Mathematical Biosciences and Engineering

MBE, 21(4): 5782–5802.
DOI: 10.3934/mbe.2024255
Received: 02 January 2024
Revised: 21 February 2024
Accepted: 22 February 2024
Published: 26 April 2024

http://www.aimspress.com/journal/MBE

Research article

Research on a vehicle and pedestrian detection algorithm based on improved attention and feature fusion

Wenjie Liang*

Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China

* **Correspondence:** Email: liang15516956046@163.com.

Abstract: With the widespread integration of deep learning in intelligent transportation and various industrial sectors, target detection technology is gradually becoming one of the key research areas. Accurately detecting road vehicles and pedestrians is of great significance for the development of autonomous driving technology. Road object detection faces problems such as complex backgrounds, significant scale changes, and occlusion. To accurately identify traffic targets in complex environments, this paper proposes a road target detection algorithm based on the enhanced YOLOv5s. This algorithm introduces the weighted enhanced polarization self attention (WEPSA) self-attention mechanism, which uses spatial attention and channel attention to strengthen the important features extracted by the feature extraction network and suppress insignificant background information. In the neck network, we designed a weighted feature fusion network (CBiFPN) to enhance neck feature representation and enrich semantic information. This strategic feature fusion not only boosts the algorithm's adaptability to intricate scenes, but also contributes to its robust performance. Then, the bounding box regression loss function uses EIoU to accelerate model convergence and reduce losses. Finally, a large number of experiments have shown that the improved YOLOv5s algorithm achieves mAP@0.5 scores of 92.8% and 53.5% on the open-source datasets KITTI and Cityscapes. On the self-built dataset, the mAP@0.5 reaches 88.7%, which is 1.7%, 3.8%, and 3.3% higher than YOLOv5s, respectively, ensuring real-time performance while improving detection accuracy. In addition, compared to the latest YOLOv7 and YOLOv8, the improved YOLOv5 shows good overall performance on the open-source datasets.

Keywords: deep learning; road object detection; YOLOv5s; self-attention mechanism; feature fusion

1. Introduction

Traffic target detection in road scenes is an important research area in traffic monitoring and intelligent driving systems, as well as a key technology for achieving autonomous driving. Accurate and real-time traffic target detection algorithms are particularly important for environmental perception in road scenes. The current common object detection algorithms can be divided into traditional methods and deep learning-based methods.

Traditional object detection methods mainly consist of three parts: region selection, feature extraction, and classification. To locate the target position in the image, sliding windows with different scales and aspect ratios are set to traverse the possible positions of the target and obtain candidate regions. Then, manually designed features are used to extract features within candidate regions. Finally, a classifier is used to classify the extracted features. Guzman et al. [1] proposed an outdoor vehicle detection method based on the directed gradient histogram (HOG) and support vector machine (SVM). Guo et al. [2] proposed a classifier combining Adaboost and support vector machines for pedestrian detection in intelligent transportation systems. Razali et al. [3] proposed a visual analysis technique that combines hue saturation and value HSV color segmentation with support vector machines to detect emergency vehicles in images captured by traffic surveillance cameras. Zhu et al. [4] used Haar-like features to extract vehicle contours and texture features, and used Adaboost classifiers for classification and identification, improving the detection of vehicles in the longitudinal dimension ahead. However, the above methods often select manually designed feature representations. When facing complex and ever-changing targets, traditional shallow feature learning has relatively poor robustness and weak generalization ability.

With the development of deep learning, the accuracy of object detection algorithms based on computer vision technology and deep learning fusion is constantly improving, gradually becoming the mainstream method in this field. The object detection algorithm based on deep learning automatically learns the features of the target through convolutional neural networks, which can better adapt to different detection task requirements. This method can be divided into two categories: the first category is two-stage object detection algorithms, such as R-CNN [5], fast R-CNN [6], and faster R-CNN [7], etc. This type of method uses a region proposal network (RPN) to generate several candidate region proposals, and then detects the targets in the region proposals within the candidate regions to complete classification recognition. The generation of complex and redundant candidate regions is very time-consuming and cannot achieve real-time target detection. Another type are single-stage object detection algorithms, such as SSD [8], EfficientDet [9], RetinaNet [10], and the YOLO series, etc. This type of method directly obtains the position and category information of the target from the input image, transforming the detection problem into a regression problem, having a faster detection speed.

Before the emergence of the YOLO algorithm, two-stage object detection methods were the mainstream in the field of object detection, gaining widespread attention and application. Shi et al. [11] employed an object detection model based on incremental learning and fast R-CNN to detect vehicles. Yin et al. [12] proposed an improved domain-adaptive faster R-CNN model, adding three domain-adaptive components and enhancing the PRN network to improve the detection accuracy of small target vehicles on highways. However, the issue of model computational time persisted. Consequently, many researchers shifted their focus to one-stage object detection algorithms. The YOLO model, balancing the advantages of accuracy and speed, has found extensive applications in various domains. Zhao et al. [13] introduced an RDD-YOLO algorithm based on the improved YOLOv5 model for detecting defects on steel surfaces. Cai et al. [14] proposed a NAM-YOLOv7 model for detecting

SVCV-infected fish, enhancing the detection accuracy of abnormal fish by introducing the NAM-Attention mechanism and MPDIoU. Roy et al. [15] enhanced the disease detection performance in tomato plants by adding a DenseNet structure to YOLOv4, improving the path-aggregation network, and using the Hard-Swish activation function for enhanced nonlinear feature learning.

YOLOv5, known for its fast detection speed, high accuracy, and strong adaptability, is considered an advanced object detection model. Therefore, this study will use YOLOv5s as the baseline model for road object detection. In recent research, Kasper-Eulaers et al. [16] used the YOLOv5 model to detect heavy trucks in rest areas under winter conditions and predict occupancy rates, but faced challenges in handling occlusion, leading to instances of missed detections. Shi et al. [17] improved detection accuracy by adding new detection heads and integrating them with an attention mechanism. Zhang et al. [18] proposed a vehicle detection method based on YOLOv5, introducing the OSA aggregation module and utilizing non-local attention mechanisms and weighted non-maximum suppression to filter detection boxes, improving performance but increasing model complexity. Gao et al. [19] increased the model's attention to small targets by introducing a receptive field module, an attention mechanism, and adding a small target detection head in the YOLOv5 feature extraction network.

Although the above methods are of great significance for road target detection, there are still the following problems: (1) Small target objects are easily confused with the background, making it difficult to extract the feature information of the objects, making them difficult to detect. (2) Under complex background conditions, in scenes with dense traffic targets, varying degrees of occlusion between targets can result in the loss of target features, leading to missed and false detections of occluded targets. (3) Although the detection accuracy of the model has been improved, there are issues such as high computational complexity and slow detection speed.

In order to achieve accurate and efficient detection of road targets, our work contributes in the following ways:

(1) We propose an improved WEPSA module, which is integrated into the backbone network to adaptively balance channel attention and spatial attention, enhance useful target features, and enhance model perception ability.

(2) CBiFPN is used instead of PANet in the neck network, and cross-layer features are fused in a weighted form to enrich semantic information.

(3) Replace the original CIoU bounding box loss function of YOLOv5 with the EIoU [20] loss function to alleviate the occlusion problem between targets.

The rest of this paper is organized as follows. Section II introduces the YOLOv5s model and the new method. Section III presents an experimental study of the proposed method. Section Ⅳ discusses the experimental results. Section V concludes the work in this paper.

2. Methods

2.1. YOLOv5s network structure

Figure 1 shows the network structure of the YOLOv5s algorithm, which comprises four main parts: input, backbone, neck, and head. In the input module, operations such as Mosaic data augmentation and adaptive scaling are applied to preprocess the input images. The backbone consists of CBS modules, C3 modules, and spatial pyramid pooling fusion (SPPF) modules for feature extraction from the input images. The CBS module consists of a standard convolutional layer, a batch normalization layer, and a nonlinear activation function Silu, which is used for downsampling the input.

