electronics

MDPI

Article

An Improved YOLOv5s Algorithm for Object Detection with an Attention Mechanism

Tingyao Jiang, Cheng Li *, Ming Yang and Zilong Wang

College of Computer and Information, China Three Gorges University, Yichang 443002, China
* Correspondence: lc@ctgu.edu.cn

Abstract: To improve the accuracy of the You Only Look Once v5s (YOLOv5s) algorithm for object detection, this paper proposes an improved YOLOv5s algorithm, CBAM-YOLOv5s, which introduces an attention mechanism. A convolutional block attention module (CBAM) is incorporated into the YOLOv5s backbone network to improve its feature extraction ability. Furthermore, the complete intersection-over-union (CIoU) loss is used as the object bounding-box regression loss function to accelerate the speed of the regression process. Experiments are carried out on the Pascal Visual Object Classes 2007 (VOC2007) dataset and the Microsoft Common Objects in Context (COCO2014) dataset, which are widely used for object detection evaluations. On the VOC2007 dataset, the experimental results show that compared with those of the original YOLOv5s algorithm, the precision, recall and mean average precision (mAP) of the CBAM-YOLOv5s algorithm are improved by 4.52%, 1.18% and 3.09%, respectively. On the COCO2014 dataset, compared with the original YOLOv5s algorithm, the precision, recall and mAP of the CBAM-YOLOv5s algorithm are increased by 2.21%, 0.88% and 1.39%, respectively.

Keywords: object detection; YOLOv5s; attention mechanism; deep learning

Citation: Jiang, T.; Li, C.; Yang, M.; Wang, Z. An Improved YOLOv5s Algorithm for Object Detection with an Attention Mechanism. *Electronics* **2022**, *11*, 2494. https://doi.org/10.3390/electronics11162494

Academic Editor: Stefanos Kollias

Received: 1 July 2022
Accepted: 9 August 2022
Published: 10 August 2022

1. Introduction

In recent years, due to the advent of the era of big data and the rapid development of computer graphics cards, the computing power of computers has also improved, which has accelerated the development of artificial intelligence in the computer field. There are more and more studies related to artificial intelligence, for example, the research in [1–4] has good application value, and the research of object detection has also developed accordingly.

Object detection has a wide range of applications in many areas of artificial intelligence, including robot navigation [5], autonomous driving [6], medical imaging [7] and human–object interaction [8]. Current object detection algorithms are mainly divided into single-stage detection algorithms and two-stage detection algorithms. Single-stage detection algorithms are represented by the You Only Look Once (YOLO) series [9–13], single-shot multibox detector (SSD) series [14–17], etc. Two-stage detection algorithms are represented by the region-based convolutional neural network (R-CNN) series [18–20]. A single-stage detection algorithm simultaneously classifies and locates the object of interest during object detection, while a two-stage detection algorithm performs these tasks separately. The characteristics of single-stage detection algorithms include that their detection speeds are very fast, but their accuracies are low. A two-stage detection algorithm is the opposite of a single-stage detection algorithm, with high accuracy but a slow detection speed. At present, most object detection tasks are real-time detection problems based on video, which require high detection speed, so a single-stage object detection algorithm is more suitable.

The latest single-stage object detection algorithm is the YOLOv5 algorithm. Compared with other single-stage object detection algorithms, the YOLOv5 algorithm has a faster detection speed and a smaller model. YOLOv5 is divided into four different algorithms: YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x. The network structures of these four

different algorithms are roughly the same, but their differences lie in the depths and widths of the networks. Among them, YOLOv5s has a faster detection speed and a smaller model than the other three algorithms, but the disadvantage is that its accuracy is low. In response to this problem, this paper proposes an improved YOLOv5s algorithm, CBAM-YOLOv5s, which introduces an attention mechanism.

In recent years, the attention mechanism has been widely used in various fields of deep learning [21–24], including image processing, speech recognition and natural language processing. There are many attention mechanism modules in the field of computer vision, among which the most classic ones are the squeeze-and-excitation network (SENet) [23] and the convolutional block attention module (CBAM) [24]. SENet is the champion of the ImageNet2017 image recognition competition, and CBAM is the champion of the 2018 classification competition. In this paper, these two classical modules are introduced respectively for comparative experiments.

At present, there are many improved object detection models based on the attention mechanism, which have good results, but most of the model parameters are large, and the detection speed is not fast enough. The improved method in this paper has good performance in both detection effect and detection speed.

