Research on Improvement of Environment Perception Algorithm for Autonomous Driving Vehicles Based on YOLOv5

Yuxi Yang

{yang3611@tongji.edu.cn}

College of Transportation, Tongji University, Cao'an Road, Shanghai, China

Abstract. Autonomous driving relies heavily on vehicle object detection, and YOLOv5s is presently one of the best algorithms for this purpose. However, in extreme environments such as severe weather, cars have poor perception of the environment, and their ability to detect dynamic targets is greatly affected, resulting in low accuracy and poor robustness of YOLOv5 object detection algorithm in pedestrian and vehicle detection. This article proposes an improved YOLOv5s algorithm. Firstly, a selective attention mechanism (SimAM) module is used to weight the output of the convolutional layer, allowing the network to quickly capture regions of interest and suppress irrelevant information; Simultaneously using lightweight convolution GSConv instead of the conventional convolution to compensate for semantic information loss and reduce model complexity; Secondly, adding a shallow detection layer changes the original algorithm's three scale detection to four scale detection, enhancing the learning ability for small-scale targets; Finally, SIoU Loss is used as the bounding box regression loss function to achieve more accurate localization of the predicted boxes. The improved YOLOv5s algorithm was tested on the CARLA simulation dataset, and simulation results showed that the average detection accuracy of the improved model reached 96.67%, which improved the detection accuracy for complex scenes.

Keywords: Object detection, Autonomous driving, YOLOv5s, Lightweight convolution, Multi scale detection, Loss function.

1 Introduction

Autonomous driving technology [1] has become a popular technology in the world, especially in the automotive industry, where its application has reached unprecedented heights. With the deep integration of deep learning technology [2] in the field of computer vision, the intelligence of vehicles is no longer limited to assisted driving functions. More and more practical cases have proven that vehicles equipped with autonomous driving technology can achieve autonomous and safe driving in specific scenarios. However, due to the large amount of computing resources required by deep learning techniques, this presents a challenge in terms of hardware computing power. To address this issue, it is necessary to optimize algorithms to increase the model's detecting speed while ensuring that the detection accuracy is not affected. Such improvements will contribute to the wider application of autonomous driving technology in vehicles and other traffic scenarios, promoting its popularization and development in practical applications.

CONF-MLA 2024, November 21, Adana, Turkey

DOI 10.4108/eai.21-11-2024.2354630

Since Hinton [3] et al. initially introduced the idea of deep learning in 2006, the science of computer vision, particularly object identification methods, has rapidly advanced. Deep learning-based object detection algorithms can be broadly classified into two-stage and one-stage approaches. Two stage detection algorithms, such as R-CNN, Fast R-CNN [4], and Faster R-CNN [5], generate candidate regions and use CNN for classification, but have limitations in training and processing speed. In 2014, Girshick [6] et al. proposed the R-CNN detection algorithm. Although there were breakthroughs in detection accuracy, its multi-stage training and slow processing limited its application. In 2018, Tian [7] et al. improved Faster R-CNN by using multi-level feature fusion, contextual clues, and generating new bounding boxes to enhance the detection speed of small targets. However, due to computational complexity, the detection speed of these methods in practical applications still needs to be improved. One-stage detection methods, for example YOLO [8] and SSD [9], generate anchor boxes directly on the image through a single neural network and perform classification and bounding box regression, simplifying the detection process and significantly improving speed compared to two-stage algorithms. However, this method may result in false positives and false negatives when dealing with small targets, affecting detection accuracy.

At present, the algorithms developed under the YOLO system are the most widely used . Researchers have adopted various strategies, including optimizing network structures to reduce computational complexity, such as improving YOLOv3 and YOLOv4 tiny, and introducing attention mechanisms to focus on targets in images and utilize contextual information. In addition, by improving the loss function, such as transitioning from traditional L1/L2 loss to more advanced EIoU loss, the model can more accurately predict bounding boxes. Meanwhile, multi-scale feature fusion techniques such as Feature Pyramid Network (FPN) and BiFPN are employed to enhance the feature representation of small targets by combining low-level high-resolution features with high-level semantic information. Data augmentation techniques and reasonable training strategies are also employed to increase the generalization ability and detection accuracy of models. When these techniques are combined, object detection models perform better when handling small targets.

