Improving YOLOv12 with Selective Kernel Networks for Autonomous Driving Object Detection

Pandu Pangestu
Department of Informatics
Siliwangi University
Tasikmalaya, Indonesia
pangestup60@gmail.com

Aradea
Department of Informatics
Siliwangi University
Tasikmalaya, Indonesia
aradea@unsil.ac.id

Vega Purwayoga
Department of Informatics
Siliwangi University
Tasikmalaya, Indonesia
vega.purwayoga@unsil.ac.id

Rina Mardiati
Department of Electrical Engineering
UIN Sunan Gunung Djati Bandung
Bandung, Indonesia
r_mardiati@uinsgd.ac.id

***Abstract*—Object detection in autonomous vehicles requires a model that can preserve accuracy under object-size variation caused by camera distance and dynamic road environments. YOLOv12 provides a real-time architecture strengthened by attention mechanisms, but its multiscale feature representation can still be improved through adaptive receptive-field selection. This paper proposes YOLOv12n-SKNet, a modified YOLOv12n architecture that integrates the Selective Kernel Network into the feature aggregation path to dynamically select kernel contributions through split, fuse, and select operations for improved feature representation. The evaluation is conducted on the KITTI dataset with eight object classes using precision, recall, mAP50, mAP50-95, parameter count, GFLOPs, and inference time. Experimental results show that YOLOv12n-SKNet improves precision from 90.1% to 94.8%, recall from 76.0% to 83.1%, mAP50 from 87.0% to 90.6%, and mAP50-95 from 64.3% to 69.8%. The additional computational cost remains manageable, with parameters increasing from 2.5M to 2.9M and GFLOPs from 5.8 to 7.7, while inference time remains 1.6 ms per image. These results indicate that SKNet strengthens YOLOv12n detection performance with acceptable computational overhead.**

Keywords—autonomous vehicles, deep learning, KITTI dataset, object detection, Selective Kernel Network, YOLOv12

I. INTRODUCTION

The development of autonomous vehicles represents a significant innovation in smart transportation systems due to its potential to enhance safety, driving efficiency, and reduce human error and traffic congestion [1]. The implementation of Automated Driving Systems is also projected to yield annual social benefits of nearly 800 billion US dollars by 2050 through accident reduction, energy efficiency, and increased productivity [2]. These autonomous vehicles require a visual perception system capable of accurately identifying various traffic objects such as vehicles, pedestrians, cyclists, trams, and road signs within milliseconds [3]. The accuracy and latency of the object detection system directly determine the reliability and safety of autonomous vehicle navigation [4].

Camera-based object detection has become a primary approach because cameras can provide rich visual information at a relatively low implementation cost [5]. Deep learning approaches, particularly those based on Convolutional Neural Networks (CNNs), have dominated modern AV perception systems due to their superior hierarchical feature extraction capabilities compared to traditional methods [6]. Among CNN-based object detector architectures, the YOLO family has emerged as a leading choice for real-time applications, owing to its single-stage formulation that jointly predicts bounding box coordinates and class probabilities in one forward pass, thereby providing an optimal balance between accuracy and inference speed [7].

YOLOv12 introduces a real-time object detection architecture that maintains convolutional operations while enhancing feature representation through Area Attention (A2) and the Residual Efficient Layer Aggregation Network (R-ELAN) [8]. The highway environment continues to present challenges in the form of complex object scale variations. Traffic objects in highway scenarios have a very wide scale range due to differences in distance and camera perspective [9]. Trucks located close to the camera can occupy a large pixel area in the image, whereas pedestrians or cyclists located far away often appear as small objects with limited spatial detail. This condition poses a challenge for standard convolution operations because fixed kernel sizes tend to produce a static receptive field that is insensitive to variations in object scale [10].

The Selective Kernel Network (SKNet) [11], offers a relevant approach to address the limitations of static receptive fields through a dynamic kernel selection mechanism. In standard convolution, the kernel size is fixed at each layer, so the model tends to use the same feature extraction pattern for objects of different sizes. SKNet processes the feature map through two parallel convolution branches with different kernel sizes, then adaptively selects the contribution of each branch based on the visual content of the input, allowing the network to adjust the focus of its representation between local details and broader spatial context.