2. CBAM-YOLOv5s

In this section, the YOLOv5s algorithm is first introduced, followed by detailed descriptions of the improvements made to the YOLOv5s network structure and the object bounding-box regression loss function used by the algorithm proposed in this paper.

2.1. YOLOv5s Algorithm

The YOLOv5s algorithm includes three parts: a feature extraction backbone network, a feature fusion neck network and a detection head. The network structure is shown in Figure 1. The detection process of the YOLOv5s algorithm is roughly divided into three steps. The first step is to extract features, adjust the scale of the input image to 640×640, and input the adjusted image into the backbone network. The BottleneckCSP-2 module, the BottleneckCSP-3 module, and the BottleneckCSP-4 module output three different scales of feature maps with sizes of 80×80, 40×40 and 20×20, respectively; these three feature maps contain different feature information. The second step is feature fusion. The three different scales of feature maps obtained through the backbone network are transmitted to the neck network, and the neck network performs a series of upsampling, convolution, channel concatenation and other operations to fully integrate the information provided by the feature maps. The third step is to output the detection heads. After the neck network fully integrates the features, three detection heads with sizes of 80×80, 40×40 and 20×20 are output. These three detection heads with different scales are used to detect small objects, medium objects and large objects.

Compared with YOLOv4, YOLOv5s adds a focus module to the backbone network. The main function of this module is to periodically extract pixels from high-resolution images and reconstruct them into low-resolution images to improve the receptive field of each pixel while retaining relatively complete original information. The design of the module is mainly used to reduce the number of calculations and speed up the algorithm. YOLOv4 only uses a cross-stage partial network (CSP) [25] structure in the backbone network, while YOLOv5s uses CSP structures in both the backbone network and the neck network. A CSP structure is used for local cross-layer network fusion, which reduces the number of calculations while simultaneously ensuring accuracy.

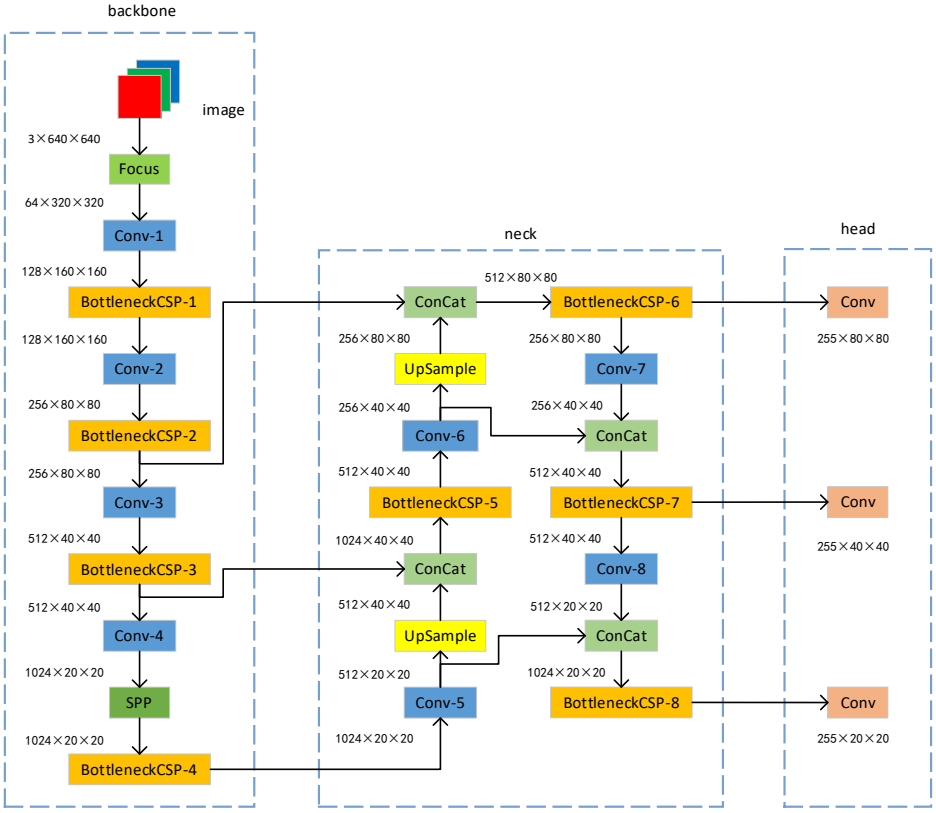

Figure 1. Structure of the YOLOv5s network.