This article is based on the YOLOv5 object detection algorithm and uses GSConv+SimAM network structure to replace the original backbone network of YOLOv5; Improve the multi-scale detection mechanism by adding a detection layer for small target vehicles at the head output end; Using SIOU as the loss function of the model; The simulation results demonstrate that the improved YOLOV5 algorithm increases the detection accuracy.

2 Principle of YOLOv5 algorithm

2.1 Overview

Depending on the network depth, automatic YOLOv5, one of the well-liked algorithms in the YOLO series, can be further classified. The shallowest network depth model in the YOLOv5 series is YOLOv5s, which can be implemented on mobile devices and has numerous uses in the field of self-driving cars. The input, backbone network, neck, and head output are the four primary components of the YOLOv5s algorithm. The input improves the algorithm's resilience and inference speed by utilizing adaptive anchor boxes and Mosaic data augmentation technologies; The primary components of the backbone network, which are in charge of feature extraction from input images, are slicing modules and cross-stage local network structures; A feature pyramid and a path aggregation network, which combine features, make up the neck; For multi-scale prediction, the head output makes use of three detection heads with varying scales.

2.2 Network structure

As an important algorithm for first stage object detection, the YOLO series was developed by Ultralytics. YOLOv5 [10] is one of the most popular algorithm models since the emergence of the YOLO series. Its overall performance is weaker than the previous generation YOLOv4 [11], but its speed and flexibility are better than YOLOv4, allowing it to be installed on the majority of terminal devices. The YOLOv5 network structure [11] is divided into four parts: input layer, Backbone (backbone network), Neck network, and Head (output terminal), as shown in Figure 1.

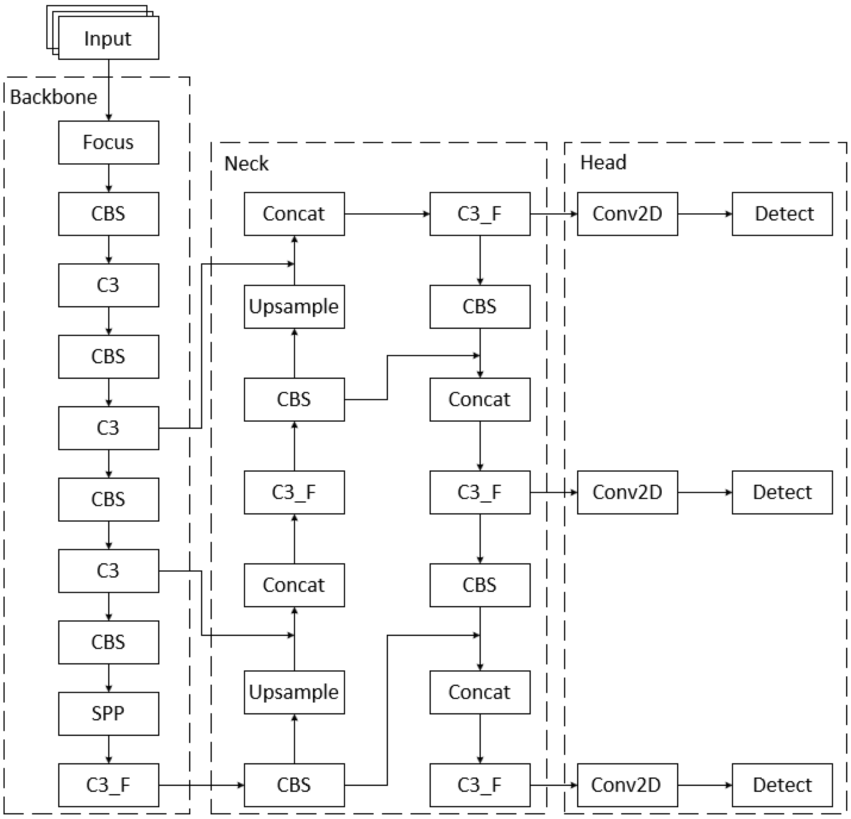


Fig. 1. Network structure of YOLOv5

When inputting images, the Mosaic data augmentation method is used to concatenate the four obtained images to obtain a new image, which improves training efficiency. YOLOv5 can adaptively adjust the value of anchor boxes for different datasets. Compared with YOLOv4, YOLOv5 has less computation and improves overall efficiency.