Based on the proposed innovations, this study proposes YOLOv12n-SKNet, a modified YOLOv12n architecture that integrates a Selective Kernel Network (SKNet) to enhance the model's ability to handle object-scale variations in autonomous vehicle scenarios. This approach is expected to improve detection accuracy for various object types on

979-8-3195-1150-8/26/$31.00 ©2026 IEEE

highways while maintaining computational complexity that remains suitable for real-time perception systems, namely:

1. Adopting YOLOv12n as the baseline one-stage detector for object detection tasks in autonomous vehicles, considering that the nano variant has low computational complexity and is suitable for initial evaluation in real-time scenarios.
2. Introducing YOLOv12n-SKNet by integrating the Selective Kernel Network into the YOLOv12n pipeline to strengthen adaptive receptive-field selection capabilities, enabling the model to handle variations in the scale of traffic objects caused by differences in camera distance and perspective.
3. Evaluating the impact of SKNet integration on YOLOv12n performance on the KITTI dataset using accuracy and efficiency metrics, including precision, recall, mAP50, mAP50-95, number of parameters, GFLOPs, and inference time.

II. Related Works

Object detection for autonomous vehicle perception has advanced significantly alongside the growing demand for robust, real-time road scene understanding. Deep learning-based approaches have come to dominate this field, and are broadly classified into two-stage and one-stage detectors [12]. Each category presents distinct trade-offs between detection accuracy, inference latency, and suitability for deployment in time-critical systems [13].

Two-stage detectors operate through two main processes, generating candidate object regions and classifying the proposed regions. Approaches such as Faster R-CNN and Cascade R-CNN are known for their strength in object localization because the region proposal process allows the model to examine object candidates in a more targeted manner [14]. The main drawback of the two-stage approach lies in its higher computational cost, making it less ideal for autonomous vehicle perception, which requires rapid responses on the millisecond timescale [15]. Comparative studies in highway scenarios also show that two-stage models are often used as accuracy baselines, but the significant computational overhead leads to one-stage architectures being preferred when inference efficiency is a priority [16].

One-stage detectors formulate object detection as a direct prediction of the bounding box and object class in a single network pass. This approach is more suitable for autonomous vehicles because the perception system's latency must be kept low to enable real-time navigation decisions [17]. YOLO has become representative of one-stage detectors because it processes images comprehensively and generates object predictions simultaneously [18]. Subsequent developments in YOLO have focused on improving accuracy for small objects, multiscale feature fusion, and computational efficiency in complex traffic environments [19].

Several studies have developed YOLO-based models for autonomous vehicle applications. MCS-YOLO extends YOLOv5 with Coordinate Attention and a multi-scale detection structure to enhance sensitivity toward small and densely packed objects in highway environments [20]. CMS-YOLO modifies YOLOv5s through Cross Stage Partial DWNeck, a Multi-scale Feature Fusion Pyramid Network, and a Special Decoupled Head to improve multi-object detection performance in road scenes [21]. DAN-YOLO uses YOLOv7-Tiny with a Dilated Aggregation Network to expand the receptive field and strengthen vehicle detection under challenging driving conditions [22]. DCW-YOLO integrates a dynamic head, Coordinate Attention, and Wise-IoU v3 into YOLOv8 to improve feature representation and the localization of road objects [23].

Other developments focus on small object detection, feature resolution enhancement, and model efficiency. SNCE-YOLO utilizes SPD-Conv, Normalization-based Attention, a C2f structure, and EIOU loss to improve target detection in complex road scenes [24]. MSSD employs multi-scale self-distillation so that feature map information at various scales can help the model detect small targets without an external teacher model [25]. LEAD-YOLO extends YOLOv11n with a Convolutional Gated Transformer, Dilated Feature Fusion, a Hierarchical Feature Fusion Module, and a Small Feature Detection Head to strengthen small object detection for autonomous driving [26]. SCCA-YOLO combines spatial attention and channel self-attention in YOLOv8 to enhance the road perception system, particularly in rural road scenarios [27].

