YOLOv11-ASM: Enhancing Real-Time Object Detection for Autonomous Driving

Yi Wu, Yuhang Jin *

College of Science, Jiangsu University of Science and Technology, Zhenjiang, China, 212100

* Corresponding Author Email: 15751148542@163.com

Abstract. This paper presents YOLOv11-ASM, a lightweight object detection model designed for autonomous driving, aiming to enhance detection accuracy and robustness in complex environments. The model integrates a Spatial-Aware Feature Modulation Network (SAFMN) to dynamically adjust feature representations via spatial context, improving adaptability. A Multi-Level Cross Attention (MLCA) mechanism is introduced to effectively fuse multi-scale features, while an Adaptive Task-Focused Loss (ATFL) function is designed to optimize regression performance, particularly benefiting small object detection and mitigating class imbalance. Experimental evaluations on the COCO128 dataset demonstrate that YOLOv11-ASM achieves superior performance compared to baseline models, with a precision of 0.939, recall of 0.783, and mAP@50 and mAP@50:95 reaching 0.919 and 0.789, respectively. The proposed model improves mAP@50 by 3.5%–5.9% over existing approaches, showcasing its strong generalization and real-time detection capabilities. These results highlight the model's potential for deployment in real-world autonomous systems requiring accurate and efficient object detection.

Keywords: Automatic Driving, Real-Time Monitoring Of Targets, YOLOv11, ATFL Loss Function, MLCA Mechanism.

1. Introduction

AI and computer vision advancements have spurred significant research in intelligent transportation, particularly in autonomous driving. Accurate target detection is crucial for vehicle path planning and safety decisions, requiring real-time, high-accuracy, low-latency, and robust algorithms to enhance autonomous systems [1]. Although deep learning has advanced detection, practical deployment still faces challenges. Detection accuracy drops in complex road scenarios [2], and models struggle with small, distant, or obscured targets, hindering real-time and reliable performance [3].

The current target detection algorithms for autonomous driving are categorised into two primary types: one-stage and two-stage algorithms. Two-stage algorithms, including **Regions with Convolutional Neural Networks (**R-CNN), Fast-RCNN and Faster-RCNN, have gained significant popularity. Girshick et al. [4] proposed R-CNN, which attains high-precision object detection. Despite its high detection accuracy, the algorithm's speed is a drawback. Li et al. [5] used Fast-RCNN for pedestrian detection, which improves the speed compared to R-CNN, but the large computational volume and high resource consumption remain unsolved. Xiao et al. [6] used Faster-RCNN to reduce the computation time by generating candidate regions, but there are still limitations in deformed, rotated and camouflaged object detection. Two-stage algorithms achieve high detection performance through candidate region generation, classification, and regression but suffer from high computational overhead and slow inference, making them unsuitable for real-time detection. Their performance fluctuates in complex scenarios due to inefficiencies in region generation. In contrast, one-stage algorithms like You Only Look Once(YOLO) bypass this process, directly regressing bounding boxes and categories, improving inference speed and making them ideal for real-time applications. End-to-end training further boosts speed and efficiency by eliminating multi-stage bottlenecks. One-stage algorithms offer clear advantages in real-time performance, efficiency, and adaptability, making them a key focus in target detection research. Meftah et al. [7] explored YOLOv1 for autonomous driving, treating object detection as a regression problem through a single-channel pipeline for real-time

detection. However, YOLOv1 struggles to balance detection between large and small targets. Hoffmann et al. [8] used YOLOv3 for autopilot detection, achieving high accuracy in complex scenes but struggling with occlusion and fast-moving targets, leading to localization errors. Hoffmann et al. [9] improved small target detection and reduced complexity with YOLOv5, but the performance gap for large objects in complex traffic remains. Yang et al. [10] optimized inference speed and reduced computational complexity with YOLOv8, making it suitable for resource-constrained devices, though multi-object localization remains challenging. Geetha et al. [11] employed YOLOv10 for autopilot detection, improving complex target detection but facing limitations in poor lighting or weather. To address these, Rabinandan Kishor [12] used YOLOv11 for optimal detection accuracy and recall in high-precision tasks. However, this may cause significant latency in resource-intensive applications, failing to meet real-time requirements. Therefore, more efficient target detection algorithms are needed to ensure real-time performance, accuracy, and robustness in complex autonomous driving environments.

This paper presents a refined YOLOv11-ASM detection model to improve accuracy and robustness in autonomous driving. The Spatial-Aware Feature Modulation Network (SAFMN) module adjusts feature distribution dynamically, enabling adaptation to changes in illumination, viewing angle, and scene. The Multi-Level Cross Attention (MLCA) mechanism enhances information fusion across scales and semantic levels, improving detection in multi-scale target recognition and complex backgrounds. The Adaptive Task-Focused Loss (ATFL) function dynamically adjusts loss weights, boosting small target recognition and addressing category imbalance to reduce misdetection. The integration of these modules significantly enhances YOLOv11's detection performance, providing a more accurate, efficient, and stable solution for autonomous driving systems.