The C3 module comprises multiple bottleneck modules and three CBS modules, which extract, fuse, and enrich semantic information from the input. The SPPF module further enriches feature semantic information through pooling and feature fusion. The neck module employs a feature pyramid network (FPN) to transmit top-level features to the bottom, obtaining more global semantic information and enhancing the network's perception of large-scale targets. Additionally, it utilizes a path aggregation network (PAN) to transmit bottom-level features to the top, fully integrating information from different scale feature maps, enhancing the network's detection capability for small-scale targets. As the detection module, the head uses the Conv module to adjust the number of feature channels in the three feature layers, and predicts the final feature map, outputting target categories, and bounding box position information.

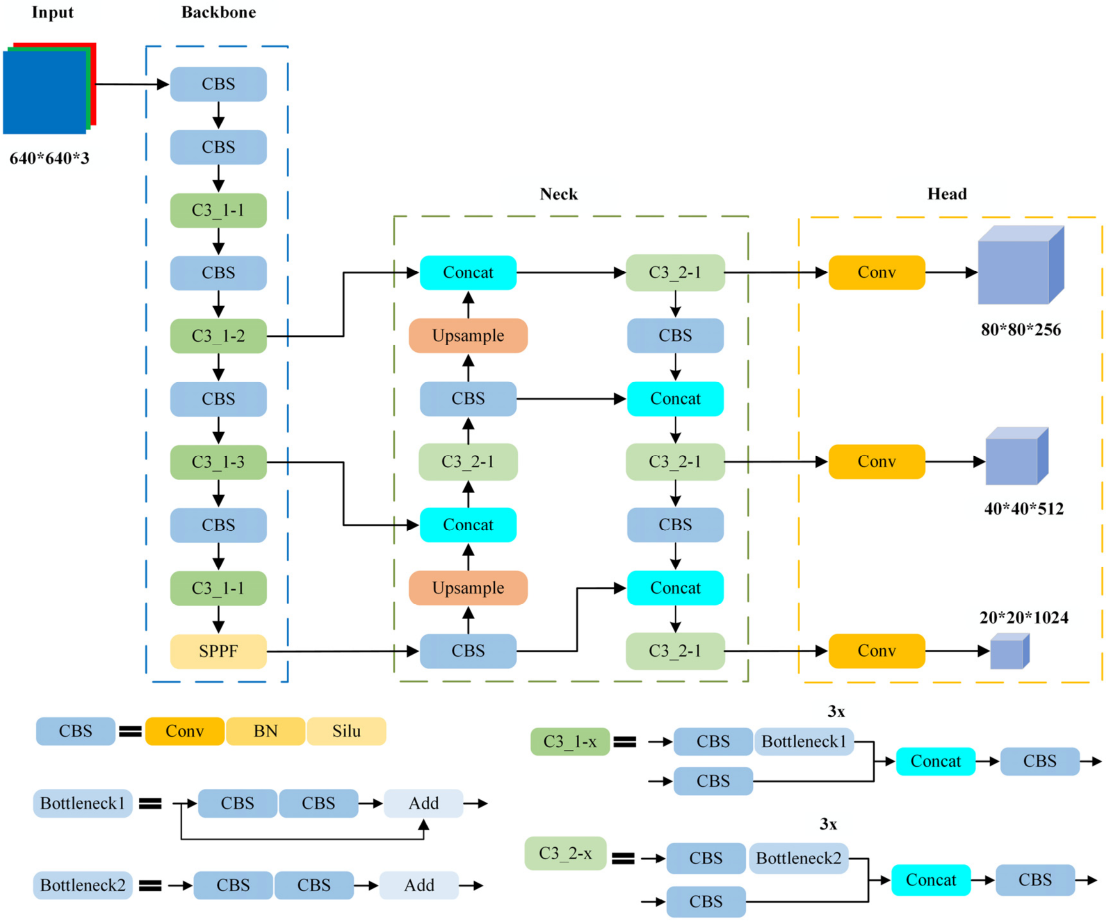


Figure 1. YOLOv5s network structure.

2.2. Improved YOLOv5s algorithm

The modified architecture of the YOLOv5s network is illustrated in Figure 2. Following the C3 module in the backbone network, the WEPSA mechanism is introduced, operating in both spatial and channel dimensions to enable the model to focus on regions of interest and suppress irrelevant information [21]. This contributes to the accurate capture of target features. To reinforce connections between feature information, the CBiFPN is proposed, assigning weights to each input feature for comprehensive integration and utilization of multi-scale target features. Finally, the EIoU loss function is introduced to expedite network convergence.

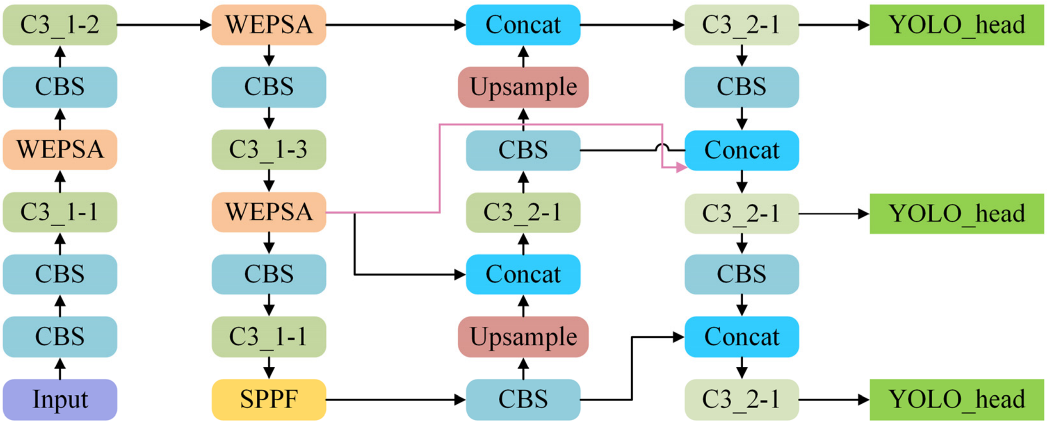


Figure 2. The network structure of improved YOLOv5s.

2.2.1. The improved polarized self-attention module (WEPSA)

The attention mechanism originates from the study of human vision, where the visual attention mechanism will help people quickly search for the region of interest in the whole image and obtain important information from it. In the field of target detection, the attention mechanism is used to acquire global and local features of an image, thereby enhancing overall information extraction. The polarized self-attention (PSA) mechanism [22] employs the polarized filtering idea of completely collapsing features in one direction when dealing with pixel-level regression tasks. However, it is not compressed to a great extent in both the spatial and channel dimensions. The benefit of this design decision is significantly reduced information loss while retaining more global image information. Additionally, its computational complexity is maintained at a relatively low level. Therefore, this paper integrates it into the improved model.

As shown in Figure 3, the PSA module can be divided into two parts: spatial self-attention and channel self-attention. Specifically, in the spatial self-attention module, the input features X are passed through two 1 × 1 convolutions to obtain W_v^2(C/2*H*W) and W_q^2(C/2*H*W), reducing the channel number by a half. The spatial dimensions are maintained at high-resolution H*W. For W_v^2, it is reshaped into (C/2*HW), and for W_q^2, global average pooling (GAP) is applied, causing full compression in spatial dimensions, inevitably leading to information loss. Subsequently, the Softmax function is employed to enhance the information of W_q^2. Next, the reshaped W_v^2 and the augmented W_q^2 are cross-multiplied to obtain W_z^2. After reshaping, it is fed into the Sigmoid function, keeping all the parameter values between 0 and 1. It is dot-multiplied with the input feature X to obtain the spatial attention output W_{sq}.

In the channel self-attention module, the same 1 × 1 convolutions are employed to obtain W_v^1 (C/2*H*W) and W_q^1 (1*H*W). W_q^1 undergoes complete channel compression, reducing the channel number by a half for W_v^1. After feature reshaping and applying the Softmax function, the reshaped W_v^1 (C/2*HW) and enhanced W_q^1 (HW*1) are cross-multiplied to yield W_z^1 (C/2*1*1).

Following a 1 × 1 convolution and layer normalization, the result is input to the Sigmoid function, ensuring that all parameters are constrained between 0 and 1. Finally, the obtained values are multiplied element-wise with the input features X to produce the output W_{ch}.