These studies demonstrate that improvements in object detection performance for autonomous vehicles are generally achieved through three main approaches, the addition of attention mechanisms, the optimization of multiscale feature fusion, and the refinement of the detection head structure. Attention mechanisms help the model prioritize features more relevant to the detection target. However, most approaches still employ attention patterns that do not explicitly select receptive field sizes based on object scale [18]. On the other hand, multiscale feature fusion can enrich the representation of both small and large objects, but overly complex modifications risk increasing computational load and reducing inference speed [28]. This gap highlights the need for a mechanism capable of enhancing scale adaptation without altering the overall real-time detection paradigm.

The Selective Kernel Network offers a different approach through adaptive kernel selection based on the response of the input feature map [11]. This mechanism allows the network to adjust the dominance of local features or broader spatial context based on the characteristics of objects in the image.

This review positions YOLOv12n-SKNet as a modified YOLOv12n architecture for autonomous vehicle object detection that uses SKNet to support adaptive receptive-field selection. Unlike YOLO-based attention mechanisms that mainly reweight spatial or channel features, SKNet performs dynamic kernel selection in the feature aggregation path. This mechanism allows the model to adapt its receptive-field response to different object scales, which is important for road scenes containing both small distant objects and large nearby objects. Therefore, the proposed contribution is positioned as a scale-adaptive receptive-field refinement strategy for YOLOv12n rather than only an additional attention module, while maintaining the real-time detection paradigm required by autonomous vehicle perception systems.

III. Proposed Method

The adaptability of the proposed architecture is determined by contextual visual knowledge captured from real-world road scenes, particularly object-scale variation caused by camera distance, perspective, and environmental uncertainty [29]. This concept is consistent with contextual adaptation in self-adaptive systems, where system behavior is

adjusted according to contextual requirements [30] [31]. It also extends adaptive-kernel-based pattern recognition in vision systems, where kernel-level adaptation improves feature representation under varying input characteristics [32].

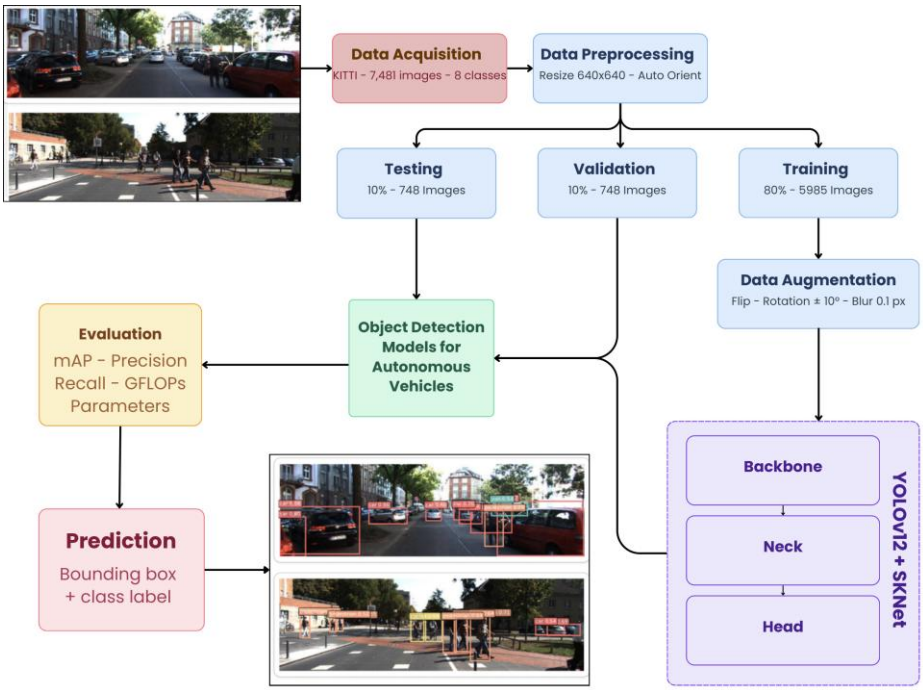

Fig. 1. *General flow architecture for developing YOLOv12n-SKNet*

YOLOv12n serves as the baseline architecture in this study. The proposed model, referred to as YOLOv12n-SKNet, integrates the Selective Kernel Network (SKNet) into the feature aggregation path. This integration preserves the one-stage detection paradigm of YOLOv12n while enabling adaptive kernel selection to improve feature sensitivity toward object-scale variations in road images.