The original model adopts the C3 network model, which uses three standard convolutions and Bottleneck modules to enhance the learning ability of residual features. The C3 structure has two forms, as shown in Figure 2. In the C3 module, the inner loop represents the stacking of multiple Bottleneck structures. The C3_f module only goes through one basic convolution module. Merge two forms for Concat operation. Implemented algorithm lightweighting and ensured algorithm detection accuracy.

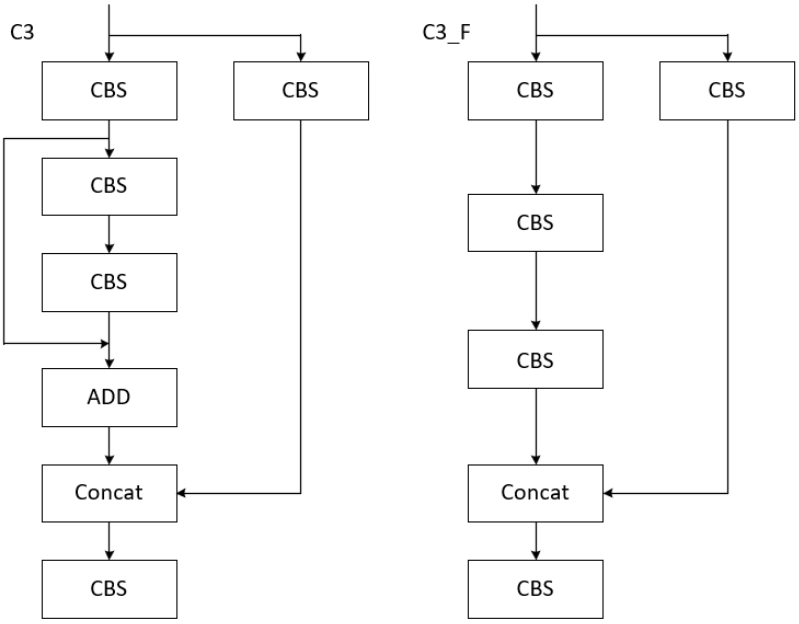


Fig. 2. Two forms of C3 module

In the Backbone section, as shown in Figure 3, they correspond to the important modules in Figure 1, and their internal structures are displayed to facilitate understanding of their working principles. As shown in Figure 3, CBS is used to perform two-dimensional convolution on the signal, which is then normalized by BN before entering the activation function section. As shown in Figure 4, SPP (Spatial Pyramid Pooling) is a spatial pyramid pooling layer that undergoes downsampling through three different max pooling layers before being concatenated and fused. As shown in Figure 5, Focus is used to achieve fast downsampling, which reduces computational complexity and improves inference speed through slicing operations. As the backbone layer, its function is to retrieve feature information from the picture for later use.