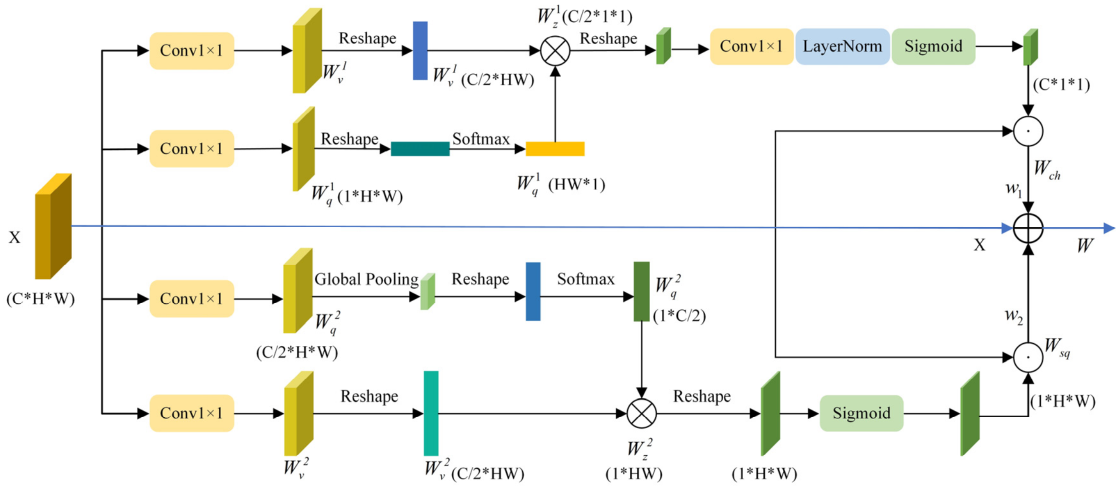


Figure 3. The improved polarization self-attention module.

From Figure 3, we observe that, in this PSA mechanism, the GAP operation is employed in the spatial self-attention part to obtain global information from each channel feature map. While significantly reducing the size of the feature map, using GAP may lead to information loss for small objects with fewer feature details. This is because merging all spatial information into a single point using GAP could result in the loss of information, making it challenging for the model to capture precise object locations, edges, and other detailed information. Additionally, the polarized filtering mechanism involves folding or compressing image features, which may lead to information loss in feature dimensions, thereby reducing the model's comprehensive understanding and expression of the original features. In response to the issues in the PSA module, we have made further improvements to better balance the relative importance of channel self-attention and spatial self-attention. We enhanced the module's adaptability and named it WEPSA. Specifically, we retained the original input features and introduced two weight parameters, w_1, and w_2, for the outputs of the two branches. This enables the model to automatically find an appropriate balance between the channel and spatial attention during training, thereby better-expressing image features. Denoting the original input as X, the original output as W, the output from the channel branch as W_{ch}, and the output from the spatial branch as W_{sq}, the original output and the improved output W_{out} can be calculated using the following formulas:

$$W = W_{ch} + W_{sq} \tag{1}$$

$$W_{out} = \mathrm{X} + W_{ch} \cdot w_1 + W_{sq} \cdot w_2 \tag{2}$$

2.2.2. Feature fusion module (CBiFPN)

In object detection tasks, to comprehensively capture target features of different scale sizes, FPN is introduced to aggregate feature information of different dimensions [23]. Common feature pyramid structures include FPN [24], PANet [25], and BiFPN [26]. FPN establishes a top-down pathway, progressively upsampling high-level information layer by layer and fusing it with shallow-level information. However, after multiple upsampling operations, high-level semantic information may be compromised. Considering the limitations of the unidirectional transmission of FPN, Liu et al. proposed the PANet structure, which adds a bottom-up pathway to FPN, conveying low-level semantic information to high-level layers, further enhancing the fusion capability of features. Based on PANet, the BiFPN structure removes nodes with only one input edge and introduces an additional skip connection line from the same-level input node directly to the same-level output node (indicated by the dashed line), thereby fusing more features without adding excessive computational complexity. Finally, different weights are assigned to each input undergoing feature fusion, enabling the network to gradually learn the importance of varying feature maps during subsequent training processes. As shown in Figure 4, the diagrams represent the network structures of FPN, PANet, and BiFPN, respectively.

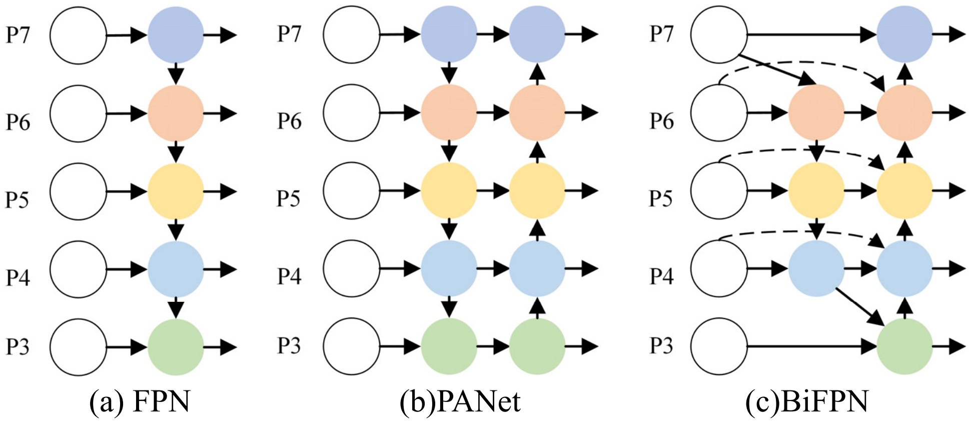


Figure 4. Three types of FPN structures.

In the feature fusion process of BiFPN, the Add operation is utilized to aggregate the feature maps, requiring the input feature maps to have consistent sizes and channel numbers. Adjustments to the contribution of each input are made through learnable weight parameters. While this approach can to some extent learn important information from different feature maps, it still has limitations, specifically information loss while maintaining consistent channels. To overcome this limitation, this paper proposes the CBiFPN module to improve the effectiveness of feature fusion. As shown in Figure 5, specifically, we assign different weights to the feature layers P3, P4, P5, and other feature inputs for fusion that are output from the backbone network, and then perform the Concat connection operation directly without adjusting the number of channels of the feature map when performing feature fusion at different scales. Here, C represents the Concatenation operation, and w represents the weight coefficient. Compared to the previous method, CBiFPN preserves the channel features of each feature map, avoiding potential total information loss caused by multiple adjustments to feature channels. In summary, the CBiFPN module, by maintaining the consistency of channel features and allowing the concatenation of features at different scales, more effectively capture multi-scale information in images. This improvement enhances the effectiveness of feature fusion and contributes to the performance improvements of the object detection models.

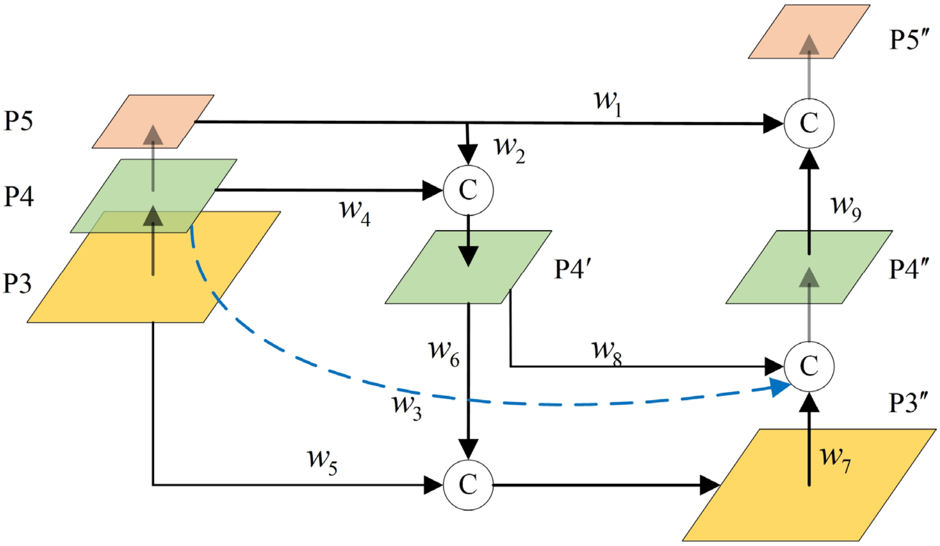


Figure 5. The improved feature fusion module.