Fig. 1 illustrates the research process, which begins with the acquisition of KITTI data, consisting of 7,481 images and eight object classes. The data is then processed through a resizing step to 640 x 640 pixels and auto-orientation before being divided into training, validation, and test sets. The training data was augmented with horizontal flipping, 10° rotation, and 0.1-pixel blurring to expand visual variation [33].

Architecturally, YOLOv12n is retained as the baseline single-stage detector. Input images are processed by the backbone to generate feature maps at multiple resolution levels, these features are then fused through the neck before being passed to the detection head.

Fig. 2 shows that SKNet is inserted after the A2C2f block on the top-down path in the neck section. This placement was chosen because the neck serves to combine semantic information from low-resolution features with spatial details from high-resolution features. The fused features can therefore be enhanced first through adaptive receptive field selection before being passed on to the bottom-up aggregation process and the detection head.

SKNet operates through three main stages, namely split, fuse, and select [11]. In the split stage, the input feature map is processed through two parallel convolutional branches that produce feature representations $\tilde{U}$ and $\hat{U}$. These two branches have different receptive fields, enabling them to capture object characteristics at different visual scales. The complete formulation of the SKNet mechanism is shown in equations (1), (2), (3), (4), and (5) [11].

$$U = \tilde{U} + \hat{U} \tag{1}$$

Represents the element-wise summation of the feature responses generated by the two convolutional branches. This operation combines local and wider spatial information into a unified feature representation U, which is then used as the basis for the next stage.

$$s_c = F_{gp}(U_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} U_c(i,j) \tag{2}$$

Applies global average pooling to each channel of U. This process compresses the spatial information of each feature map into a channel descriptor s_c, allowing the network to capture global contextual information from the fused feature representation.

$$z = F_{fc}(s) = \delta\big(\mathcal{B}(Ws)\big) \tag{3}$$

In the select stage, projects the channel descriptor $s$into a more compact feature representation z. This transformation is performed using a fully connected layer, followed by batch normalization and a nonlinear activation function. The compact representation is used to reduce computational complexity while preserving important channel-wise information.

$$a_c = \frac{e^{A_c z}}{e^{A_c z} + e^{B_c z}}, \quad b_c = \frac{e^{B_c z}}{e^{A_c z} + e^{B_c z}} \tag{4}$$

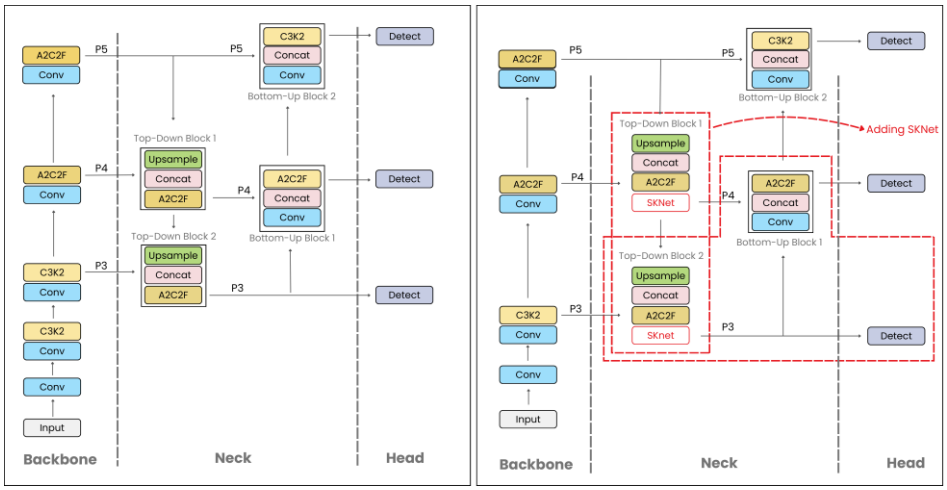

Fig. 2. *Architecture of the YOLOv12n baseline and YOLOv12n-SKNet*

Final output of the SKNet block is obtained by multiplying the attention weights by each branch's feature map, then summing them. This mechanism allows the model to assign greater weights to branches with small receptive fields for local details or to branches with wider receptive fields for larger spatial contexts.