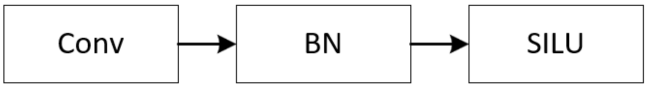


Fig. 3. CBS module structure

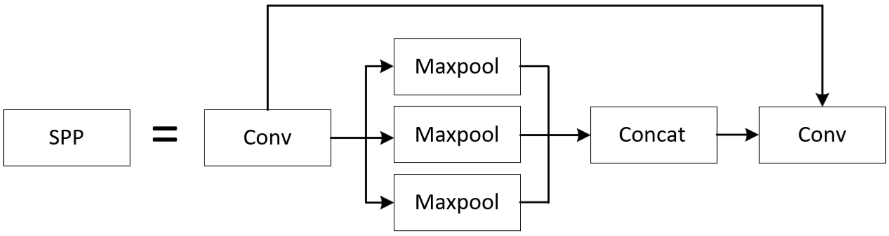


Fig. 4. SPP module structure

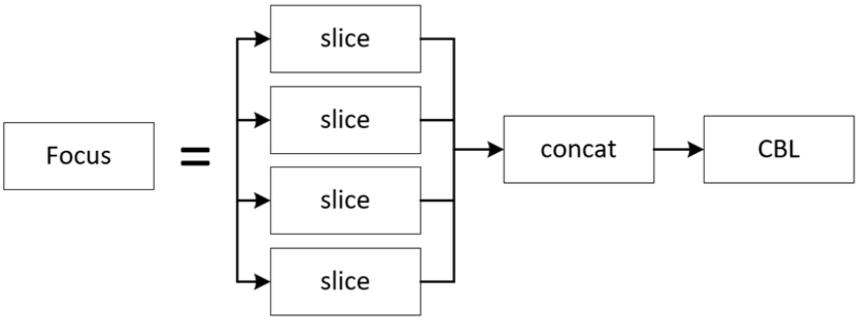


Fig. 5. Focus module structure

The Neck section includes feature pyramid network (FPN) and path aggregation network (PAN) structures. The feature pyramid module is used to generate feature maps of different sizes, consisting of multiple convolutional layers and pooling layers with different kernel sizes. The module for feature fusion intends to combine feature maps with varying scales. The PAN multi-scale feature fusion structure adopts a top-down feature aggregation path, which enhances the model's perception ability of targets of different scales through layer-by-layer aggregation.

The Head module includes a bounding box loss function and non-maximum suppression (NMS). The Head module mainly outputs prediction results, including the position, size, category, and confidence of the target, and then uses NMS to screen the target and obtain the final detection result.

3 Improvement of YOLOv5 algorithm

3.1 Network structure improvement

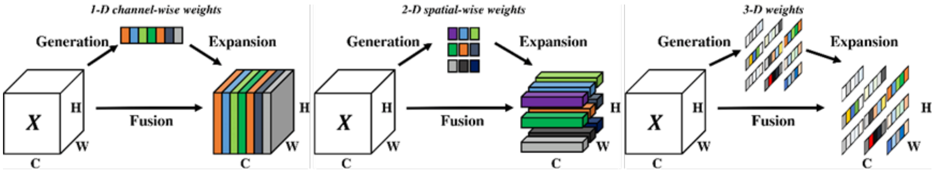
SimAM [12] is a three-dimensional weight attention mechanism proposed by Sun Yat sen University based on famous neuroscience theory, which can simultaneously emphasize the significance of each channel and spatial position feature. It is mainly used to improve the correlation of image features and increase the accuracy of image recognition. It includes similarity calculation and feature interaction.

Local self-similarity of images is the foundation of SimAM. Adjacent pixels in an image typically have a high degree of similarity, whereas distant pixels typically share a lower degree.

SimAM makes use of this capability to create attention weights by figuring out how similar each feature map pixel is to its neighboring pixels, and uses feature interaction to improve the correlation between feature vectors [13].

Specifically, SimAM adopts an attention mechanism based on image segmentation to divide the image into multiple regions, calculate the similarity between regions, and then apply the similarity to the interaction of feature vectors within the regions to enhance the correlation between regions. SimAM can be applied to a number of image recognition applications, including segmentation, detection, and classification.

Experimental results have shown that SimAM can improve the accuracy and resilience of models, particularly when dealing with images that contain complex sceneries and multiple targets. Therefore, the SimAM attention mechanism is incorporated into the Backbone layer of YOLOv5 to improve the model's capacity to focus on features. Specifically, the SimAM module is often inserted into the last convolutional layer of each residual block in the CSPDarknet53 network. In this structure, the SimAM module follows closely behind a convolutional layer and weights its output, emphasizing important feature channels and suppressing useless feature channels. This can increase the model's capacity to express features and raise the accuracy of detection. Figure 6 depicts the SimAM structure.



(a) Channel-wise Attention (b) Spatial-wise Attention (c) 3-D Weights for Attention

Fig. 6. SimAM module structure

The input image experiences channel expansion and spatial compression during feature extraction in the backbone network, which causes semantic information to be lost and channels to become disconnected. As a type of lightweight convolutional network, GSConv can preserve these connections with less time complexity, solving the problem of channel information separation in the calculation of Depthwise Separable Convolution (DSC) neural networks, while reducing model parameters and computational complexity.