For example, the calculation formula for P4′ and P4′ can be expressed as follows:

$$\mathrm{P4}' = Concat\big(Upsmple\big(Conv(\mathrm{P5})\big) * w_2, \mathrm{P4} * w_4\big) \tag{3}$$

$$\mathrm{P4}'' = Concat\big(Downsample\big(Conv(\mathrm{P3}'')\big) * w_7, Conv(\mathrm{P4}') * w_8, Conv(\mathrm{P4}) * w_3\big) \tag{4}$$

2.2.3. Optimization of the loss function

The loss function is commonly used to measure the difference between the predicted results and true annotated values, and it is utilized to optimize the model's training. The loss function of the YOLOv5s object detection algorithm consists of three parts: confidence loss, classification loss, and localization loss. Confidence loss and classification loss are computed using the binary cross-entropy loss (BCELoss), while the localization loss employs the CIoU loss. IoU represents the overlap between the predicted box and the actual box, where a higher IoU indicates higher prediction accuracy and vice versa. The formulas for IoU and CioU are as follows:

$$\mathrm{IoU} = \frac{\mathrm{A} \cap \mathrm{B}}{\mathrm{A} \cup \mathrm{B}} \tag{5}$$

$$\mathrm{CioU} = \mathrm{IoU} - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v \tag{6}$$

$$\alpha = \frac{v}{(1 - \mathrm{IoU}) + v} \tag{7}$$

$$v = \frac{4}{\pi^2}\left(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}\right)^2 \tag{8}$$

In the formulas, A and B represent the predicted and actual boxes, respectively. α is the weight function, v is a parameter measuring aspect ratio consistency, and w, w^{gt}, h, and h^{gt} are the width and height of the predicted box and the actual box.

Compared to IoU, CIoU considers factors such as the distance and scale similarity between the predicted box and the actual box. However, when faced with target boxes with the same center point and aspect ratio, the penalty term v becomes 0, leading to inaccuracies in describing the differences in the width and height of the target boxes and hindering further refinement of the algorithm. To address this issue, EIoU calculates the losses for width and height separately, effectively resolving the ambiguous definition of the aspect ratio in the CIoU loss function. This reduces errors in the horizontal and vertical directions, making the predicted box dimensions closer to the actual scale, thereby accelerating model convergence and precision regression. Therefore, this paper adopts EIoU as the regression loss function for target boxes. The formula is as follows:

$$\mathrm{EIoU} = \mathrm{IoU} - \frac{\rho^2(b,b^{gt})}{c^2} - \frac{\rho^2(w,w^{gt})}{c_w^2} - \frac{\rho^2(h,h^{gt})}{c_h^2} \tag{9}$$

In this formula, $\rho^2(b, b^{gt})$ is the Euclidean distance between the center points of the predicted box and the actual box, c is the diagonal distance of the minimum bounding rectangle enclosing A and B, c_w and c_h are the width and height of the minimum bounding rectangle, and $\rho^2(w, w^{gt})$ and $\rho^2(h, h^{gt})$ are the Euclidean distances between the differences in width and height between the predicted box and the actual box.

3. Experiment and analysis

3.1. Datasets

To verify the effectiveness of the improved model, this paper conducted experimental research using the KITTI and Cityscapes datasets and supplemented the validation on a self-built dataset.

1) KITTI dataset

The KITTI dataset, collected in Karlsruhe, Germany, comprises 7481 images. This study this data originates from focused on the detection of vehicles and pedestrians, merging categories such as “Van”, “Truck”, and “Tram” into “Car”. Additionally, the class “Person sitting” is consolidated into “Pedestrian”. The final dataset consists of three detection categories: “Car”, “Pedestrian”, and “Cyclist”. The dataset is divided into 6058 training images, 674 validation images, and 749 test images.

2) Cityscapes dataset

The Cityscapes dataset was obtained by the DAI Lab team using in-car cameras in over 50 German cities. This dataset contains rich real-world scenes such as urban streets, buildings, vehicles, and pedestrians. In the data preprocessing stage, we selected five common categories: car, bus, bicycle, rider, and person. The dataset is divided into 2780 training images, 347 validation images, and 348 testing images.

3) Self-built dataset

A total of 3455 images in JPG format with different scenes and densities are obtained by downloading the car driving recorder videos from the internet and intercepting them frame by frame. These images include four common targets: car, bus, person, and truck. Annotated with the help of the LabelImg tool in VOC format, 2790 images were obtained for the training set, 310 images for the validation set, and 345 images for the test set. Figure 6 shows some of the images in the dataset.

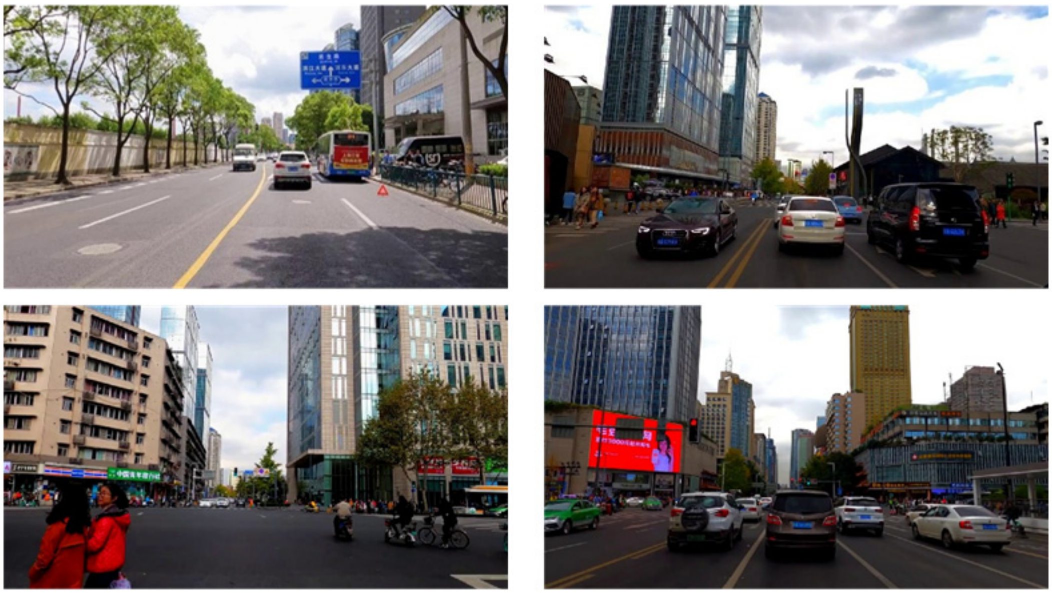


Figure 6. Partial self-built dataset samples.

3.2. Experimental environment and parameter configuration

The experimental setup for this study employed a system running Windows 10, equipped with an Intel(R) Core (TM) i7-6700K CPU @ 4.00 GHz and an Nvidia GeForce RTX 2060 GPU with 6 GB of VRAM. The deep learning framework used was PyTorch, accelerated with CUDA 10.2. The code was developed using Python 3.6 in the PyCharm Community Edition 2022.3 IDE. The training of the neural network model utilized the Adam optimizer with an initial learning rate of 0.001, a batch size of 16, and was executed for 300 epochs.

We employed common evaluation metrics in object detection as the criteria for our experiments, namely recall, precision, average precision (AP), and mean average precision (mAP). The formulas for these metrics are as follows:

$$Recall = \frac{TP}{TP+FN} \tag{10}$$

$$Precision = \frac{TP}{TP+FP} \tag{11}$$

$$AP = \int_0^1 PdR \tag{12}$$

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

In the formulas, TP represents the count of true positive samples predicted as positive, TN represents the count of true negative samples predicted as negative, FP represents the count of false positive samples indicated as positive, and FN represents the count of false negative samples predicted as negative. AP_i represents the area enclosed by the precision-recall (P-R) curve for class i. mAP represents the average value of class N samples AP. $mAP@0.5$ represents the average AP value for each class when IoU is 0.5. $mAP@0.5:0.95$ represents the average mAP value at different IoU thresholds.

