Original Research Article
Research on Small Object Detection Algorithm Based on Improved YOLOv5

ABSTRACT

The YOLOv5 algorithm is widely used in object detection due to its efficient inference speed and high accuracy. However, it still faces challenges in small object detection. This paper proposes a series of improvements, including the addition of small object detection layers, the integration of the CBAM attention mechanism, and the optimization of the loss function by introducing EIoU, to enhance the model's feature extraction capability and detection accuracy. First, the paper enhances the network's perception of small objects by adding pyramid low-level semantic layers and constructing new small object detection heads. Second, the CBAM module is integrated into the C3 module, improving the model's feature representation ability and effectively preventing information loss. Finally, by introducing the EIoU loss function, the quality contribution of anchor boxes is enhanced, improving the model's detection accuracy and regression speed. Experimental results show that the improved YOLOv5 algorithm performs excellently on the BDD100K dataset, especially in small object detection. Compared with the original algorithm, it shows improvements in detection accuracy, recall rate, and mean average precision (mAP), despite the slight increase in parameters and computation, it still meets real-time requirements. This research provides strong support for further enhancing small object detection in autonomous driving scenarios.

Keywords: YOLOv5, small object detection, CBAM, EIoU

1. INTRODUCTION

In recent years, with the rapid development of computer technology and the continuous improvement of automobile manufacturing, autonomous driving has become a technological hotspot in the field of intelligent transportation. Vehicle object detection, as an essential component of the data acquisition process for autonomous driving[1], provides crucial support to ensure the safe operation of autonomous vehicles. Therefore, improving the accuracy of vehicle object detection algorithms is of paramount importance.

Currently, numerous vehicle object detection algorithms have been proposed, typically categorized into traditional object detection algorithms and deep learning-based object detection algorithms. Traditional object detection algorithms often involve sliding windows[2], image segmentation[3], feature classifiers[4], and template matching methods[5]. These methods lead to computational complexity that grows exponentially with the increase in image pixels, thereby raising the demand for computational power. Moreover, traditional algorithms rely on hand-crafted features, which lack robustness against variations in object diversity, resulting in low detection efficiency and accuracy that cannot meet practical needs. With the advent of deep learning, object detection technology has achieved revolutionary progress by leveraging the powerful fitting and feature extraction capabilities of deep convolutional neural networks (CNNS). Common deep learning methods include R-CNN, Fast R-CNN, and Faster

R-CNN[6], which use CNNs for feature extraction and introduce a region proposal network (RPN) to generate candidate boxes, thereby improving detection efficiency and accuracy.

In recent years, object detection methods represented by YOLO (You Only Look Once) have rapidly emerged. YOLO methods are single-stage object detection networks that[7], compared to previous two-stage detection networks, offer high inference speed and relatively high accuracy, making them widely applicable in object detection. YOLOv5s is the fifth-generation version of this algorithm series and has been applied in various scenarios such as defect detection and object recognition. It has achieved high accuracy in detecting large and medium objects, but still faces significant challenges in the field of small object detection. Small objects usually have smaller sizes and lower resolutions in images or videos, with fewer available features. Background noise often interferes, leading to lower detection accuracy. Thus, small object detection remains a hot topic in the field of visual research.

To enhance the small object detection performance of the YOLO algorithm, researchers like Yu Jun et al.[8] have designed new CFM and FSM modules to supplement contextual information and suppress multi-scale feature fusion conflicts, thereby improving the detection effect of small objects. Their improvements have increased the mAP@0.5 value by 6.9% compared to the original model, but the large parameter size makes it difficult to meet real-time requirements. Chen Fankai et al.[9] have introduced the up-sampling operator CARAFE to increase the receptive field for data feature fusion, thus improving the performance of the feature pyramid network and enhancing the detection accuracy for dense and small objects. Yang et al.[**Error! Reference source not found.**] proposed the Query Det algorithm for small object detection, which uses coarse localization of small objects and sparse-guided high-resolution features to calculate precise detection results, resulting in improved mAP for small object detection. Dong et al.[11] incorporated C3 Ghost and Ghost modules in the YOLOv5 neck to reduce floating-point operations during feature channel fusion, and introduced the CBAM[12] attention module in the backbone network to enhance the extraction of important information for vehicle detection tasks, thereby improving the detection accuracy of the algorithm. They also adopted the Complete Intersection Over Union Loss (CIOU Loss)[13] to improve the localization precision of the algorithm. Although the performance of small object detection has improved, issues such as low detection rate and poor real-time performance still persist.