This article introduces GSConv convolution blocks into the feature fusion module, replacing the standard convolution SC for feature fusion. A feature fusion network based on GSConv is designed, and a simple and efficient Neck model is constructed to reduce redundant information, parameter count, and model complexity. The feature information extracted from the 2.2 main network is fully utilized to improve the network's feature fusion capability.

The GSConv network uses shuffle to fuse the feature maps extracted by DSC and standard convolution (SC), so that the output of standard convolution is fully integrated into DSC. The main idea is to enhance its performance by adding DSC layers and utilizing the powerful nonlinear expression ability of shuffle layers. This method lowers the computing cost while bringing the convolution calculation's output as near to SC as feasible [14].

The GSConv network model is shown in Figure 7.

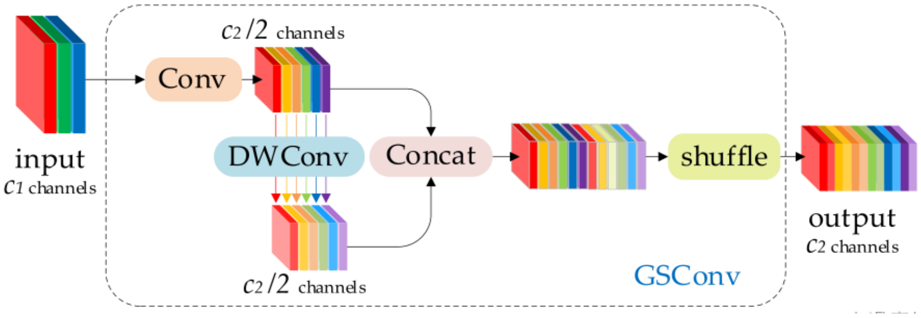


Fig. 7. GSConv module structure

3.2 Improvement of multi-scale detection mechanism

The YOLOv5s algorithm adopts a three scale detection mechanism, which uses 20 × 20, 40 × 40, and 80 × 80 scales to detect large, medium, and small targets, overcoming the limitations of single scale object detection algorithms in detecting smaller targets. However, in real road environments, distant target vehicles occupy fewer pixels in the image, and the 80 × 80 detection layer cannot fully detect smaller vehicle targets. On the basis of YOLOv5s three scale detection, this article adds a detection layer of 160 × 160 size to achieve four scale detection, as shown by the Head in the figure. The new detection layer can retain more position and contour information of small-scale vehicles, which can effectively improve the algorithm's detection ability for small target vehicles.

3.3 Loss Function and Optimization Methods

The original YOLOv5s algorithm uses the CIOU Loss [15] loss function, taking into account the overlapping area, center point distance, and aspect ratio of bounding box regression

$$L_{CIOU} = 1 - IOU + \frac{\rho^2(b,b^{gt})}{C^2} + \alpha v \tag{1}$$

Among them, IOU represents the intersection to union ratio, C^2 is the diagonal distance of the intersection to union ratio, IOU is the intersection to union ratio, and smallest bounding rectangle, $\rho^2(b, b^{gt})$ is the Euclidean distance between the predicted box center point b and the actual box center point b^{gt}, and α is the balance parameter

$$\alpha = \frac{v}{(1-IOU)+v} \tag{2}$$

v is the aspect ratio consistency parameter

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{3}$$

Among them, the aspect ratios of the actual and predicted boxes are represented by $\frac{w^{gt}}{h^{gt}}$ and $\frac{w}{h}$ respectively.

According to the calculation formula of CIOU Loss, the aspect ratio consistency parameter v reflects the difference in aspect ratio, which fails to fully reflect the actual difference between the height and width of the detection target and its confidence, resulting in the CIOU Loss function being unable to effectively learn the similarity between the predicted box and the actual box. In actual images, due to the small proportion of small target vehicles in the entire image, confidence loss can lead to unbalanced training samples. That is, in one image, there are very few high-quality anchor boxes and many low-quality anchor boxes, which results in poor quality anchor boxes having excessive gradients and affecting the training effectiveness of the model.