$$V_c = a_c \cdot \widehat{U_c} + b_c \cdot \widehat{U_c} \quad (5)$$

Integrating SKNet into YOLOv12n is expected to improve the model's sensitivity to small objects while preserving balanced representations of larger objects. Model performance is evaluated using standard object detection metrics commonly adopted in YOLO-based studies, including precision, recall, F1-score, mAP, parameter count, GFLOPs, and inference time. Precision measures the correctness of positive predictions, recall reflects the model's ability to detect all target objects, and mAP evaluates detection quality across different IoU thresholds. Parameter count, GFLOPs, and inference time are used to assess computational feasibility for real-time deployment.

IV. Experiment

The experiment was conducted using the KITTI dataset, which is a standard benchmark in the development of autonomous vehicle perception systems [34]. The dataset contains 7,481 real-world traffic images with eight classes: car, van, truck, pedestrian, person_sitting, cyclist, tram, and misc [35]. Since the official KITTI benchmark withholds ground-truth labels for its test split, a custom random split was applied in this study using a fixed seed of 42, dividing the dataset into 5,985 training images, 748 validation images, and 748 testing images, corresponding to an 80:10:10 ratio. This splitting strategy was applied to prevent data leakage, so the validation and test images remained unseen during model training.

Models are trained using the Ultralytics YOLOv12 framework on the Kaggle platform with two NVIDIA Tesla T4 GPUs. Model performance is assessed using detection accuracy metrics and computational efficiency metrics, including precision, recall, mAP50, mAP50-95, parameter count, GFLOPs, and inference time measured using Ultralytics' built-in profiling on a single image per inference. The training configuration used in this experiment is summarized in Table I.

TABLE I. Experimental Configuration

Parameter	Value
Epoch	250
Patience	25
Batch Size	32
Optimizer	Stochastic Gradient Descent
Initial Learning Rate	0.01
Image Size (Input)	640 x 640 pixels
Seed, Deterministic	42, True

As shown in Table I, all models were trained for 250 epochs with a batch size of 32 and an early stopping patience of 25. Optimization was performed using Stochastic Gradient Descent (SGD) with an initial learning rate of 0.01. To ensure reproducibility, deterministic mode and a random seed of 42 were applied. To determine the optimal placement for the SKNet module within the architecture, an experimental scheme was conducted to evaluate the trade-off between computational cost and expected detection accuracy. Since the experiment was conducted using a single random seed, the reported results are interpreted as single run performance under a controlled training configuration.

TABLE II. Comparison of SKNet Insertion Points in YOLOv12n

Position	Params	GFLOPs	mAP50	mAP50-95
Backbone	3.125.456	10.6	87.4%	63.2%
Neck	**2.897.872**	**7.7**	**90.6%**	**69.8%**
Head	4.103.184	8.7	90.6%	70.0%

Inserting SKNet at the backbone results in the highest computational cost with the lowest accuracy, indicating that early-stage feature maps are not sufficiently semantic for effective kernel selection. Head insertion achieves a slightly higher mAP50-95 of 70.0% but introduces significantly more parameters 4.10M and GFLOPs 8.7. As summarized in Table II, neck insertion offers the best trade-off achieving the lowest parameter count 2.90M and 7.7 GFLOPs while matching the head's mAP50 of 90.6% with only a 0.2% drop in mAP50-95.

Performance comparison among YOLO-based models is presented in Table III. The comparison uses precision, recall, mAP50, and mAP50-95 as detection accuracy metrics, while parameter count, GFLOPs, and inference time are used to assess computational efficiency.