This paper integrates the C3 module, which can obtain richer gradient information, with the CBAM (Convolutional Block Attention Module) that combines channel attention and spatial attention, to create a new feature extraction module called C3_CBAM. This integration enhances the feature extraction capability of the algorithm without significantly increasing the number of parameters. Additionally, by adding new small object detection layers and introducing EIoU (Extended Intersection over Union) as the bounding box regression loss function, the detection and localization capability for small-scale objects is effectively improved. The improved algorithm was tested on the BDD100K[14] dataset and compared with YOLOv3[15], YOLOv4[16], and YOLOv5s[17]. Results indicate that the improved algorithm shows varying degrees of enhancement in detection speed and accuracy, meeting the real-time requirements of autonomous vehicles.

2. IMPROVED YOLOV5S ALGORITHM

Although the YOLOv5s algorithm is widely used in object detection, it still faces significant issues with missed detections and false detections, especially for small objects and occluded targets. To further improve the detection accuracy of YOLOv5s, this paper proposes the following improvements to the YOLOv5s algorithm structure:

1) Adding a small object detection head in the network’s head to enhance the model's detection performance for small objects and reduce the rates of false positives and missed detections.
2) Integrating the CBAM attention mechanism into the C3 module to enhance the model's feature representation capability and overall performance.

3) To improve the detection network's ability to detect occluded targets, introducing EIoU (Efficient Intersection over Union) as the bounding box regression loss function. EIoU offers faster regression speed and better results, fully considering the center point distance, overlap ratio between the candidate box and the ground truth box, as well as length and width losses. The improved YOLOv5s is illustrated in Figure 1.