Based on the above issues, some scholars have proposed EIOU [16] (Efficient Intersection over Union) (reference citation), which clearly evaluates the differences in three geometric factors—overlapping area, center point, and edge length—and breaks down the aspect ratio based on CIOU. At the same time, Focal Loss is implemented to address the issue of unbalanced difficult and easy samples. However, the above method ignores the direction of the expected mismatch between the real box and the forecast box and only takes into account the distance, overlap area, and aspect ratio between the two boxes. Therefore, the SIOU loss function is selected as the bounding box regression loss function for the model in this article. The calculation formula for SIOU is as follows:

$$\mathrm{SIOU_{Loss}} = 1 - \mathrm{IoU} + \frac{\Delta+\Omega}{2} \tag{4}$$

$$\Lambda = 1 - 2\sin^2\left(\arcsin(x) - \frac{\pi}{4}\right) \tag{5}$$

$$\Omega = \sum_{t=w,h}(1 - e^{-w_t})^{\theta} \tag{6}$$

$$\omega_w = \frac{|w-w^{gt}|}{\max(w,w^{gt})}, \omega_h = \frac{|h-h^{gt}|}{\max(h,h^{gt})} \tag{7}$$

The advantage of using the SIOU loss function over the original YOLOv5 loss function is evident in the way it considers good loss factors like overlapping area, center point distance, aspect ratio, and direction angle in the bounding box regression, making the model converge faster and more accurately.

4 Algorithm simulation

4.1 Evaluation criteria

This article uses Average Precision (AP), Recall, Mean Average Precision (mAP), and Frame Per Second (FPS) as evaluation criteria, with an IOU threshold of 0.5 for predicted and real boxes.

One of the frequently used evaluation criteria in object detection tasks is MAP, which is used to measure the accuracy of models at different confidence thresholds. The corresponding calculation formulas are as follows:

$$P = \frac{TP}{TP+FP} \tag{8}$$

$$R = \frac{TP}{TP+FN} \tag{9}$$

$$AP = \int_0^1 P(R)dR \tag{10}$$

$$\mathrm{mAP} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{AP_i} \tag{11}$$

The average precision of a single detection category is represented by the formula's AP (average precision); The likelihood that every anticipated positive sample will turn out to be positive is denoted by P (precision); The likelihood that a sample that is truly positive will also be positive is known as R(recall); N is the total number of categories found; The actual number of cases is shown by TP. FP is the quantity of cases that are falsely positive. The quantity of false counterexamples is denoted by FN. MAP is the average precision mean of all detection categories; $\mathrm{AP_i}$ is the i-th detection category's average accuracy.

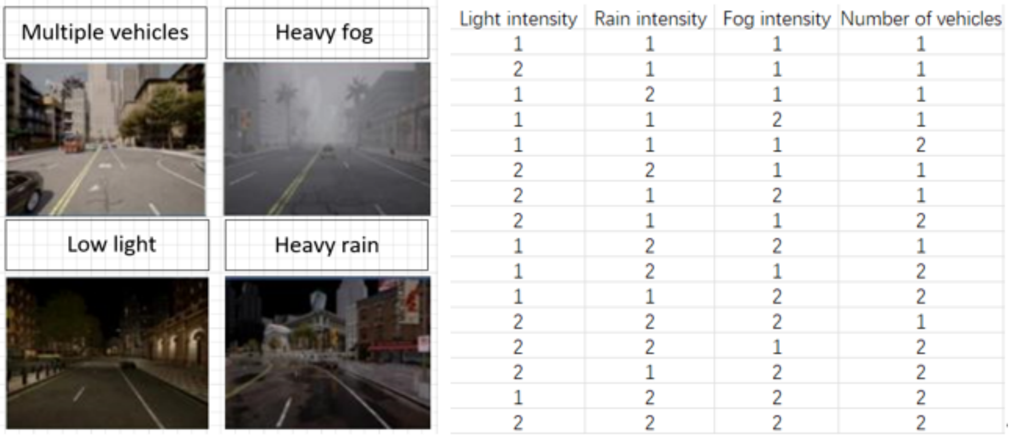
4.2 Simulation environment

This experiment is run on the Windows 11 system, with AMD R7-5800H CPU, 32GB memory, NIVIDA GeForce RTX3080Ti GPU, 12GB video memory, acceleration environment CUDA 11.5.0, CUDNN 8.1.1, integrated development environment PyCharm Professional Edition 2023.2, and programming language Python 3.9.