3.3. Experimental results and analysis

3.3.1. Comparative experiment of improving attention mechanism

In order to further explore the effectiveness of the improved PSA attention module in enhancing the effect of the target detection model, this paper conducts ablation experiments on the KITTI dataset, aiming to evaluate the impact of different PSA configuration modules on the model performance. The results are shown in Table 1. In Table 1, PSA-1 is the model after embedding the original PSA module, PSA-2 is the model that adds initial features to the PSA output, PSA-3 is the model that sets different weights for the two output branches; and PSA-4 is the model after adding the initial features to the PSA output and assigning different weights to the two branches.

Table 1. The influence of the improved PSA attention module on algorithm performance.

Model	Precision (%)	Recall (%)	mAP@0.5 (%)	Parameters (M)
YOLOv5s	93.6	84.0	91.1	7.03
PSA-1	94.4	84.5	91.6(+0.5)	7.20
PSA-2	94.6	85.1	91.9(+0.8)	7.20
PSA-3	94.2	84.7	91.7(+0.6)	7.20
PSA-4	94.8	86.0	92.0(+0.9)	7.20

The experimental findings reveal that integrating the attention module into the backbone network helps the model concentrate more on crucial target information, thereby effectively enhancing the model’s detection accuracy. Specifically, upon incorporating the original PSA module, the model’s parameter count increased by 0.17 M, while the detection accuracy improved by 0.5 percentage points, suggesting a significant performance enhancement with the PSA module’s inclusion. Compared to the original YOLOv5s, PSA-2 achieved a 0.8% improvement in detection accuracy. This is because PSA-2 retains the original input features, avoiding complete compression of spatial or channel information in the self-attention module, thereby reducing information loss. PSA-3, by assigning different weights to channel branches and spatial branches based on the importance of varying channel or spatial position information, allows the model to selectively focus on specific information, resulting in a 0.6% improvement in detection accuracy. PSA-4 combines the advantages of PSA-2 and PSA-3, obtaining more diverse contextual information and further refining the importance of information. The model achieved a detection accuracy of 92.0%, showcasing the adaptive nature of this approach and effectively enhancing the model’s generalization capability.

In addition, to further validate the superiority of the WEPSA attention module proposed in this paper, comparative experiments were conducted with classic attention modules such as CA (channel attention), CBAM (convolutional block attention module), and SE (squeeze-and-excitation). The experimental results are presented in Table 2.

Table 2. The detection performance of different attention mechanisms.

Model	Precision (%)	Recall (%)	mAP@0.5 (%)	Parameters (M)
YOLOv5s-CA	94.4	84.7	91.6	7.05
YOLOv5s-CBAM	94.1	84.6	91.4	7.04
YOLOv5s-SE	94.7	84.5	91.5	7.08
YOLOv5s-WEPSA	94.8	86.0	92.0	7.20

The experiments showed that embedding CA, CBAM, and SE attention modules into the backbone network resulted in accuracy improvements of 0.5, 0.3, and 0.4 percentage points, respectively, for the object detection model. These different attention mechanisms focus on distinct aspects: CA and SE emphasize channel attention, neglecting spatial relationships between pixels; CBAM combines channel attention and spatial attention, primarily focusing on local feature relationships, whereas WEPSA emphasizes global features. Through comprehensive analysis, WEPSA, by preserving high-resolution in both image space and channel dimensions, significantly reduces information loss. Compared to other attention mechanisms, the model achieves higher accuracy and is better suited for road object detection tasks.

3.3.2. Comparative experiment of feature fusion module

Three sets of comparative experiments were conducted on the KITTI dataset in this study to validate the impact of the improved weighted feature fusion module on network performance in the neck section. The neck sections include the original FPN + PAN structure, the original BiFPN, and the proposed CBiFPN. The experimental results are presented in Table 3.

Table 3. The influence of different feature fusion modules on algorithm performance.

Model	Precision (%)	Recall (%)	mAP@0.5 (%)	Parameters (M)
YOLOv5s	93.6	84.0	91.1	7.03
YOLOv5s-BiFPN	94.5	83.3	91.4	5.23
YOLOv5s-CBiFPN	95.0	85.7	92.1	7.10

From the experimental results in Table 3, it can be observed that the feature fusion using BiFPN and CBiFPN structures demonstrates better performance compared to the PANet structure, with detection accuracy improvements of 0.3% and 1.0%, respectively. This reflects the significant impact of feature fusion methods on the performance of object detection models. BiFPN and CBiFPN introduce learnable weight parameters for dynamically weighting different-scale feature maps, enabling the model to learn the importance of features at different scales automatically. This adaptive feature fusion effectively handles multi-scale objects in images, contributing to improved model performance. Additionally, CBiFPN benefits from the Concatenation operation, which concatenates image information from different channels. Despite a potential increase in parameter count, this approach reduces information loss, thereby further enhancing the model's detection performance.

3.3.3. Comparative experiment of different loss functions

The original YOLOV5s used CIoU as the loss function between the target box and the actual box. To verify the effectiveness of the EIoU loss function while ensuring that other modules remain unchanged, this experiment analyzed the impact of different loss functions on model performance on the KITTI dataset by comparing them. The experimental results are shown in Table 4, and the loss decrease curves of different loss functions during the training process are shown in Figure 7.

Table 4. The influence of different loss functions on algorithm performance.

Loss	Precision (%)	Recall (%)	mAP@0.5 (%)	Parameters (M)
CIoU	93.6	84.0	91.1	7.03
EIoU	94.5	84.4	91.5	7.03

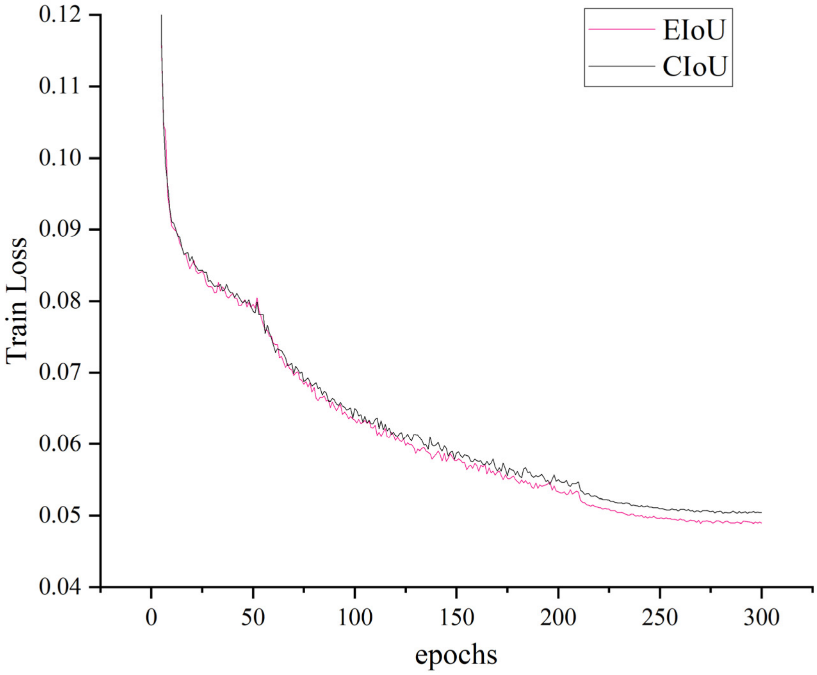


Figure7. The training loss value of different loss functions.

Overall, the original CIoU achieved a mAP@0.5 of 91.1%, while adopting the EIoU loss function resulted in an improvement of 91.5%. The model's average detection accuracy increased by 0.4%. Examining the training curve in Figure 6, it is evident that when using the EIoU loss function, the descent of the loss is faster, gradually plateauing after 250 epochs and ultimately converging. Compared to the original CIoU loss function, the EIoU loss function achieves a final convergence value of approximately 0.048, lower than the original CIoU loss function’s convergence value of about 0.050. Moreover, the EIoU loss function achieved higher detection accuracy, enhancing the model's detection performance effectively. Therefore, based on the experimental results, EIoU is selected as the loss function for training the object detection model in this study.