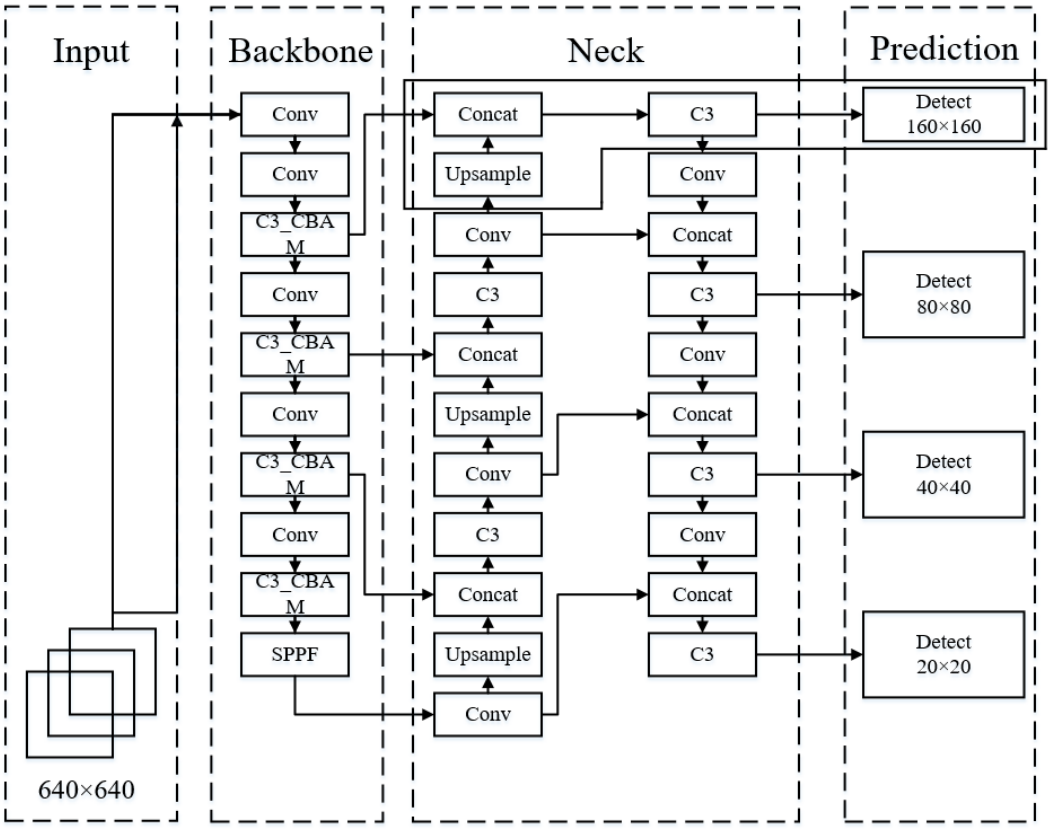

Figure 1: Improved Model Structure

2.1ADDING A SMALL OBJECT DETECTION LAYER

The YOLOv5 algorithm constructs three detection layers, performing down-sampling on feature maps at 32x, 16x, and 8x scales, and constructs three detection heads for object localization at these down-sampling points. Down-sampling increases the receptive field and ensures that information is not lost. YOLOv5 obtains prediction results from feature maps at three scales. After three rounds of down-sampling, the input image generates feature maps of sizes 20×20, 40×40, and 80×80, respectively. These three scales of feature maps correspond to detecting large, medium, and small-sized objects. However, in practical scenarios, objects that are further from the camera occupy too few pixels in the entire image, leading to missed and false detections for small objects in the image. To address these issues, a shallow feature map is constructed using multi-scale feature fusion on the basis of YOLOv5, adding an additional tiny object detection layer to improve the model's performance for small object detection.

2.2 INTEGRATING THE CBAM ATTENTION MECHANISM

The Convolutional Block Attention Module (CBAM) is a typical hybrid attention structure that integrates both channel attention and spatial attention. The CBAM module utilizes both Global Average Pooling (GAP) [18] and Global Max Pooling (GMP)[19], combining these two pooling strategies to prevent information loss. Applying the CBAM attention mechanism to the YOLOv5 model can enhance the model's feature representation capabilities and performance without significantly increasing the model's complexity, thereby retaining more useful information.The CBAM module consists of two different sub-components: the Channel Attention Module (CAM) [20], which operates on the channel dimension, and the Spatial Attention Module (SAM), which operates on the spatial dimension. These two sub-modules are combined in series to sequentially generate attention feature maps in the channel and spatial dimensions. The network structure is shown in Figure 2.

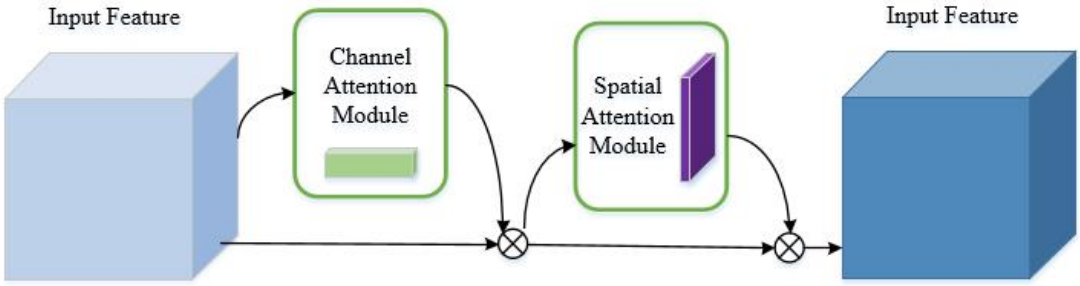

Figure 2: Network Structure Diagram of the CBAM Module

The CBAM process is as follows: First, the Channel Attention Module applies adaptive feature refinement to the input feature map F,resulting in the refined feature map F' .Next, the Spatial Attention Module further refines F' ， ultimately producing the feature map F'' processed by the CBAM module.