2. YOLOv11

The latest YOLOv11 [12] achieves a balance between speed, accuracy, and efficiency through architectural innovations and optimized training. YOLOv11 maintains the end-to-end approach, completing tasks with a single forward inference. A new model system (YOLOv11n/s/m/l/x) adjusts depth and width to meet the needs of both edge devices and the cloud. Its dynamic resource allocation optimizes computation paths based on image complexity, and NMS-free training simplifies post-processing, boosting real-time performance. The unified multitasking architecture supports concurrent target detection, instance segmentation, and pose estimation, efficiently utilizing shared feature backbones and task-specific headers. The YOLOv11 framework is shown in Fig. 1.

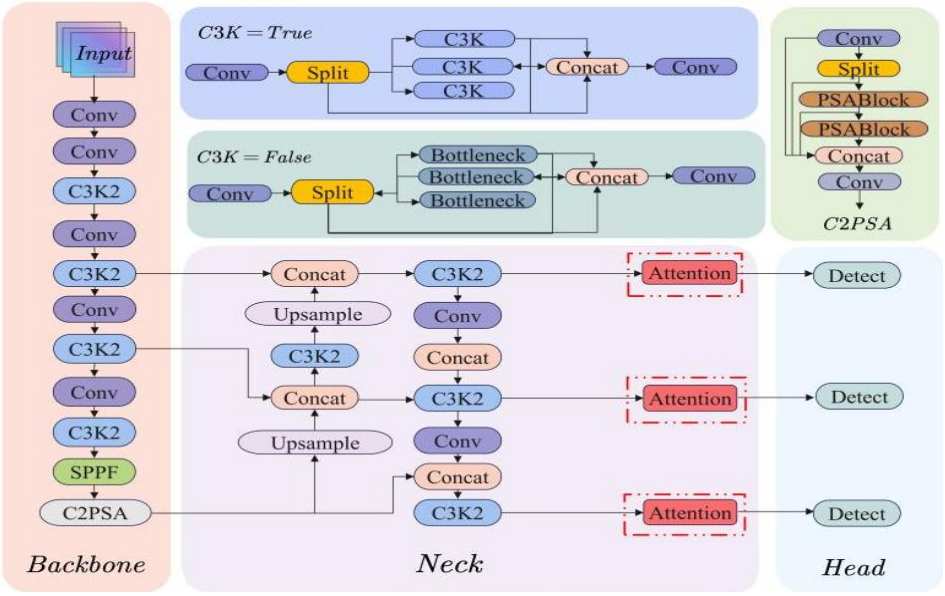

Fig 1. YOLOv11 network architecture.

YOLOv11's architecture includes four main modules. The input module uses adaptive scaling with Mosaic data augmentation and multi-scale training for robustness across resolutions. The lightweight C3k2 backbone features SPPF fast pyramid pooling for multi-scale fusion. Optional MobileNetV4 or EfficientViT architectures optimize feature extraction with partial convolution (PConv). The Neck combines SPPF+C2PSA for bidirectional FPN+PAN interactions, enhancing semantic and shallow feature richness. The Head decouples classification and regression heads; the classification head uses DWConv and channel attention, while the regression head applies distribution matching loss. A new OBB head efficiently predicts rotating frames using polar coding.

While YOLOv11 excels in detection, it faces challenges in autonomous driving, such as target size variation, with many small targets complicating detection. Complex environments require both low-level texture and high-level semantic information, but conventional Backbone features often fail to capture these effectively. Category imbalance, with background dominance and scarce positive samples, also affects performance. Factors like image blurring and resolution further reduce prediction confidence, impacting detection stability and accuracy. Thus, YOLOv11 needs further optimization.

3. Improved YOLOV11 algorithm with multi-strategy integration

To tackle challenges in autonomous driving target detection, this paper proposes the YOLOv11-ASM model to improve robustness and accuracy. Key challenges include target scale variation, background interference, occlusion, and dynamic targets [13]. The SAFMN module optimizes feature distribution through spatially-aware modulation, adapting to changing conditions and improving detection generalization. The MLCA mechanism enhances feature extraction by integrating high-level semantics and low-level details, improving target differentiation in complex scenes and multi-scale detection. The ATFL loss function dynamically adjusts weights, improving small target detection and addressing category imbalance, reducing misdetection and omission. These enhancements boost YOLOv11's stability and robustness, providing reliable target sensing. The enhanced YOLOv11 model is shown in Fig. 2.