3.4. Ablation experiment

In order to better verify the influence of each improvement strategy on the detection performance of the algorithm, this paper conducts the following ablation experimental studies on the KITTI dataset, Cityscapes dataset, and the self-built dataset. The experimental results are shown in Tables 5–7.

Table 5. Ablation experimental results of the improved algorithm on the KITTI dataset.

Model	EIoU	CBiFPN	WEPSA	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Parameters (M)
1				91.1	55.4	7.03
2	√			91.5	56.1	7.03
3		√		92.1	57.4	7.10
4			√	92.0	56.5	7.20
5		√	√	92.5	58.5	7.27
6	√	√	√	92.8	60.2	7.27

Table 6. Ablation experimental results of the improved algorithm on the Cityscapes dataset.

Model	EIoU	CBiFPN	WEPSA	mAP@0.5 (%)	mAP@0.5:0.95 (%)
1				49.7	27.8
2	√			50.3	27.5
3		√		52.1	28.6
4			√	51.9	28.4
5		√	√	52.8	29.3
6	√	√	√	53.5	29.5

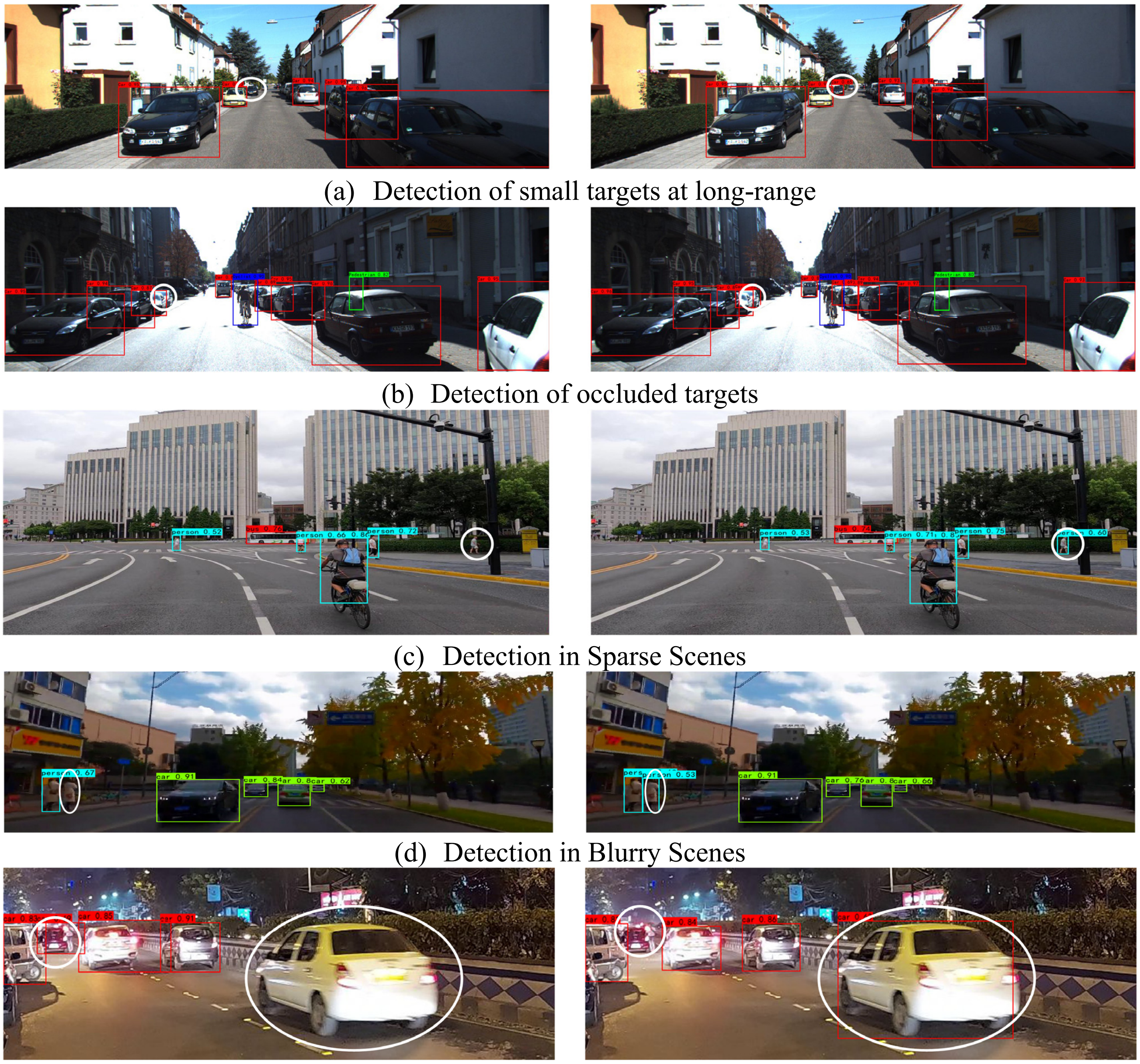
Table7. Ablation experimental results of the improved algorithm on the self-built dataset.

Model	EIoU	CBiFPN	WEPSA	mAP@0.5 (%)	mAP@0.5:0.95 (%)
1				85.4	57.4
2	√			86.4	57.6
3		√		86.7	58.2
4			√	87.4	59.2
5		√	√	87.9	58.9
6	√	√	√	88.7	59.8

Model 1 represents the original YOLOv5s. In model 2, the adoption the EIOU loss results in an increase in mAP@0.5 by 0.4%, 0.6%, and 1%, respectively. Model 3 uses the CBiFPN weighted feature fusion module to assign a weight to each input for feature fusion, allowing the model to gradually learn the importance of different scale features in the training process and effectively transmit information. mAP@0.5 is increased by 1%, 2.4%, and 1.3%, respectively. Model 4 introduces the WEPSA attention mechanism to help the network better capture the feature information of image objects and effectively improve the model detection performance; mAP@0.5 increased by 0.9%, 2.2%, and 2%, respectively. Model 5 uses the CBiFPN module based on Model 4; the number of parameters is increased by 3.4%, and the accuracy of the model is improved by 1.4%, 3.1%, and 2.5%, respectively. It is proved that the introduction of the WEPSA attention module in the backbone network and the weighted CBiFPN module in the neck network are effective for feature extraction and feature fusion. Model 6 integrates all improvement strategies, and mAP@0.5 is improved by 1.7%, 3.8%, and 3.3% to 92.8%, 53.5%, and 88.7%, respectively. In general, this paper significantly enhances the model performance without substantially increasing parameters, demonstrating the practicality of the improved modules.

3.5. Visualization results and analysis

In order to illustrate the performance differences between the improved model and the original model visually, this paper selected partial test set images from the KITTI dataset and the self built dataset for testing. The experimental results are shown in Figure 8, where the left images represent detections from the original model, and the right images depict detections from the improved model.



(a) Detection of small targets at long-range

(b) Detection of occluded targets

(c) Detection in Sparse Scenes

(d) Detection in Blurry Scenes

(e) Detection in Nighttime Scenes

Figure 8. Visualization results of target detection in different scenarios.

The detection of vehicle targets in the roadway, as depicted in Figure 8(a), reveals that vehicles are distributed on both sides of the lane. In the left picture of Figure 8(a), the distant black car is missed; however, it is accurately detected in the right picture. Moving on to Figure 8(b), under high-intensity illumination, cars on both sides of the road are heavily occluded; nevertheless, these occluded targets can be accurately detected in the right figure. Regarding Figure 8(c), which represents object detection in a sparse scene, the right image successfully detects a pedestrian leaning toward the right. In Figure 8(d), under blurry scenes caused by low resolution and pixel blurring, the edges and texture information of the target become unclear, posing a challenge to the network's feature learning. This difficulty in feature extraction leads to missed detections. The improved YOLOv5s algorithm addresses this issue by enhancing the network's focus on target features, improving the network's ability to extract features, and thereby mitigating the problem of missed detections. Figure 8(e) shows the detection performance of different models in nighttime environments. Due to the influence of light sources such as streetlights and headlights on nighttime images, the distribution of light may be uneven, resulting in certain areas being too bright or too dark, which affects the model's perception of the target. The rear of the white vehicle on the left side was illuminated by light and reflected, resulting in missed detection. However, the improved model successfully detected this situation. However, it can be seen that the improved model failed to detect black vehicles, resulting in missed detections. Based on comprehensive analysis and evaluation, we firmly believe that, when faced with target occlusion challenges, employing an attention mechanism aids in capturing partial target features by intensifying focus on areas likely to contain targets. Moreover, weighted feature fusion retains original features to facilitate accurate determination of target location and category even amidst occlusion scenarios while enhancing model robustness.