The Channel Attention Module in the CBAM mechanism is similar to the SE module. As shown in Figure 3, the input feature map is first processed using Global Average Pooling (GAP) and Global Max Pooling (GMP) operations, generating two 1×1×C feature maps. These two feature maps are then fed into a shared Multilayer Perceptron (MLP). The two channel attention vectors output by the MLP are summed, followed by the application of the Sigmoid activation function, ultimately producing the channel attention weights M_c. The structure of the Spatial Attention Module is shown in Figure 4. The feature map is first processed using GAP and GMP pooling operations along the channel dimension, resulting in two 1×H×W feature maps. These are concatenated to form a 1×H×W feature map. A 7×7 convolution operation is then performed, which both expands the receptive field of the feature map and reduces its dimension to 1×H×W. Finally, the feature map is fed into the Sigmoid activation function to obtain the spatial attention vector M_s. M_c and M_s can be calculated using equations (1) and (2), respectively.

$$M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \quad (1)$$

$$M_s(F) = \sigma(f^{7\times7}([AvgPool(F); MaxPool(F)])) \quad (2)$$

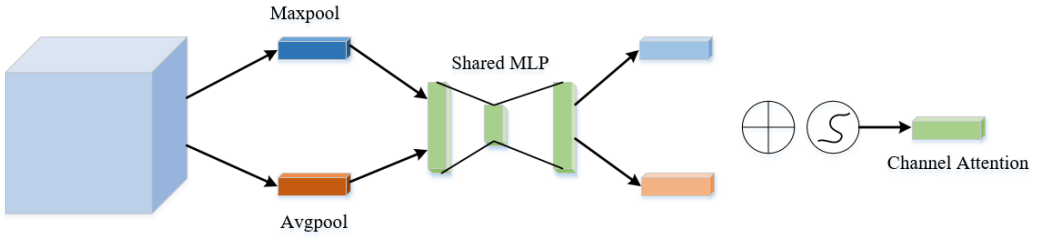

Figure 3: Structure Diagram of the Channel Attention Module

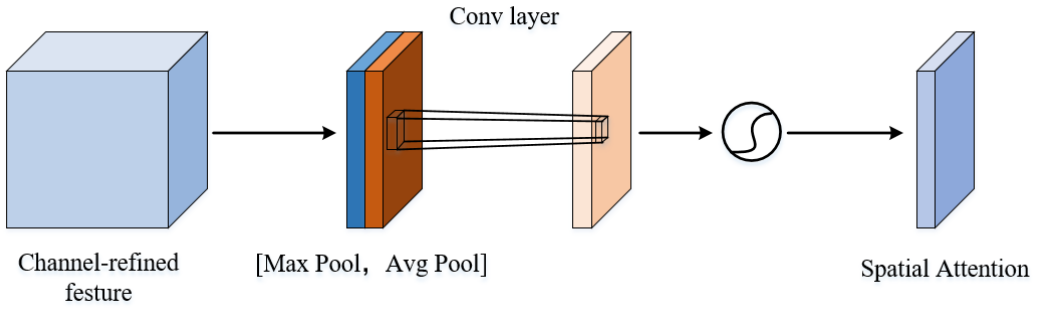

Figure 4: Structure Diagram of the Spatial Attention Module

Here,M_c represents the weights output by the Channel Attention Mechanism, and M_s represents the weights output by the Spatial Attention Mechanism. σ denotes the sigmoid function, MLP stands for Multilayer Perceptron, $MaxPool(F)$ and $AvgPool(F)$ are the outputs after Global Max Pooling and Global Average Pooling, respectively. $f^{7\times7}$ refers to a convolution operation with a kernel size of 7×7.