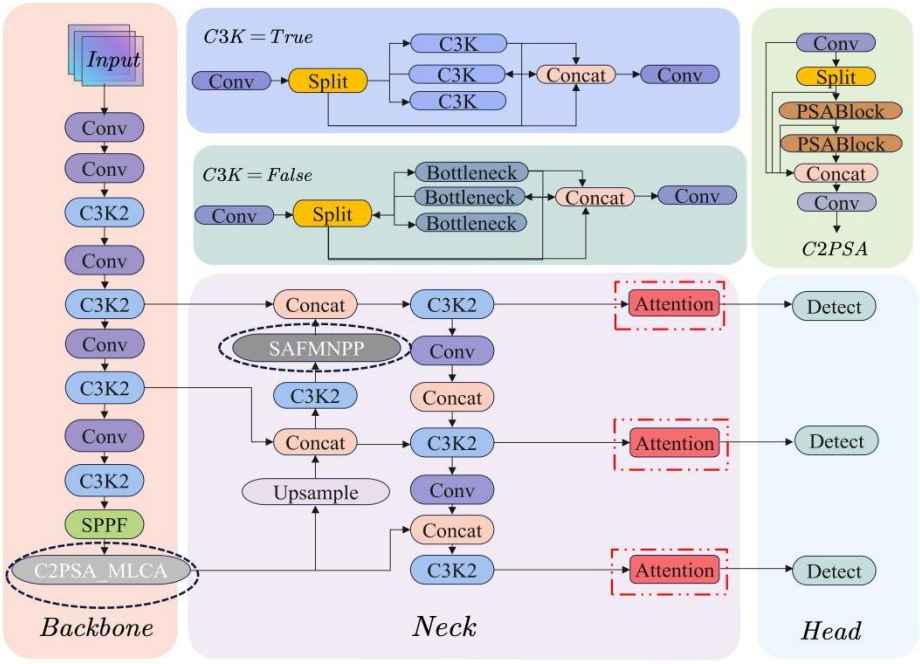

Fig 2. Schematic diagram of the improved YOLOv11 network structure.

3.1. SAFMN module

The SAFMN [14] model, based on a lightweight reconstruction network for image super-resolution, combines convolution's local modeling with the attention mechanism's selective

expression. It enhances key region focus and suppresses redundancy through multi-scale feature representation and dynamic spatial modulation. The SAFM layer generates informative feature maps by modeling independent channels and adaptively fusing scales. The SAFMN module integrates multi-scale feature inputs and uses a convolutional channel mixer to encode local contexts and fuse channel information, improving spatial modeling while keeping computational costs low.The framework diagram of SAFMN is shown in Fig. 3.

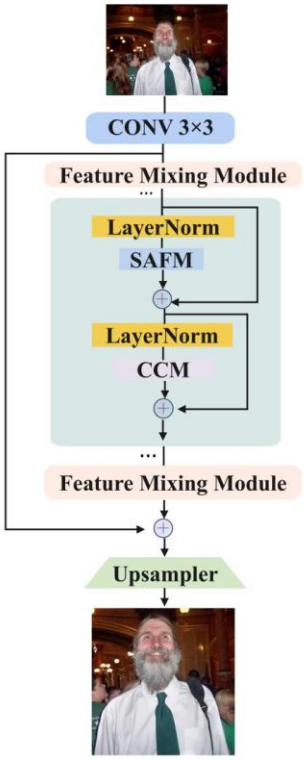

Fig 3. SAFMN architecture diagram

The SAFMN model, based on a lightweight image super-resolution reconstruction network, combines convolution's local modeling with the attention mechanism for selective expression. It enhances key region focus and suppresses redundancy through multi-scale feature representation and dynamic spatial modulation. The SAFM layer generates more informative feature maps by modeling independent channels and adaptively fusing scales. The SAFMN module integrates multi-scale feature inputs and uses a convolutional channel mixer to encode local contexts and fuse channel information, improving spatial modeling with low computational cost.

In terms of structural implementation, the first step in the process is to divide the input feature map $X \in R^{C\times H\times W}$ into four subgroups $[X_0, X_1, X_2, X_3]$ along the channel. In this regard, X_0 retains the original scale for local detail extraction, while the remaining three sub-groups are down-sampled by different multiplicities and then extracted by 3x3 depth-separable convolution to extract the contextual features. These are finally up-sampled back to the original size.

$$[X_0, X_1, X_2, X_3] = Split(X) \tag{1}$$

$$\hat{X}_0 = DW - Conv_{3\times3}(X_0) \tag{2}$$

$$\hat{X}_i = \uparrow_p (DW - Conv_{3\times3}(\downarrow_{\frac{p}{2^i}} (X_i))), 1 \le i \le 3 \tag{3}$$

Multi-scale features are spliced together and fused by 1×1 convolution, and then activated by the GELU activation function. This process generates a spatial modulation map A, which is then multiplied element by element with the original feature map. This results in adaptive enhancement.

$$A = \phi(Conv_{1\times1}(Concat([\hat{X}_0, \hat{X}_1, \hat{X}_2, \hat{X}_3]))) \tag{4}$$

$$\overline{X} = A \otimes X \tag{5}$$

Where $\phi(\cdot)$ denotes the GELU activation function and $\otimes$ denotes element-by-element multiplication. This modulation strategy enables the model to dynamically adjust the spatial attention distribution, thereby enhancing the response strength of the target region and suppressing the background interference.