4.3 Simulation dataset

After connecting to the simulation server, we chose the default map scene in CARLA and added vehicles and pedestrians to the environment, set up autonomous vehicles, RGB cameras, and various extreme environments. The RGB camera is fixed in the middle of the autonomous vehicle and returns an image every ten frames.

In the extreme environment setting, we selected four environmental factors: multiple vehicles, heavy fog, low light, and heavy rain for simulation. At the same time, we combined the four environmental factors to obtain a total of sixteen different environments. The same number of photos were collected for each environment, totaling 3200 images. Sixteen scenario settings are shown in Figure 8, where 1 represents non extreme conditions and 2 represents extreme conditions, providing convenience for subsequent regression analysis.



Light intensity	Rain intensity	Fog intensity	Number of vehicles
1	1	1	1
2	1	1	1
1	2	1	1
1	1	2	1
1	1	1	2
2	2	1	1
2	1	2	1
2	1	1	2
1	2	2	1
1	2	1	2
1	1	2	2
2	2	2	1
2	2	1	2
2	1	2	2
1	2	2	2
2	2	2	2

Fig. 8. Extreme environmental scenario settings

4.4 Simulation result

To confirm the efficacy of any improvement measure, the average precision of the IOU is set at 0.5 (mAP@0.5). Under the same circumstances, comparison tests were carried out between the enhanced algorithm suggested in this study and the basic YOLOv5 method. Table 1 displays the parameter settings used during model training, and the performance evaluation indicators of the two algorithms after training are shown in Table 2. This article improves the algorithm mAP@0.5 After 100 epochs, it increased from 0.678 to 0.9667, while the basic YOLOv5 algorithm mAP@0.5 From 0.525 to 0.9567, the enhanced algorithm in this article has increased the average accuracy by 1 percentage point compared to the basic YOLOv5. The comparison of simulation results is shown in Figure 9.

Table 1. Model training parameters

Parameters	Value
Batch_size	8
Momentum	0.937
Epochs	99
Lr0	0.01
Lrf	0.01
Weight_decay	0.0005
Img_size	640
Num_works	1

Table 2. Performance testing and evaluation indicators for improved algorithms and basic YOLOv5 algorithms

Algorithms	AP/%		Recall	mAP@0.5/%	FPS
	Car	Person			
Improved algorithms	97.25	96.10	0.98	96.67	44
YOLOv5 algorithms	95.98	95.37	0.97	95.67	31