2.2. Improved YOLOv5s with an Attention Mechanism

An attention mechanism is a data processing method in that is widely used in various types of machine learning tasks, such as natural language processing, image recognition and speech recognition. An attention mechanism is essentially similar to the mechanism by which humans observe external objects; when humans observe external objects, they are first inclined to observe some important local information about these objects and then combine the information derived from different regions to form an overall impression of the observed objects.

2.2.1. CBAM

The CBAM is a lightweight module that includes a channel attention submodule and a spatial attention submodule. The channel attention submodule focuses on important feature information, and the spatial attention submodule focuses on object location information. The structure of the CBAM is shown in Figure 2.

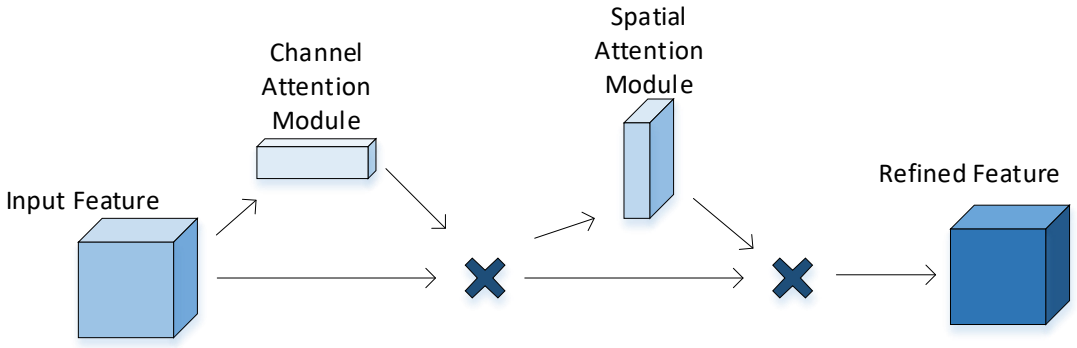

Figure 2. Structure of the CBAM.

The operation process of the channel attention submodule: The input feature map uses a global average pooling operation and a global maximum pooling operation to aggregate the spatial information of the input feature map to obtain a one-dimensional channel attention vector, sends it to a shared network, passes the added elements through a sigmoid activation function to obtain the resulting channel attention vector and finally multiplies the channel attention vector with the initial input to obtain the output of the channel attention submodule.

The operation process of the spatial attention submodule: The output of the channel attention submodule is subjected to an average pooling operation and a maximum pooling operation to obtain a spatial attention tensor; this is followed by channel concatenation. Then, the spatial attention tensor is obtained through a convolution operation and the sigmoid activation function; finally, the spatial attention tensor is multiplied with the output of the channel attention submodule to obtain the output of the spatial attention submodule.

2.2.2. YOLOv5s Introduces the CBAM

The CBAM is incorporated into the backbone network of YOLOv5s, and the network structure is shown in Figure 3. The function of the module is to let the network know which part to focus on and to accordingly achieve prominent representations of important features while suppressing the less important features; this module can adjust the attention weight of the feature map and improve the feature extraction ability of the network.