The SAFMN module offers strong multi-scale modeling, using an asymmetric strategy and multi-path context fusion to enhance feature complementarity across different sensory fields. Compared to traditional attention mechanisms like SE and CBAM, SAFMN unifies channel modeling, spatial modeling, and lightweight design, making it ideal for real-time detection tasks.

3.2. MLCA Attention Mechanism

In target detection, feature maps must balance low-level details and high-level semantics, especially in sub-optimal conditions like autonomous driving. Traditional Backbone features often suffer from missing information and blurring. While YOLOv11 uses deep, wide feature extraction, its reliance on local convolution limits inter-channel relationship modeling. To address this, the paper introduces MLCA [15] into YOLOv11's backbone, improving the model's response to key regions and channels. The MLCA module combines local spatial modeling with channel weighting, enabling better multi-scale target and complex texture modeling with minimal computational overhead, especially for small and edge targets. The MLCA framework is shown in Fig. 4.

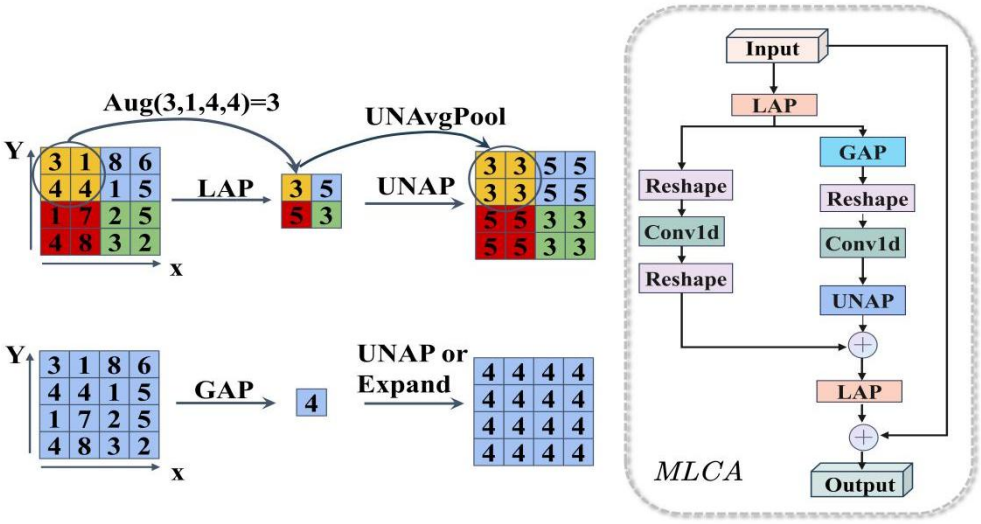

Fig 4. MLCA architecture diagram.

The input feature maps undergo local and global average pooling to extract local and global channel features. One-dimensional convolution is then applied to model cross-channel dependencies. To address the limitations of a fixed convolution kernel, the kernel size k is adaptively determined using the following formula:

$$k = \phi(C) = \left| \frac{\log_2(C)}{\gamma} + b \right|_{odd} \tag{6}$$

In this context, C denotes the number of channels, γ, b represent hyper parameters, and the $|\cdot|_{odd}$ symbol indicates the nearest odd number. This strategy adjusts the convolutional receptive field based on the number of channels to improve local channel modeling. The fused attention vectors, activated by Sigmoid, enhance key region responses and suppress redundant information, thereby boosting detection performance.

Compared to traditional attention mechanisms, MLCA balances performance and efficiency more effectively. Unlike SE, which focuses on global information, MLCA uses local average pooling to improve regional structure perception. It enhances features like CBAM but with lower computational complexity. MLCA also reduces inference overheads compared to CA and other location-sensitive mechanisms, making it ideal for real-time detection. By merging local and global context in channel modeling, MLCA improves target recognition across scales through dual-scale pooling and fusion, while using shared convolutions to reduce parameters, balancing performance and deployability. This efficient design significantly contributes to the field.

3.3. ATFL loss function

In autonomous driving target detection, category imbalance leads to low confidence in predictions for many positive samples, causing overfitting on easy samples and misdetection of difficult ones. Traditional Cross Entropy loss struggles with this, and Focal Loss [16] uses a fixed focusing factor (γ) that lacks flexibility. To address this, this paper proposes integrating ATFL into YOLOv11's loss function, applying larger gradients to difficult samples to improve learning of low-confidence targets.