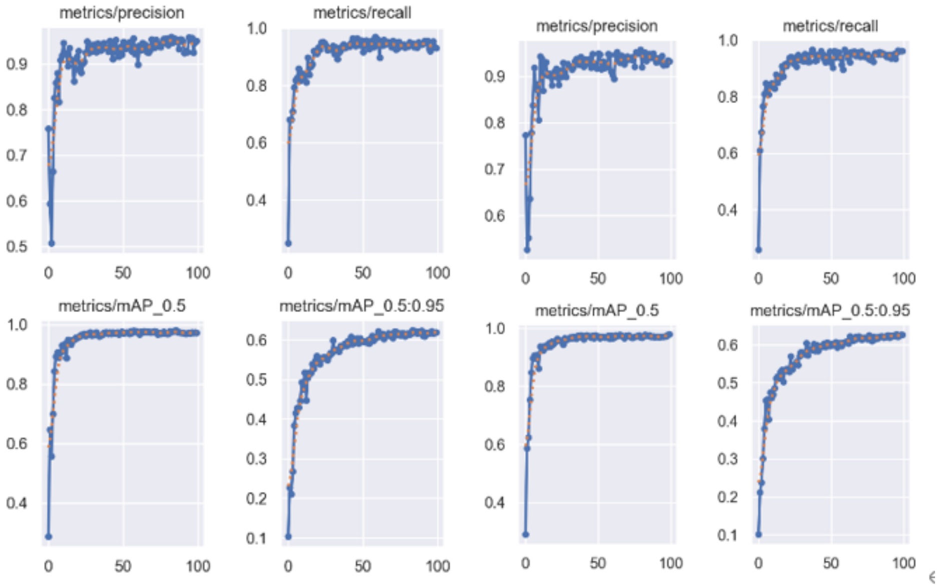


Fig. 9. Simulation result

5 Conclusion

This article proposes an improved YOLOv5s algorithm that effectively enhances the detection accuracy of vehicles in extreme environments. Adding a small target correction detection layer to form a four scale detection and correction network structure, correcting feature maps at various sizes to enrich the feature information of small targets and further improve their detection capabilities; Improving the backbone network's feature extraction capabilities by implementing SimAM; In the feature fusion module, GSConv network is used instead of standard convolution to reduce model parameters and enhance the network's feature fusion ability. Replace the CIoU loss function with SIoU in the calculation of the bounding box regression loss function. The simulation results on the CARLA simulation dataset show that, while maintaining a fast enough real-time detection speed, the algorithm's detection accuracy in this paper has improved by one percent, which can provide reference and guidance for improving the visual environment perception ability of autonomous vehicle.

References

[1] Li C, Guo Y. A Brief Analysis of the Development Status, Trends, and Challenges of Autonomous Driving Technology Era Automotive, 2022(14): 4–6.

[2] Li K, Chen Y, Liu J, et al. Overview of deep learning based object detection algorithms Computer Engineering, 2022, 48(7): 1–12. [doi: 10.19678/j.issn. 1000-3428.0062725]

[3] Hinton GE, Salakhutdinov RR. Reducing the dimensionality-of data with neural networks. Science, 2006, 313(5786): 504–507. [doi:10.1126/science.1127647]

[4] Arora N, Kumar Y, Karkra R, et al. Automatic vehicle detection system in different environment conditions using fast R-CNN[J]. Multimedia Tools and Applications, 2022, 81(13):18715-18735.

[5] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(06):1137-1149.

[6] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus: IEEE, 2014. 580–587. [doi: 10.1109/CVPR. 2014.81]

[7] Tian Y, Gelernter J, Wang X, et al. Lane marking detection via deep convolutional neural network. Neurocomputing, 2018, 280: 46–55. [doi: 10.1016/j.neucom.2017.09.098]

[8] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 779–788. [doi: 10. 1109/CVPR.2016.91]

[9] Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector. Proceedings of the 14th European Conference on Computer Vision. Amsterdam: Springer, 2016. 21–37. [doi: 10.1007/978-3-319-46448-0_2]

[10] Shao S, Zhang D, Chu H, et al. A review of YOLO object detection based on deep learning Journal of Electronics and Information Technology, 2022, 44(10): 3697–3708. [doi:10.11999/JEIT210790

[11] Wang P, Huang H, Wang M. Improve the complex road object detection algorithm of YOLOv5 Computer Engineering and Applications, 2022, 58(17): 81–92. [doi: 10.3778/j.issn.1002-8331. 2205-0158]

[12] Yang L, Zhang R Y, Li L, et al. Simam: A simple, parameter-free attention module for convolutional neural networks[C]//International conference on machine learning. PMLR, 2021: 11863-11874.

[13] Qin X, Li N, Weng C, et al. Simple attention module based speaker verification with iterative noisy label detection[C] / / IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). New York: IEEE, 2022: 6722-6726.

[14] Li H, Li J, Wei H, et al. Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles[J]. arXiv preprint arXiv:2206. 02424, 2022.

[15] Wang X, Song J. ICIoU: improved loss based on complete intersection over union for bounding box regression[J]. IEEE Access, 2021, 9: 105686-105695.

[16] Ahmed F, Tarlow D, Batra D. Optimizing expected intersection over-union with candidate-constrained CRFs[C] / / Proceedings of the IEEE International Conference on Computer Vision. New York: IEEE, 2015: 1850-1858.