In the original YOLOv5s algorithm, the input image's feature map is first obtained using a convolutional module, followed by feature extraction using the C3 module. Within the C3 module, the Bottleneck structure processes the input feature map through multiple convolutional layers, enabling the extraction of more advanced and representative features. The C3 module itself is an efficient structure for feature fusion and enhancement. Integrating CBAM into the Bottleneck structure of the C3 module can significantly improve the model's feature extraction capability and detection performance without significantly increasing computational complexity. This improvement is particularly notable in small object detection tasks.

2.3 LOSS FUNCTION OPTIMIZATION

In deep learning, the loss function typically uses mean squared error to calculate the loss of the center coordinates and the bounding box dimensions. The location information of the predicted bounding box in object detection algorithms is independent and has no explicit connection to the ground truth bounding box coordinates. Mean squared error cannot describe the overlap relationship between the predicted box and the ground truth, and the calculated confidence score cannot indicate the quality of the prediction. Using the Intersection over Union (IoU) loss function for the predicted and ground truth boxes can better describe the loss. The IoU loss expression is shown in (3):

$$L_{IoU} = 1 - \left|\frac{S_A \cap S_B}{S_A \cup S_B}\right| \quad (3)$$

In the formula: $S_A \cap S_B$ represents the area of the intersection between the predicted box S_A and the ground truth box S_B; $S_A \cup S_B$ represents the area of the union between the predicted box S_A and the ground truth box S_B. The more the areas overlap, the closer the predicted box is to the ground truth.
The original YOLOv5s algorithm uses the CIoU Loss function, which takes into account the overlapping area, center point distance, and aspect ratio for bounding box regression. The calculation expression is shown in (4):

$$L_{CIoU} = L_{IoU} - \frac{\rho^2(O_A, O_B)}{c^2} - av \quad (4)$$

In the formula: O_A and O_B represent the center points of the predicted box S_A and the ground truth box S_B, respectively; O represents the Euclidean distance between the center points of S_A and S_B; c represents the diagonal distance of the bounding box C; a is the weight coefficient. The calculation expression is shown in (5):

$$a = \frac{v}{(1 - L_{IoU}) + v} \quad (5)$$

In the formula: v represents the aspect ratio similarity between the bounding boxes S_A and S_B. The calculation expression is shown in (6). The closer the aspect ratio of S_A is to that of S_B, the higher the accuracy of the predicted box.

$$v = \frac{4}{\pi^2}(arctan\frac{w^A}{h^A} - arctan\frac{w^B}{h^B})^2 \quad (6)$$

In the formula: w^A is the width of the ground truth; h^A is the height of the ground truth; w^B is the width of the predicted value; h^B is the height of the predicted value.
From the calculation formula of CIoU Loss, it is clear that the aspect ratio consistency parameter v reflects the difference in aspect ratios but does not adequately capture the actual differences in height and width relative to their confidence levels. This results in the CIoU Loss function not being able to effectively learn the similarity between the predicted box and the ground truth box. To address this issue, this paper introduces EIoU, which divides the aspect ratio regression into length loss and width loss based on CIoU. The EIoU-defined loss function consists of IoU loss, center point loss, length loss, and width loss. The calculation expression is shown in (7):

$$L_{EIoU} = 1 - L_{IoU} - \frac{\rho^2(A,B)}{c^2} + \frac{\rho^2(w^A, w^B)}{c_w^2} + \frac{\rho^2(h^A, h^B)}{c_h^2} \quad (7)$$

In the formula: $c_w$is the width of the bounding box C; is the height of the bounding box C.

EIoU considers four important geometric factors: overlapping area, center point distance, length, and width. This results in higher speed and accuracy for bounding box regression compared to the CIoU loss, leading to faster network convergence and better performance of the predicted bounding boxes.