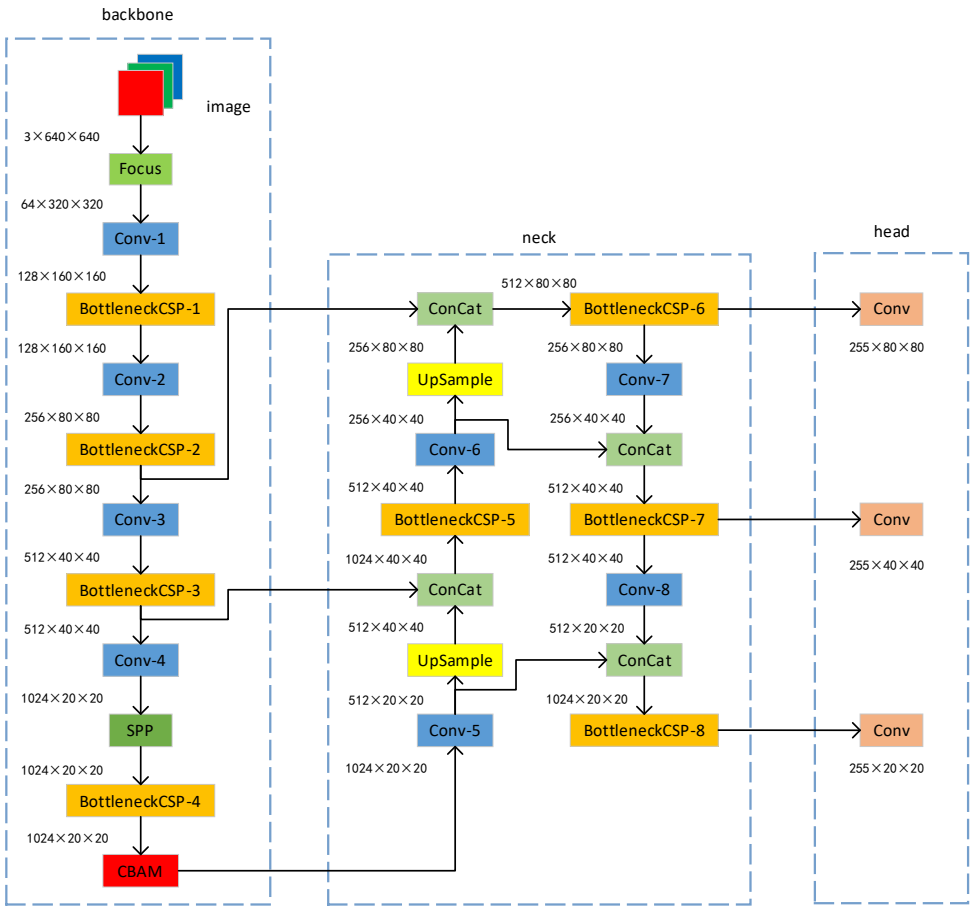

Figure 3. Structure of the CBAM-YOLOv5s network.

The specific operation of the CBAM is mainly divided into two steps, as shown in Figure 4.

In the first step, the channel attention operation is performed on the input feature map. The input 1024 × 20 × 20 feature map is processed through a maximum pooling operation and an average pooling operation to obtain two 1024 × 1 × 1 feature maps, and then these two feature maps are each compressed by the first fully connected layer to compress the number of channels to 64, thereby reducing the computational cost. This is followed by an expansion operation performed through the second fully connected layer to output two 1024 × 1 × 1 feature maps. Then, the feature information of the two feature maps is added and passed through the sigmoid activation function to obtain a 1024 × 1 × 1 feature map, and finally, the feature map is multiplied by the initial input to obtain an output of size 1024 × 20 × 20 with constant dimensions.