The traditional Cross Entropy loss function treats all samples equally during the training process and is defined as follows:

$$CE(p_t) = -log(p_t) \tag{7}$$

Where denotes p_t the predictive probability of the model for the true category. The function under discussion produces a non-zero gradient when the samples are correctly classified $(p_t \to 1)$, resulting in a large number of 'easy samples' continuing to participate in the loss accumulation. This, in turn, inhibits the model's ability to focus on difficult samples.

To address this issue, Lin et al. [16] proposed the Focal Loss function, which involves the suppression of the gradient contribution of high-confidence samples by introducing a modulation factor $(1-p_t)^{\gamma}$, thereby enhancing the model's training strength on difficult samples:

$$FL(p_t) = -(1-p_t)^{\gamma} \log(p_t) \tag{8}$$

The parameter γ is employed to regulate the degree of suppression. Focal loss can be regarded as a structural extension of cross-entropy. On the basis of this, the present paper introduces the mechanism of 'adaptive thresholding' and constructs ATFL. The method is characterised by the setting of a confidence threshold τ, which results in the retention of samples exhibiting confidence lower than τ during the loss calculation process. This facilitates the accentuation of low confidence and difficult-to-classify regions. The definition of this function is given below:

When $p_t < \tau$

$$ATFL(p_t) = -(1-p_t)\log(p_t) \tag{9}$$

Furthermore

$$ATFL(p_t) = 0 \tag{10}$$

It is possible to make a dynamic adjustment to the value of $\tau \in (0,1)$ based on the phase of model training.

The ATFL, as introduced in this paper, improves upon Focal Loss with a fixed γ by adaptively adjusting the loss term based on prediction confidence, enhancing focus on low-confidence targets. It also adapts to category imbalance, guiding the model to form a more robust discriminative boundary, outperforming traditional loss functions.

4. Experiment and Analysis

4.1. Experimental environment

4.1.1. Introduction to the dataset

This study uses the COCO128 dataset [17] for target detection training and evaluation. A subset of the Microsoft COCO dataset, COCO128 contains 128 images across 80 categories, including pedestrians, vehicles, and animals, representing diverse scenes. Created by Ultralytics from COCO train2017, it is widely used for fast YOLO model testing. The dataset is split into training, validation, and test sets in an 8:1:1 ratio, with YOLO-formatted annotations. Despite its small size, COCO128's standardized annotations and broad category coverage make it valuable for evaluating lightweight detection models in early validation stages.

4.1.2. Simulation environment

The experimental environment configuration is shown in Table.1.

Table 1. Experimental hardware environment.

Hardware Configuration	Parameters	
Device Processor	12th Gen Intel(R) Core(TM) i5-12500H	2.50 GHz
Configured Environment	Pytorch 2.0.0, Python 3.9.0	
CUDA	11.8	
GPU	NVIDIA Geforce RTX 3080	
Operating system	Windows 11	

4.1.3. Parameter settings

All experiments were conducted under the same environment configuration, and the experimental parameter settings are detailed in Table.2.

Table 2. Experimental parameters.

Parameter Name	Parameter Value
Epoch	400
Bach-size	32
Workers	16
Optimizer	SGD
Ir0	0.01
Momentu	0.937
Iamge Size	640×640

4.1.4. Evaluation

The model performance in this experiment is evaluated using four metrics: precision, recall, mean average precision at an IOU threshold of 0.5, and billion floating point operations per second [18].

The precision rate is the ratio of the number of positive samples correctly detected by the model to the total number of samples predicted to be positive by the model.

$$\mathrm{Pr}\,ecision = \frac{TP}{TP + FP} \tag{11}$$

Where TP is the number of positive samples correctly detected by the model; FP is the number of negative samples incorrectly detected as positive by the model.

Recall indicates the number of positive samples correctly identified by the model as a proportion of the total number of all true positives.

$$\mathrm{Re}\,\mathrm{call} = \frac{TP}{TP + FN} \tag{12}$$

Where *FN* is the number of actual positive samples that were incorrectly identified as negative by the model.

The average precision at an IOU threshold of 0.5 is the mean precision across all predicted categories when the intersection-over-union (IOU) threshold is set to 0.5.

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{13}$$

$$AP = \int_0^1 P(r)dr \tag{14}$$

Where *AP* is the area enclosed by the precision-recall ($P-R$) curve and the axis.

GFLOPS Is the computing power of the computer (especially CPU, GPU) and the deep learning model during inference or training?

$$GFLOPS = \frac{TotalFLOPS}{Time(\text{sec}onds)} \times 10^{-9} \tag{15}$$

Where Total Floating Point Operations refers to the total number of floating point operations required for a model or task to complete inference. Reasoning Time is the time taken for the task or model to perform inference.

4.2. Model validation

In order to verify the validity of the model, we collated the loss function changes of the model during the training process and recorded them in Fig. 5.