TABLE III. PERFORMANCE COMPARISON OF YOLO-BASED MODELS ON THE KITTI DATASET

Model	Precision	Recall	mAP50	mAP50-95	Params	GFLOPs	Inference Speed
YOLOv10n	83.9%	79%	84.7%	63.4%	2.265.363	6.7	1.8 ms/image
YOLOv11n	88%	77%	88%	65%	2.583.712	6.5	1.1 ms/image
YOLOv12n	90.1%	76%	87%	64.3%	2.509.904	5.8	1.6 ms/image
YOLOv26n	85%	78%	85%	62%	**2.376.396**	**5.2**	**0.8 ms/image**
YOLOv12n-SKNet	**94.8%**	**83.1%**	**90.6%**	**69.8%**	2.897.872	7.7	1.6 ms/image

Table III presents a comparative evaluation of YOLO variants on the KITTI dataset. YOLOv12n achieved 90.1% precision, 76.0% recall, 87.0% mAP50, and 64.3% mAP50-95, while YOLOv12n-SKNet improved these results to 94.8% precision, 83.1% recall, 90.6% mAP50, and 69.8% mAP50-95. To further evaluate the contribution of the attention mechanism, Table IV compares YOLOv12n-SKNet against variants using Squeeze-and-Excitation (SE) [36] and Convolutional Block Attention Module (CBAM) [37].

TABLE IV. COMPARISON WITH OTHER ATTENTION MECHANISM

Model	Precision	Recall	mAP50	mAP50-95
YOLOv12n-SE	93.4%	82.5%	90.2%	66.4%
YOLOv12n-CBAM	94%	83%	90%	69%
YOLOv12n-SKNet	**94.8%**	**83.1%**	**90.6%**	**69.8%**

As shown in Table IV, YOLOv12n-SKNet achieves the highest results across all evaluation metrics compared to the SE and CBAM variants. These results confirm that SKNet's adaptive kernel selection provides more effective feature representation for autonomous driving object detection than channel-reweighting or spatial-channel attention approaches.

Quantitative evaluation in Tables III and IV provides an overall comparison of detection performance across models and attention mechanisms. To further examine these results at a finer granularity, per-class AP50 evaluation on the KITTI test set shows that YOLOv12n-SKNet improves detection across all eight classes. Vehicle categories such as car 96.47% to 97.16%, van 93.36% to 95.41%, truck 97.52% to 98.06%, and tram 96.09% to 97.09% show consistent but moderate gains, while more substantial improvements are observed in small and difficult categories, particularly person_sitting 47.75% to 59.59%, cyclist 83.34% to 87.71%, and pedestrian 77.66% to 81.29%. These results indicate that SKNet's adaptive receptive-field selection is most effective for objects with limited spatial detail caused by distance and scale variation, which directly supports the improvement in recall and mAP reflected in the qualitative detection results shown in Fig. 3.

Fig. 3. *Detection comparison on the KITTI test set: (a) baseline YOLOv12n and (b) YOLOv12n-SKNet*

As shown in Fig. 3, the baseline YOLOv12n detects several objects in the road scene, but some small objects or objects located far from the camera are still missed. In contrast, YOLOv12n-SKNet produces additional detections for several objects that were previously undetected, demonstrating that the integration of SKNet helps improve multi-scale feature representation, particularly for objects with small sizes or limited visual details.

V. CONCLUSION

This study proposes the integration of the Selective Kernel Network into the YOLOv12n architecture to improve object detection performance in autonomous vehicles. Experimental results on the KITTI dataset show that YOLOv12n-SKNet improves precision from 90.1% to 94.8%, recall from 76.0% to 83.1%, mAP50 from 87.0% to 90.6%, and mAP50-95 from 64.3% to 69.8%. The additional computational cost remains manageable because the parameter count increases from 2.509M to 2.897M and GFLOPs increased from 5.8 to 7.7, while the inference time remains 1.6 ms per image. These findings indicate that YOLOv12n-SKNet provides consistent accuracy improvement with a measurable overhead for real-time object detection scenarios.

Future development can focus on evaluating the proposed model on additional benchmark datasets such as nuScenes and BDD100K to assess generalization beyond the KITTI domain, as well as enhancing the model's adaptability to more diverse complex environments, such as nighttime conditions, fog, rain, snow, and visibility changes due to adverse weather.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Machine Learning and Natural Language Processing Research Laboratory and the Intelligent Systems and Informatics (ISI) Research Group at Siliwangi University,

Tasikmalaya, West Java, Indonesia, for their academic support and constructive research discussions throughout this work. The authors also acknowledge the Department of Electrical Engineering at UIN Sunan Gunung Djati Bandung, Indonesia, for its valuable support and contribution to the completion of this research.