3.6. Comparison of different detection algorithms

To further verify the superiority of the improved detection algorithm in this paper, under the same experimental conditions, the experimental comparison and analysis are carried out on the KITTI dataset and Cityscapes dataset with common mainstream detection algorithms such as YOLOv3[27], YOLOv4-tiny, YOLOx [28], YOLOv7 [29], YOLOv7-tiny, YOLOv8n, SSD, and Faster R-CNN. The experimental results are shown in Tables 8 and 9.

Table 8. The detection performance of different target detection algorithms on the KITTI dataset.

Algorithm	AP (%)			mAP@0.5 (%)	FPS	Parameters (M)	FLOPs (G)
	Car	Pedestrian	Cyclist				
YOLOv5s	96.32	84.01	92.93	91.1	72	7.03	15.97
YOLOv4-tiny	78.43	33.98	42.92	51.8	145	5.88	6.84
YOLOv3	87.52	53.92	60.08	67.2	40	61.95	66.17
SSD	71.70	26.25	36.95	45.0	62	23.88	60.96
Faster R-CNN	88.90	68.82	77.62	78.5	10	137.1	370.21
YOLOx-s	96.67	86.52	95.83	93.0	73	8.97	26.97
YOLOv7	96.20	87.98	96.02	93.1	27	37.62	106.47
YOLOv7-tiny	92.26	73.12	86.43	83.9	91	6.23	13.86
YOLOv8n	96.63	82.91	85.93	88.5	114	3.16	8.9
Proposed method	96.45	85.45	96.47	92.8	63	7.27	17.13

Table 9. The detection performance of different target detection algorithms on the Cityscapes dataset.

Algorithm	AP (%)					mAP@0.5 (%)	FPS	Weight (M)
	Car	Bus	Bicycle	Rider	Person			
YOLOv5s	67.90	34.41	46.36	51.39	48.57	49.7	90	27.9
YOLOv4-tiny	61.25	25.80	32.70	35.80	44.83	40.1	135	22.4
YOLOv3	44.80	37.40	36.30	40.23	46.91	41.1	40	236
SSD	23.80	30.40	26.70	29.90	39.60	30.1	38	91.6
Faster R-CNN	46.90	38.90	40.30	39.80	45.12	42.2	15	110
YOLOx-s	73.27	42.00	46.15	51.73	51.20	52.9	62	34.3
YOLOv7	72.80	41.90	41.40	52.40	49.30	51.6	30	143
YOLOv7-tiny	68.93	36.84	47.50	51.23	49.81	50.9	92	23.1
YOLOv8n	69.77	35.97	44.08	48.38	52.18	50.1	104	11.6
Proposed Method	72.93	41.83	49.08	51.17	52.29	53.5	87	29

From the above ten sets of experimental comparisons, it can be observed that, compared to the lightweight object detection models YOLOv4-tiny, YOLOv7-tiny, and YOLOv8n, although the improved model in this paper has a slightly higher number of parameters and greater computational requirements, its overall detection performance is superior on the Cityscapes dataset; the proposed algorithm outperforms the state-of-the-art object detection algorithms YOLOv7 and YOLOv8n, with an increase of 1.9% and 3.4% in mAP@0.5, ranking first in precision. On the KITTI dataset, the detection accuracy of this paper reaches 92.8%, comparable to YOLOx and YOLOv7, but with a lower parameter count and network complexity, significantly improving training efficiency. The improved algorithm's mAP@0.5 increased by 41%, 25.6%, 47.8%, 14.3%, and 8.9% compared to YOLOv4-tiny, YOLOv3, SSD, Faster R-CNN, and YOLOv7-tiny, respectively. The detection accuracy for each category has been enhanced, with the detection accuracy for the Cyclist category reaching an optimal 96.47%. Overall, the average detection accuracy of the improved algorithm in this paper has been increased compared to the original algorithm. And, the detection speeds have maintained good real-time performance. The experiments validate the effectiveness of the model improvements proposed in this paper, meeting the detection requirements for road targets in complex environments.

3.7. Comparison of existing algorithm performance

Table 10 presents the results of different studies on the KITTI dataset. To ensure fairness, we adopt the same evaluation criteria for comparison. In this paper, our main focus lies on two crucial metrics: mAP@0.5 and FPS. Analyzing Table 9 reveals that, in terms of detection accuracy, our method performs exceptionally well on the KITTI dataset, trailing only behind Lightweight YOLOv3-promote by 0.5%. Concerning the mAP@0.5 metric, our method surpasses ORNet by 1.79%, outperforms YOLOx-s by 3.1%, and leads CenterNet-DHRNet by 5.7%. This indicates a significant competitive advantage of our algorithm in terms of accuracy for object detection tasks.

Regarding detection speed, our method ranks first in detection speed (FPS), significantly outperforming other methods. This implies that the algorithm proposed in this paper can achieve rapid object detection tasks while maintaining high efficiency. Such efficiency is particularly crucial for real-world applications that demand high real-time performance.

Table10. Comparison of detection by different studies on the KITTI dataset.

Models	mAP@0.5 (%)	FPS
CenterNet [30]	86.1	30
CenterNet-DHRNet [31]	87.1	18
Gaussian YOLOv3 [32]	86.79	24.91
Lightweight YOLOv3-promote [33]	93.3	25.5
YOLOx-s [34]	89.7	31.5
ORNet [35]	91.01	—
Proposed method	92.8	63

4. Discussion

This article explores a road object detection algorithm based on YOLOv5s for detecting vehicles and pedestrians on the road, which is of great significance for achieving intelligent monitoring and AI-assisted driving systems. To test the detection performance of the proposed model in complex backgrounds, we conducted a series of ablation experiments on the KITTI dataset, Cityscapes dataset, and on a self-built dataset. Tables 5–7 respectively show the experimental results of each improvement strategy on different experimental datasets. It can be seen that these improvements effectively improve the detection performance of the model while increasing a small number of parameters. The reason behind this is that, by improving the model's feature extraction ability for input feature images and strengthening the model's attention to key feature points, the performance of the model has been improved. And, the overall performance of the proposed method is superior to the original YOLOv5s. Tables 8 and 9 show the comparison of experimental results between the improved algorithm and current mainstream detection algorithms. Compared with advanced algorithms such as YOLOv7 and YOLOv8, the improved YOLOv5s algorithm achieved the best overall performance on multiple datasets and maintained good detection efficiency.

Furthermore, Figure 8 illustrates the comparison of detection performance before and after model improvements across different scenarios. It is evident that under well-lit sunny environments, the improvement of YOLOv5s achieved good results, but there were still missed detections in the night environment. Therefore, we analyze that the detection performance of the model is still limited in more complex situations such as night, rainy days, and heavy fog. How to maximize the robustness of the model under various conditions and enhance its generalization ability to better adapt to challenging scenarios in practical applications will be a key research focus in the future.

5. Conclusions

In summary, this paper proposes an object detection based on improved YOLOv5s for vehicle and pedestrian detection, providing solutions to the shortcomings of traditional YOLOv5s object detection. The main conclusions are as follows:

(1) Introducing the WEPSA module into the backbone network of YOLOv5s effectively enhances the model's feature extraction capability and suppresses interference from irrelevant target information in complex backgrounds.

(2) The adoption of the CBiFPN feature fusion module optimizes the fusion of multi-scale information in feature maps, significantly improving the algorithm's detection performance.

(3) Using EIoU as the bounding box regression loss function greatly accelerates model convergence and improves localization accuracy.

(4) Compared to current state-of-the-art algorithms, the proposed algorithm continues to demonstrate outstanding overall detection performance.