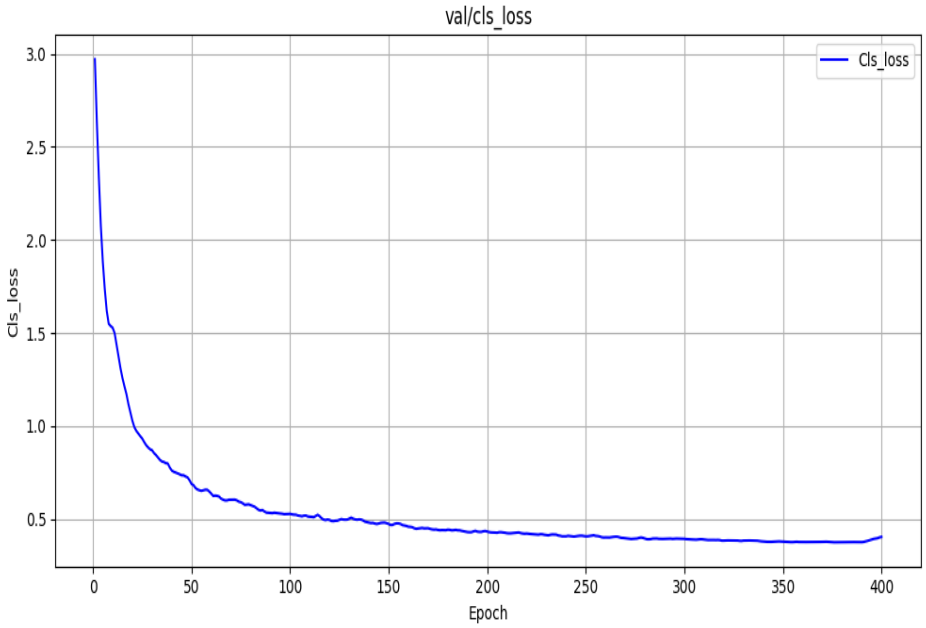

Fig 5. Loss plot.

As shown in Fig. 5, the classification loss (val/cls_loss) decreases rapidly in the first 50 rounds, from around 3.0 to below 1.0, indicating that the model captures categorical features early in training. Afterward, the loss decreases more slowly, stabilizing around the 250th round and converging to about 0.4. This indicates that the model has nearly converged, with gradual improvement in classification ability throughout the training process.

To verify the validity of the model, we collated the model's mAP@50 changes during training and plotted them in Fig. 6.

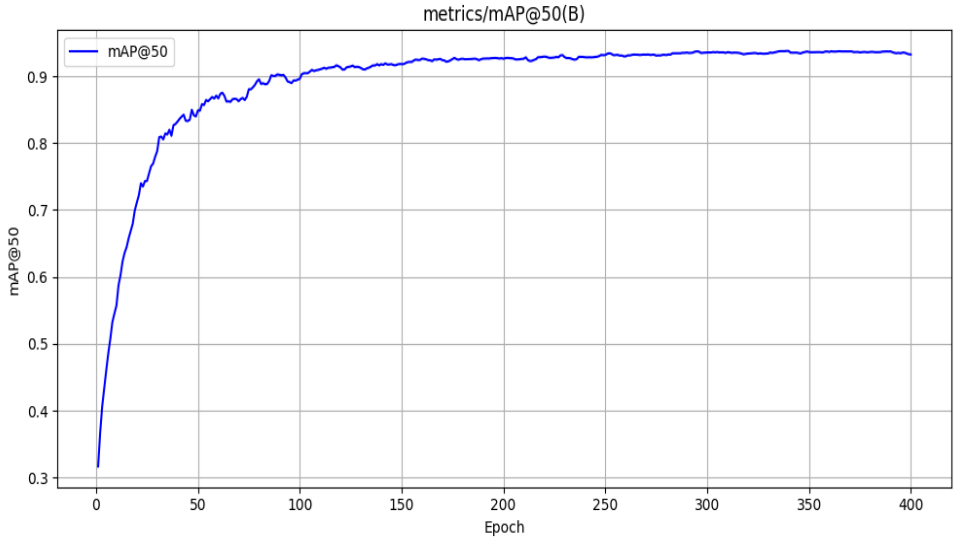

Fig 6. mAP@50 plot.

As training progresses, the mAP@50 (B) curve shows significant improvement. Initially, mAP@50 rises rapidly from around 0.3 to nearly 0.7, indicating that the model quickly learns key features and improves positive sample recognition. As training continues, the growth slows, and after about 200 rounds, performance stabilizes, converging close to 0.9. This suggests that the model's detection capability, especially for small and complex targets, is optimized as training progresses. We then collate the YOLOv11-ASM for four metrics, including six categories such as person,car, and record them in Table.3.

Table 3. Results of the indicator assessment.