3. EXPERIMENTAL RESULTS AND ANALYSIS.
3.1 DATASET AND EXPERIMENTAL ENVIRONMENT

The BDD100K dataset is an important dataset widely used for autonomous driving and computer vision research. It was released by the DeepDrive Lab at the University of California, Berkeley, and contains 100,000 images. There are 10 categories of ground truth bounding box labels: bike, bus, car, motor, person, rider, traffic light, traffic sign, train, and truck. The original dataset was preprocessed, with less frequent categories merged, and the resulting images were split into training, validation, and test sets in a 7:2:1 ratio.

The experimental training process was conducted on a Windows 10 operating system with CUDA 11.7 environment. The GPU configuration used was an RTX 3060 with 12GB of memory. The initial learning rate for training was set to 0.01, the momentum parameter was 0.9, the batch size was 32, the IoU threshold was 0.5, and the maximum number of training epochs was 200.

3.2 EVALUATION METRICS

In object detection, commonly used metrics to measure algorithm performance include Accuracy, Precision, Recall, Average Precision (AP), mean Average Precision (mAP), parameter count, Giga Floating-point Operations Per Second (GFLOPs), and Frames Per Second (FPS). Their specific meanings are as follows:

1) Precision: It represents the proportion of true positives among all instances predicted as positive. It is used to measure the probability of correct predictions in the results. The calculation method is shown in Equation 8.

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

2) Recall: It represents the proportion of true positives among the total actual positive samples. It is used to reflect the rate of missed detections. The calculation method is shown in Equation 9.

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

3) Mean Average Precision (mAP): Precision and Recall are two interrelated performance metrics. However, since each of these metrics has limitations when considered individually, they cannot fully evaluate the model's performance. Therefore, the mAP metric is introduced to balance the results of both. By plotting Precision on the y-axis and Recall on the x-axis, a Precision-Recall (P-R) curve can be obtained. The area under the P-R curve represents the Average Precision (AP) value. The mAP represents the average of the AP values across all classes in the dataset. The calculation method is shown in Equation 10.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{10}$$

3.3COMPARATIVE EXPERIMENTS

To validate the superiority of the improved YOLOv5s algorithm, several classic object detection algorithms were selected for comparative experiments. The experiments were conducted with identical parameters, with an input size of 640×640. The comparison results of different network models on the BDD100K dataset are shown in Table 1.

Table 1 Performance Comparison of Different Models on the BDD100K Dataset

Algorithm	P	R	mAP@0.5	Number of Parameters	GFLOPs	FPS
YOLOv3	59.1%	46.4%	53.2%	61.6	153.5	42
YOLOv4	63.4%	51.4%	57.9%	52.5	118.6	62
YOLOv5s	64.6%	52.1%	58.5%	7.0	15.6	110
Proposed Algorithm	67.1%	56.4%	60.7%	9.4	26.5	86

The improved YOLOv5s algorithm significantly outperforms the other three algorithms in detection accuracy while maintaining a relatively small number of model parameters. Compared to YOLOv3, YOLOv4, and YOLOv5s, this algorithm shows improvements in both Recall and mAP. Additionally, the number of parameters is significantly reduced compared to YOLOv3 and YOLOv4. However, compared to YOLOv5s, the number of parameters has increased. This is primarily due to the addition of a small object detection layer and the introduction of the CBAM attention mechanism in the C3 module in this algorithm. Although

these improvements result in an increase in the number of parameters and GFLOPs, which impacts detection speed to some extent, the algorithm still meets the real-time requirements of autonomous vehicles.

3.4 ABLATION EXPERIMENTS

To verify the effectiveness of the improvements made in each module of the proposed algorithm for enhancing small object detection accuracy, ablation experiments are first conducted. These experiments individually assess the impact of each module on the model, and the results are shown in Table 2.