REFERENCES

[1] A. Biswas and H. C. Wang, "Autonomous Vehicles Enabled by the Integration of IoT, Edge Intelligence, 5G, and Blockchain," Feb. 01, 2023, *MDPI*. doi: 10.3390/s23041963.

[2] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies," *IEEE Access*, vol. 8, pp. 58443–58469, 2020, doi: 10.1109/ACCESS.2020.2983149.

[3] J. Wei, A. As'arry, K. Anas Md Rezali, M. Zuhri Mohamed Yusoff, H. Ma, and K. Zhang, "A Review of YOLO Algorithm and Its Applications in Autonomous Driving Object Detection," 2025, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2025.3573376.

[4] J. Yoo, D. Lee, I. Chung, D. Kim, and N. Kwak, "What, How, and When Should Object Detectors Update in Continually Changing Test Domains?," 2024. [Online]. Available: https://github.com/natureyoo/ContinualTTA

[5] L. Liu *et al.*, "Deep Learning for Generic Object Detection: A Survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Feb. 2020, doi: 10.1007/s11263-019-01247-4.

[6] M. Trigka and E. Dritsas, "A Comprehensive Survey of Machine Learning Techniques and Models for Object Detection," Jan. 01, 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/s25010214.

[7] R. Sapkota *et al.*, "YOLO advances to its genesis: a decadal and comprehensive review of the You Only Look Once (YOLO) series," *Artif. Intell. Rev.*, vol. 58, no. 9, Sep. 2025, doi: 10.1007/s10462-025-11253-3.

[8] Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-Centric Real-Time Object Detectors," Feb. 2025, [Online]. Available: http://arxiv.org/abs/2502.12524

[9] E. Liang, D. Wei, F. Li, H. Lv, and S. Li, "Object detection model of vehicle-road cooperative autonomous driving based on improved YOLO11 algorithm," *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-18263-9.

[10] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," *Artif. Intell. Rev.*, vol. 57, no. 4, Apr. 2024, doi: 10.1007/s10462-024-10721-6.

[11] X. Li, W. Wang, X. Hu, and J. Yang, "Selective Kernel Networks," *ArXiv*, Mar. 2019, [Online]. Available: http://arxiv.org/abs/1903.06586

[12] Y. Sun, Z. Sun, and W. Chen, "The Evolution of Object Detection Methods," *Eng. Appl. Artif. Intell.*, vol. 133, Jul. 2024, doi: 10.1016/j.engappai.2024.108458.

[13] L. Xu, "Data Shift of Object Detection in Autonomous Driving," Aug. 2025, [Online]. Available: http://arxiv.org/abs/2508.11868

[14] M. Jamali, P. Davidsson, R. Khoshkangini, M. G. Ljungqvist, and R. C. Mihailescu, "Context in object detection: a systematic literature review," *Artif. Intell. Rev.*, vol. 58, no. 6, Jun. 2025, doi: 10.1007/s10462-025-11186-x.

[15] S. Kang, Z. Hu, L. Liu, K. Zhang, and Z. Cao, "Object Detection YOLO Algorithms and Their Industrial Applications: Overview and Comparative Analysis," Mar. 01, 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/electronics14061104.

[16] L. T. Ramos and A. D. Sappa, "A Decade of You Only Look Once (YOLO) for Object Detection: A Review," 2025, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2025.3630988.

[17] N. M. Alahdal, F. Abukhodair, L. H. Meftah, and A. Cherif, "Real-time Object Detection in Autonomous Vehicles with YOLO," in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 2792–2801. doi: 10.1016/j.procs.2024.09.392.

[18] D. Asmawati and C. Fathichah, "Integrasi YOLOv9 Dan Fine-Tuned Segment Anything Model Untuk Pengenalan Komponen Alat Pengukur Analog," *TELKA*, vol. 12, no. 1, pp. 2502–1982, 2026, doi: 10.15575/telka.v12n1.55-65.

[19] M. El-Geneedy, H. El-Din Moustafa, H. Khater, S. Abd-Elsamee, and S. A. Gamel, "Advanced real-time detection of acute ischemic stroke using YOLOv12, YOLOv11, and YOLO-NAS: a comparative study for multi-class classification," *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-18997-6.