Targets	mAP@50	P	R	mAP@50:95
Person	0.954	1	0.807	0.829
Bicycle	0.852	0.756	0.52	0.583
Car	0.721	1	0.371	0.411
Motorcycle	0.995	0.96	1	0.995
Bus	0.995	0.972	1	0.919
Train	0.995	0.943	1	0.995
All	0.919	0.939	0.783	0.789

As shown in Table.3, the YOLOv11-ASM model demonstrates strong performance in multi-category target detection. It achieves an average precision of 0.939, recall of 0.783, mAP@50 of 0.919, and mAP@50:95 of 0.789, indicating high accuracy and localization ability. For motorbikes and trains, it achieves over 0.96 precision with recall of 1, and mAP@50 and mAP@50:95 both reach 0.995, showing excellent performance in stable categories. However, recall for cars and bicycles is lower (0.371 and 0.52) with mAP@50:95 of 0.411 and 0.583, likely due to occlusion, scale variation, and weak textures. Despite these fluctuations, the overall detection remains high, with no significant misdetections, suggesting effective training convergence. YOLOv11-ASM excels in recognition accuracy, recall, and mAP, making it ideal for tasks with clear structural features. Future work could focus on optimizing data distribution or addressing category imbalance to improve recall for categories like “car” and “bicycle”, further enhancing versatility and practicality.

4.3. Ablation Experiment

In order to verify the effectiveness of the improvement strategy, this paper conducts ablation experiments by adding the MLCA attention mechanism, SAFMN module and replacing the ATFL loss function in turn, and records the metric evaluation results obtained from the experiments in Table.4.

Table 4. Ablation experiment results.

Models	mAP@50	P	R	mAP@50:95	GFLOPS
YOLOv11	0.886	0.915	0.771	0.784	6.4
YOLOv11+MLCA	0.892	0.935	0.772	0.786	6.4
YOLOv11+MLCA+SAFMN	0.907	0.937	0.774	0.787	16.7
YOLOv11+MLCA+SAFMN+ATFL	0.919	0.939	0.783	0.789	16.7

As shown in Table.4, ablation experiments evaluated the contribution of MLCA, SAFMN, and ATFL to YOLOv11's performance. Introducing MLCA improves precision (P) from 0.915 to 0.935, mAP@50 by 0.6%, and mAP@50:95 by 0.2%, enhancing feature capture and attention to key regions. GFLOPS remains at 6.4, indicating minimal computational increase. Adding SAFMN further improves precision to 0.937, recall (R) to 0.774, mAP@50 to 0.907, and mAP@50:95 to 0.787, though GFLOPS rises to 16.7, reflecting a higher computational demand balanced by improved performance. Finally, the inclusion of ATFL in YOLOv11+MLCA+SAFMN achieves optimal results: precision (P) of 0.939, recall (R) of 0.783, mAP@50 of 0.919, and mAP@50:95 of 0.789. The ATFL module improves regression loss, bounding box accuracy, and target edge fitting, with GFLOPS remaining at 16.7, showing a trade-off between complexity and performance. In summary, YOLOv11-ASM improves detection with MLCA, SAFMN, and ATFL, balancing high accuracy with computational efficiency.

4.4. Comparative Experiments

To evaluate the performance of the improved models in this study, comparative experiments were conducted, including YOLOV11+MDFM+Sparse_Self_Attention+Wise-IOUv1, YOLOV11+LKA+DFF+Wise-IOUv3, YOLOV11+LSKA+Mixstructure+Inner-CIOU, YOLOV11+MSAR+LCA+Inner-EIOU, and YOLOV11+HPA+CGLU+EfficiCIoU. All models were trained on the same dataset with identical parameters, such as learning rate, training rounds, and number of threads, to ensure objective and comparable results. The final evaluation results are shown in Table.5.

Table 5. Comparison experiment results.

Models	mAP@50	P	R	mAP@50:95	GFLOPS
YOLOv11+MDFM+Sparse_Self_Attention+Wise-IOUv1	0.860	0.931	0.781	0.758	24.4
YOLOv11+LKA+DFF+Wise-IOUv3	0.884	0.898	0.777	0.743	8.0
YOLOv11+LSKA+Mixstructure+Inner-CIOU	0.870	0.925	0.766	0.753	9.3
YOLOv11+MSAR+LCA+Inner-EIOU	0.868	0.916	0.769	0.757	12.6
YOLOv11+HPA+CGLU+EfficiCIoU	0.868	0.905	0.765	0.751	8.3

As shown in Table.5, the YOLOv11+MLCA+SAFMN+ATFL model outperforms other variants in all key metrics, demonstrating the effectiveness of the multi-module fusion strategy. The model achieves 0.939 precision, 0.783 recall, 0.919 mAP@50, and 0.789 mAP@50:95, outperforming other models by 0.8%-4.1% in precision, 3.5%-5.9% in mAP@50, and improving recall and detection accuracy. While GFLOPS increases to 16.7, reflecting a trade-off between computational demand and performance, the model delivers significantly better results, indicating the added modules enhance performance without sacrificing efficiency. In summary, YOLOv11+MLCA+SAFMN+ATFL excels in feature extraction, spatial awareness, and bounding box regression, making it ideal for complex, multi-scale target detection in practical applications like animal detection and video surveillance.

