2011.6093145.

[13] B. Zhou, L. Chen, J. Tian, and Z. Peng, "Learning-based seat belt detection in image using salient gradient," in *12th IEEE Conference on Industrial Electronics and Applications, ICIEA 2017*, 2018, vol. 2018-Febru, pp. 547–550, doi: 10.1109/ICIEA.2017.8282904.

[14] Z. Wang and Y. Ma, "Detection and recognition of stationary vehicles and seat belts in intelligent Internet of things traffic management system," *Neural Comput. Appl.*, vol. 34, no. 5, pp. 3513–3522, 2022, doi: 10.1007/s00521-021-05870-6.

[15] T. Heo, W. Nam, J. Paek, and J. Ko, "Autonomous reckless driving detection using deep learning on embedded GPUs," in *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2020, pp. 464–472, doi: 10.1109/MASS50613.2020.00063.

[16] A. Chauhan, S. Kumar, L. Mahto, and D.N. Jagadish, "Detection of reckless driving using deep learning," in *Proc. 19th IEEE International Conference on Machine Learning and Applications, ICMLA 2020*, 2020, pp. 853–858, doi: 10.1109/ICMLA51294.2020.00139.

[17] J. Sang *et al.*, "An improved YOLOv2 for vehicle detection," *Sensors*, vol. 18, no. 12, p. 4272, 2018, doi: 10.3390/s18124272.

[18] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, 2006, doi: 10.1109/TPAMI.2006.79.

[19] X. Wang, S. Wang, J. Cao, and Y. Wang, "Data-driven based Tiny-YOLOv3 method for front vehicle detection inducing SPP-Net," *IEEE Access*, vol. 8, pp. 110227–110236, 2020, doi: 10.1109/ACCESS.2020.3001279.

[20] N. Jahan, S. Islam, and M.F.A. Foysal, "Real-time vehicle classification using CNN," in *11th International Conference on Computing, Communication and Networking Technologies*, 2020, doi: 10.1109/ICCCNT49239.2020.9225623.

[21] M.A. Bin Zuraimi and F.H. Kamaru Zaman, "Vehicle detection and tracking using YOLO and DeepSORT," in *ISCAIE 2021 - IEEE 11th Symposium on Computer Applications and Industrial Electronics*, 2021, pp. 23–29, doi: 10.1109/ISCAIE51753.2021.9431784.

[22] X. Song and W. Gu, "Multi-objective real-time vehicle detection method based on YOLOv5," in *2021 International Symposium on Artificial Intelligence and its Application on Media (ISAIAM)*, 2021, pp. 142–145, doi: 10.1109/ISAIAM53259.2021.00037.

[23] M.M. Rafi *et al.*, "Performance analysis of deep learning YOLO models for South Asian regional vehicle recognition," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 9, pp. 864–873, 2022, doi: 10.14569/IJACSA.2022.01309100.

[24] H.-Q. Wu, H. Yan, H. Zhang, S.-W. Xu, F.-Y. Gao, and Z.-W. Chen, "Optimized industrial surface defect detection based on improved YOLOv11," *Struct. Durab. Health Monit.*, vol. 20, no. 1, 2026, doi: 10.32604/sdhm.2025.070589.

[25] Z. Wang, L. Fan, and J. Zou, "A crack detection technology for urban asphalt roads based on an improved YOLOv11 network model with increased ECA attention mechanism," in *Lecture Notes in Electrical Engineering*, vol. 1493 LNEE, pp. 390–397, 2026, doi: 10.1007/978-981-95-3320-6_34.

[26] R. Padilla, S.L. Netto, and E.A. B. Da Silva, "A survey on performance metrics for object-detection algorithms," in *International Conference on Systems, Signals, and Image Processing*, 2020, pp. 237–242, doi: 10.1109/IWSSIP48289.2020.9145130.

[27] Sutikno, A. Sugiharto, R. Kusumaningrum, and H.A. Wibawa, "Improved car detection performance on highways based on YOLOv8," *Bull.Electr. Eng. Inf.*, vol. 13, no. 5, pp. 3526–3533, 2024, doi: 10.11591/eei.v13i5.8031.