The proposed algorithm exhibits strong versatility, achieving average precision mAP values of 92.8%, 53.5%, and 88.7% on the KITTI dataset, the Cityscapes dataset, and on a self-built dataset, respectively. The improved model shows significant enhancement in detection accuracy on each dataset. In the future, the model will be deployed on mobile devices for real-time road object detection, and the proposed algorithm will be continuously refined in practical applications.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors received no specific funding for this work.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. S. Guzman, A. Gomez, G. Diez, D. S. Fernández, Car detection methodology in outdoor environment based on histogram of oriented gradient (HOG) and support vector machine (SVM), in *6th Latin-American Conference on Networked and Electronic Media (LACNEM 2015)*, (2015). https://doi.org/10.1049/ic.2015.0310
2. L. Guo, P. S. Ge, M. H. Zhang, L. H. Li, Y. B. Zhao, Pedestrian detection for intelligent transportation systems combining AdaBoost algorithm and support vector machine, *Exp. Syst. Appl.*, **39** (2012), 4274–4286. https://doi.org/10.1016/j.eswa.2011.09.106
3. H. Razalli, R. Ramli, M. H. Alkawaz, Emergency vehicle recognition and classification method using HSV color segmentation, in *2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA)*, (2020), 284–289. https://doi.org/10.1109/CSPA48992.2020.9068695
4. Z. M. Zhu, J. Qiao, Research of preceding vehicle identification based on HAAR-like features and Adaboost algorithm, *Electronic Measurement Technol.*, **40** (2017), 180–184. https:doi.org//10.19651/j.cnki.emt.2017.05.037
5. R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2014), 580–587. https://doi.org/10.1109/CVPR.2014.81
6. R. Girshick, Fast R-CNN, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, (2015), 1440–1448. https://doi.org/10.1109/ICCV.2015.169

7. S. Q. Ren, K. M. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *Adv. Neural Inf. Process. Syst.*, **28** (2015), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
8. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, et al., SSD: Single shot multibox detector, in *Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings*, **14**, (2016), 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
9. M. Tan, R. Pang, Q. V. Le, Efficientdet: Scalable and efficient object detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2020), 10781–10790. https://doi.org/ 10.1109/CVPR42600.2020.01079
10. T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in *Proceedings of the IEEE International Conference on Computer Vision*, (2017), 2980–2988. https://doi.org/10.1109/ICCV.2017.324
11. K. Shi, H. Bao, N. Na, Forward vehicle detection based on incremental learning and fast R-CNN, in *2017 13th International Conference on Computational Intelligence and Security (CIS)*, (2017), 73–76. https://doi.org/ 10.1109/CIS.2017.00024
12. G. Yin, M. Yu, M. Wang, Y. Hu, Y. Zhang, Research on highway vehicle detection based on faster R-CNN and domain adaptation, *Appl. Intell.*, **52** (2022), 3483–3498. https://doi.org/10.1007/s10489-021-02552-7
13. C. Zhao, X. Shu, X. Yan, X. Zuo, F. Zhu, RDD-YOLO: A modified YOLO for detection of steel surface defects, *Measurement*, **214** (2023), 112776. https://doi.org/10.1016/j.measurement.2023.112776
14. Y. Cai, Z. Yao, H. Jiang, W. Qin, J. Xiao, X. Huang, et al., Rapid detection of fish with SVC symptoms based on machine vision combined with a NAM-YOLO v7 hybird model, *Aquaculture*, **582** (2024), 740558. https://doi.org/10.1016/j.aquaculture.2024.740558
15. A. M. Roy, R. Bose, J. A. Bhaduri, A fast accurate fine-grain object detection model based on YOLOv4 deep neural network, *Neural Comput. Appl.*, **2022** (2022), 1–27. https://doi.org/10.1007/s00521-021-06651-x
16. M. Kasper-Eulaers, N. Hahn, S. Berger, T. Sebulonsen, Ø. Myrland, P. E. Kummervold, Detecting heavy goods vehicles in rest areas in winter conditions using YOLOv5, *Algorithms*, **14** (2021). https://doi.org/10.3390/a14040114
17. T. Shi, Y. Ding, W. Zhu, YOLOv5s_2E: Improved YOLOv5s for aerial small target detection, *IEEE Access*, **2023** (2023). https://doi.org/10.1109/ACCESS.2023.3300372
18. C. J. Zhang, X. B. Hu, H. C. Niu, Vehicle object detection based on improved YOLOv5 method, *J. Sichuan Univ.*, **5** (2022), 79–87. https://doi.org/10.19907/j.0490-6756.2022.053001
19. T. Gao, M. Wushouer, G. Tuerhong, DMS-YOLOv5: A decoupled multi-scale YOLOv5 method for small object detection, *Appl. Sci.*, **13** (2023), 6124. https://doi.org/10.3390/app13106124
20. Y. F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, T. Tan, Focal and efficient IOU loss for accurate bounding box regression, *Neurocomputing*, 506 (2022), 146–157.
21. B. Y. Sheng, J. Hou, J. X. Li, H. Dang, Road object detection method for complex road scenes, *Comput. Eng. Appl.*, **15** (2023), 87–96. https://doi.org/10.3778/j.issn.1002-8331.2212-0093
22. H. J. Liu, F. Q. Liu, X. Y. Fan, D. Huang, Polarized self-attention: Towards high-quality pixel-wise mapping, *Neurocomputing*, **506** (2022), 158–167. https://doi.org/10.1016/j.neucom.2022.07.054

23. J. H. Liu, G. F. Yin, D. J. Huang, Object detection in visible light and infrared images based on feature fusion, *Laser Infrared*, **3** (2023), 394–401. https://doi.org/10.3969/j.issn.1001-5078.2023.03.010
24. T. Y. Lin, P. Dollár, R. Grishick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2017), 2117–2125. https://doi.org/10.1109/CVPR.2017.106
25. S. Liu, L. Qin, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2018), 8759–8768. https://doi.org/10.1109/CVPR.2018.00913
26. M. X. Tan, R. Pang, Q.V. Le, Efficientdet: Scalable and efficient object detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2020), 10781–10790. https://doi.org/10.1109/CVPR42600.2020.01079
27. J. Redmon, A. Farhadi, Yolov3: An incremental improvement, preprint, arXiv:1804,02767. https://doi.org/10.48550/arXiv.1804.02767
28. Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, Yolox: Exceeding yolo series in 2021, preprint, arXiv:2107.08430. https://doi.org/10.48550/arXiv.2107.08430
29. C. Y, Wang, A. Bochkovskiy, H. Y. M. Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2023), 7464–7475. https://doi.org/10.1109/CVPR52729.2023.00721
30. X. Zhou, D. Wang, P. Krähenbühl, Object as points, preprint, arXiv:1904,07850. https://doi.org/10.48550/arXiv.1904.07850
31. X. Wang, Z. Li, H. L. Zhang, High-resolution network Anchor-free object detection method based on iterative aggregation, *J. Beijing Univ. Aeronaut. Astronaut.*, **47** (2021), 2533–2541. https://doi.org/10.13700/j.bh.1001-5965.2020.0484
32. J. Choi, D. Chun, H. Kim, H. J. Lee, Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (2019), 502–511.
33. H. Xu, M. Guo, N. Nedjah, et al., Vehicle and pedestrian detection algorithm based on lightweight YOLOv3-promote and semi-precision acceleration, *IEEE Trans. Intell. Transp. Syst.*, **23** (2022), 19760–19771. https://doi.org/10.1109/TITS.2021.3137253
34. S. G. Ma, N. B. Li, Z. Q. Hou, W. S. Yu, X. B. Yang, Object detection algorithm based on DSGIoU loss and dual branch coordinate attention, *J. Beijing Univ. Aeronaut. Astronaut.*, (2024), 1–14. https://doi.org/10.13700/j.bh.1001-5965.2023.0192
35. J. Chen, J. Zhu, R. Xu, Y. Chen, H. Zeng, J. Huang, ORNet: Orthogonal re-parameterized networks for fast pedestrian and vehicle detection, *IEEE Trans. Intell. Vehicles*, **2023** (2023), 2662–2674. https://doi.org/10.1109/TIV.2023.3323204