5. Conclusion

In conclusion, this study presents a comprehensive lightweight detection framework, YOLOv11-ASM, which systematically integrates SAFMN, MLCA, and ATFL modules to address key

challenges in real-time object detection for autonomous driving. The model significantly enhances the robustness and accuracy of YOLOv11, achieving 0.939 precision, 0.783 recall, 0.919 mAP@50, and 0.789 mAP@50:95 on the COCO128 dataset. Compared with baseline models, it improves mAP@50 by 3.5%–5.9% and precision by 0.8%–4.1%. The proposed model integrates multiple lightweight yet effective modules, demonstrating strong detection performance and generalization with low computational cost. These results indicate its practical applicability to real-time detection scenarios. Future work will focus on deployment in resource-constrained platforms and adaptation to more complex environments.

References:

[1] Hemmati M, Biglari-Abhari M, Niar S. Adaptive real-time object detection for autonomous driving systems [J]. Journal of Imaging, 2022, 8(4): 106.

[2] Liu X Y, Chu W F, Hu Y H, et al. enhancing intelligent road target monitoring: a novel BGS-YOLO approach based on the YOLOv8 algorithm [J]. IEEE Open Journal of Intelligent Transportation Systems, 2024, 5: 509-519.

[3] Wang L, Jiang F F, Zhu F Y, et al. Enhanced multi-Target detection in complex traffic using an improved YOLOv8 with SE attention, DCN_C2f, and SIoU[J]. World Electric Vehicle Journal, 2024, 15(12): 586.

[4] Girshick R, Donahue J, Darrell T, et al. Region-based convolutional networks for accurate object detection and segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 38(1): 142-158.

[5] Li J N, Liang X D, Shen S M, et al. Scale-aware fast R-CNN for pedestrian detection [J]. IEEE Transactions on Multimedia, 2018, 20(4): 985-996.

[6] Xiao Y, Wang X Q, Zhang P, et al. Object detection based on Faster R-CNN algorithm with skip pooling and fusion of contextual information[J]. Sensors, 2020, 20(19): 5490.

[7] Meftah L H, Abukhodair F, Cherif A, et al. Real-time object detection in autonomous vehicles with YOLO[J]. Procedia Computer Science, 2024, 246: 2792-2801.

[8] Hoffmann J E, Tosso H G, Justo J F, et al. Real-Time adaptive object detection and tracking for autonomous vehicles[J]. IEEE Transactions on Intelligent Vehicles, 2021, 6(3): 450 - 459.

[9] Mahaur B, Mishra K.K. Small-object detection based on YOLOv5 in autonomous driving systems [J]. Pattern Recognition Letters, 2023, 168: 155-122.

[10] Yang M, Fan X Y. YOLOv8-Lite: a lightweight object detection model for real-time autonomous driving systems [J]. IECE Transactions on Emerging Topics in Artificial Intelligence, 2024, 1(1): 1-16.

[11] Geetha A S, Hussain M, Allen P, et al. Comparative analysis of YOLOv8 and YOLOv10 in vehicle detection: performance metrics and model efficacy[J]. Vehicles, 2024, 6(3): 1364-1382.

[12] Rabinandan K. Performance benchmarking of YOLOv11 variants for real-time delivery vehicle detection: a study on accuracy, speed, and computational trade-offs [J]. Asian Journal Of Researchin Computer Science, 2024, 17(12): 108-122.

[13] Ruan J G, Cui H H, Huang Y H, et al. A review of occluded objects detection in real complex scenarios for autonomous driving [J]. Green Energy and Intelligent Transportation, 2023, 2(3): 100092.

[14] Sun L, Dong J X, Tang J H, et al. Spatially-adaptive feature modulation for efffcient image super-resolution[C].Proceedings Of the IEEE/CVF International Conference on Computer Vision, 2023: 13190-13199.

[15] Wan D H, Lu R S, Shen S Y, et al. Mixed local channel attention for object detection[J]. Engineering Applications of Artificial Intelligence, 2023, 123: 106442.

[16] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 318-327.

[17] Tahir A, Khalid S K, Fadzil L M. Child detection model using YOLOv5 [J]. Journal of Soft Computing and Data Mining, 2023, 4(1): 72-81.

[18] Kwon H, Choi S, Woo W, et al. evaluating segmentation-based deep learning models for real-time electric vehicle fire detection [J]. Fire, 2025, 8(2): 66.