[20] Y. Cao, C. Li, Y. Peng, and H. Ru, "MCS-YOLO: A Multiscale Object Detection Method for Autonomous Driving Road Environment Recognition," *IEEE Access*, vol. 11, pp. 22342–22354, 2023, doi: 10.1109/ACCESS.2023.3252021.

[21] Z. Lv, R. Wang, Y. Wang, F. Zhou, and N. Guo, "Road Scene Multi-Object Detection Algorithm Based on CMS-YOLO," *IEEE Access*, vol. 11, pp. 121190–121201, 2023, doi: 10.1109/ACCESS.2023.3327735.

[22] S. Cui *et al.*, "DAN-YOLO: A Lightweight and Accurate Object Detector Using Dilated Aggregation Network for Autonomous Driving," *Electronics (Switzerland)*, vol. 13, no. 17, Sep. 2024, doi: 10.3390/electronics13173410.

[23] H. Ren, F. Jing, and S. Li, "DCW-YOLO: Road Object Detection Algorithms for Autonomous Driving," *IEEE Access*, vol. 13, pp. 125676–125688, 2025, doi: 10.1109/ACCESS.2024.3364681.

[24] F. Li, Y. Zhao, J. Wei, S. Li, and Y. Shan, "SNCE-YOLO: An Improved Target Detection Algorithm in Complex Road Scenes," *IEEE Access*, vol. 12, pp. 152138–152151, 2024, doi: 10.1109/ACCESS.2024.3481642.

[25] Z. Jia, S. Sun, G. Liu, and B. Liu, "MSSD: multi-scale self-distillation for object detection," *Visual Intelligence*, vol. 2, no. 1, Dec. 2024, doi: 10.1007/s44267-024-00040-3.

[26] Y. Yang, S. Yang, and Q. Chan, "LEAD-YOLO: A Lightweight and Accurate Network for Small Object Detection in Autonomous Driving," *Sensors*, vol. 25, no. 15, Aug. 2025, doi: 10.3390/s25154800.

[27] F. Wei and W. Wang, "SCCA-YOLO: A Spatial and Channel Collaborative Attention Enhanced YOLO Network for Highway Autonomous Driving Perception System," *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-90743-4.

[28] A. A. Murat and M. S. Kiran, "A comprehensive review on YOLO versions for object detection," Oct. 01, 2025, *Elsevier B.V.* doi: 10.1016/j.jestch.2025.102161.

[29] A. Aradea, I. Supriana, and K. Surendro, "Self-adaptive model based on goal-oriented requirements engineering for handling service variability," *Journal of Information and Communication Technology*, vol. 19, no. 2, pp. 225–250, Apr. 2020, doi: 10.32890/jict2020.19.2.4.

[30] Aradea, I. Supriana, and K. Surendro, "ARAS: adaptation requirements for adaptive systems," *Automated Software Engineering*, vol. 30, no. 1, p. 2, 2022, doi: 10.1007/s10515-022-00369-3.

[31] Aradea, I. Supriana, and K. Surendro, "Self-adaptive software modeling based on contextual requirements," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 16, no. 3, pp. 1276–1288, Jun. 2018, doi: 10.12928/TELKOMNIKA.v16i3.7032.

[32] Aradea, Rianto, N. Herlina, and I. Hoeronis, "Self-Learning Model for Pattern Recognition in Vision System Based on Adaptive Kernel," *Informatica (Slovenia)*, vol. 49, no. 14, pp. 157–170, 2025, doi: 10.31449/inf.v49i14.7272.

[33] M. A. Butt *et al.*, "Convolutional Neural Network Based Vehicle Classification in Adverse Illuminous Conditions for Intelligent Transportation Systems," *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/6644861.

[34] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[35] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets Robotics: The KITTI Dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[36] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," *ArXiv*, May 2019, [Online]. Available: http://arxiv.org/abs/1709.01507

[37] Aradea, Rianto, M. Al-Husaini, D. Lestari, P. Pangestu, and G. F. Nugraha, "Optimizing Corrosion Object Detection Model on Metal Objects Based on Bami-Yolov5s," *ICIC Express Letters*, vol. 19, no. 10, pp. 1071–1080, Oct. 2025, doi: 10.24507/icicel.19.10.1071.