Table 2 Ablation Experiments on the BDD100K Dataset

NO.	Small Object Detection Layer	C3_CBAM	EIoU	mAP@0.5	FPS
1				58.5%	110
2	✓			59.7(+2.1%)	96
3	✓	✓		60.4(+1.2%)	89
4	✓	✓	✓	60.7%(+0.5)	86

As shown in the data from Table 2, the improved model demonstrates a significant enhancement in mAP, particularly after adding the small object detection layer and integrating the CBAM attention module. Adding the small object detection layer helps retain more information from lower-level features, which is beneficial for accurately determining object locations. The integration of the CBAM attention mechanism in the C3 module allows for the extraction of more feature information, further improving the mAP. However, this improvement comes at the cost of increased computational load, which results in a reduction in detection speed.

4. CONCLUSION

This paper proposes an improved YOLOv5s algorithm that effectively enhances the detection accuracy of small vehicle targets. First, a new small object detection head is constructed by adding a pyramid of low-level semantic layers to improve the network's ability to perceive small objects. The original C3 module is augmented with the CBAM module to enhance the model's feature representation capability and performance, preventing information loss and retaining more useful information. The use of EIOU Loss highlights the contribution of high-quality anchor boxes, improving the localization accuracy of small-scale vehicle targets. Experimental comparisons show a significant improvement in detection accuracy with the proposed algorithm compared to the original one, making it suitable for precision requirements in autonomous driving scenarios. In future work, further research will focus on lightweight object detection models to enhance detection performance and optimize the dataset to improve the model's generalization ability.

REFERENCES

1. Zablocki É, Ben-Younes H, Pérez P, et al. Explainability of deep vision-based autonomous driving systems: Review and challenges[J]. International Journal of Computer Vision, 2022,130(10):2425-245.
2. VIOLA P, JONES M. Rapid object detection using a boosted cascade of simple features[C]//Procedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2001.
3. FELZENSZWALB P F, GIRSHICK R B, MCALLESTER D, et al. Object detection with discriminatively trained part-based models[J]. IEEE Trans Pattern Anal Mach Intell, 2010, 32(9): 1627-1645.

4. DALAL N, TRIGGS B. Histograms of oriented gradients for human detection[C]//IEEE Computer Society Conference on Computer Vision &. Pattern Recognition. IEEE, 2005.
5. LOWE D G. Distinctive image features from scale-invariant keypoints[J]. Int J Comput Vis, 2004, 60(2): 91-110.
6. REN S Q, HE K M, GIRSHICK R, et al. Faster R-CNN: Towards realtime object detection with region proposalnetworks[J]. IEEE Trans Pattern Anal Mach Intell, 2017, 39(6): 1137-1149.
7. REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, realtime object detection[C]//2016IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016.
8. Jun Yu and Yingshan Jia. "Improved YOLOv5 Algorithm for Small Object Detection." Computer Engineering and Applications 59, no. 12 (2023): 201-207.
9. Kai Fan and Shixin Li. "Improved YOLOv5 Algorithm for UAV Target Detection." Computer Engineering and Applications 59, no. 18 (2023): 218-225.
10. YANG C, HUANG Z H, WANG N Y. QueryDet: Cascaded sparse query for accelerating high-resolution small ob-ject detection[C]//2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2022.
11. Woo S, Park J, Lee J Y, et al.1. CBAM: Convolutional Block Attention Module[J]. Springer, Cham.2018.
12. Zhang Y F, Ren W, Zhang Z, et al. Focal andefficient IOU loss for accurate bounding boxregression[J].Neurocomputing,2022,506:146-157.
13. Yu F, Chen H, Wang X, et al. Bdd100k: A diverse driving dataset for heterogeneous multitask learning[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, USA, 2020: 2636-2645.
14. Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arxiv preprint arxiv:1804.02767 (2018).
15. Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." arxiv preprint arxiv:2004.10934 (2020).
16. Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.
17. Lin M, Chen Q, Yan S. Network in network[J]. arXiv preprint arXiv:1312.4400, 2013.
18. Zhou B, Khosla A, Lapedriza A, et al. Learning deep features for discriminative localization[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2921-2929.
19. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems 25 (2012).
20. Zheng Z, Wang P, Liu W, et al. Distance-IoU loss: Faster and better learning for bounding box regression[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(07): 12993-13000.