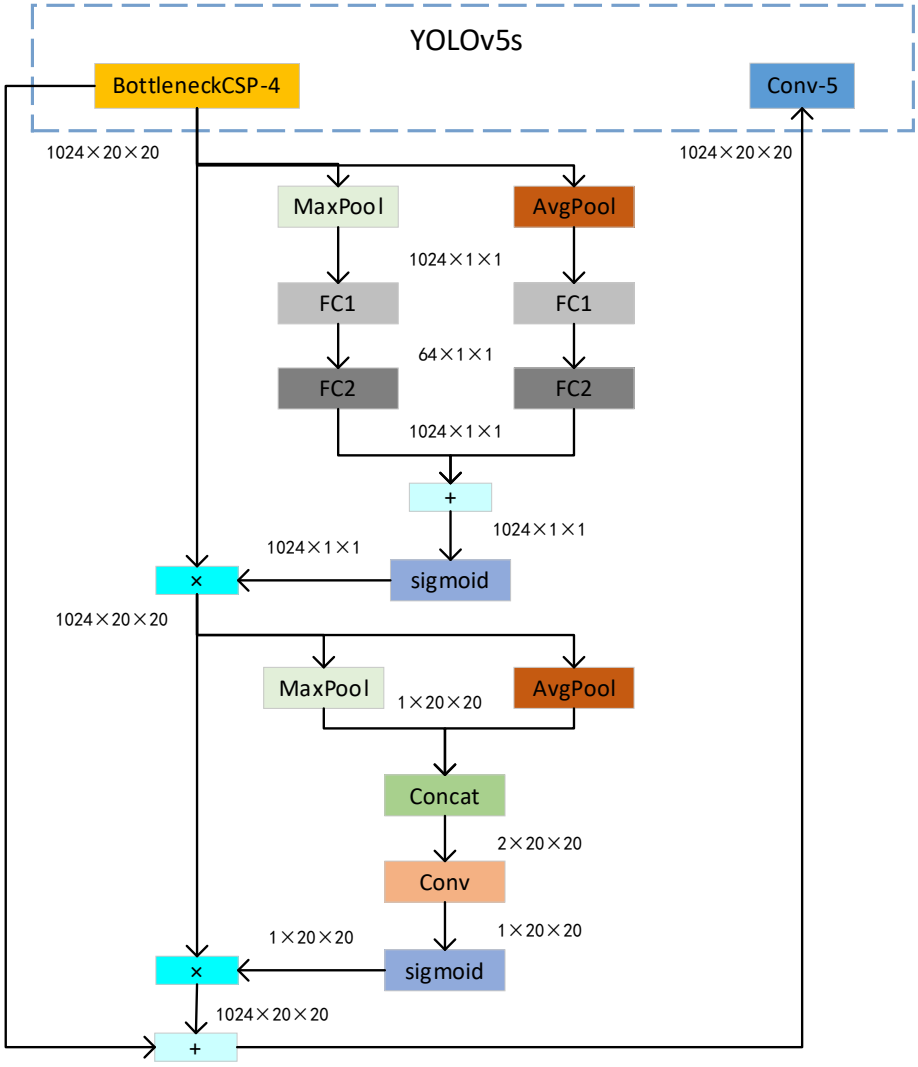

Figure 4. CBAM operation.

In the second step, the spatial attention operation is performed. The 1024 × 20 × 20 feature map obtained through the channel attention operation is subjected to a maximum pooling operation and an average pooling operation to output two 1 × 20 × 20 feature maps, and then 2 × 20 × 20 feature maps are output through the channel concatenation operation. Next, the dimensions of the feature map are restored to 1 × 20 × 20 through a convolution operation, and this is followed by the sigmoid activation function, which outputs a 1 × 20 × 20 feature map. Then, the feature map is multiplied by the initial input to obtain a 1024 × 20 × 20 feature map, and this feature map is added to the input of the BottleneckCSP-4 module to obtain the final output: a 1024 × 20 × 20 feature map. Finally,

the extracted 1024 × 20 × 20 feature map is input back into the Conv-5 module in the neck network.

2.3. The Object Bounding-Box Regression Loss Function

The object bounding-box regression loss functions of most object detection algorithms use generalized intersection-over-union (*GIoU*) loss [26] to calculate the deviation between each prediction box and the corresponding ground truth; this loss is defined as

$$L_{GIoU} = 1 - IoU + \frac{|Ac - U|}{|Ac|} \tag{1}$$

where Ac represents the area of the smallest box that contains both the ground truth and the prediction box, IoU represents the intersection over union of two bounding boxes, U represents the union of the two bounding boxes and L_{GIoU} represents the $GIoU$ loss.

The advantage of the $GIoU$ loss is that it not only focuses on the overlapping area between the prediction box and the ground truth but also focuses on other nonoverlapping areas, so it can better reflect the degree of overlap between the prediction box and the ground truth. However, the disadvantage of the $GIoU$ loss is that when the ground truth or the prediction box surrounds the other, the $GIoU$ loss function deteriorates, causing slow convergence and a large localization bias during the training process. The complete IoU ($CIoU$) loss [27] was developed in view of this problem; in addition to considering the overlapping area between the prediction box and the corresponding ground truth, the distance between the center points and the aspect ratio of the two bounding boxes are also considered. The $CIoU$ loss is given as

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v, \tag{2}$$

$$\alpha = \frac{v}{(1 - IoU) + v}, \tag{3}$$

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

where $\rho^2(b, b^{gt})$ denotes the Euclidean distance between the prediction box and the ground truth, c denotes the shortest diagonal length of the smallest box containing both the ground truth and the prediction box, α is the weight parameter, v denotes the similarity between the aspect ratios of the two bounding boxes, w^{gt} and h^{gt} denote the width and height of the ground truth, w and h denote the width and height of the prediction box, respectively, and L_{CIoU} denotes the $CIoU$ loss.

Compared with the $GIoU$ loss, the $CIoU$ loss adds loss terms for the center distance and the aspect ratio between the prediction box and the ground truth to the loss function, which makes the prediction box converge faster and the regression localization more accurate, so the algorithm in this paper uses the $CIoU$ loss as the object bounding-box regression loss function.

3. Experiments

To evaluate the improvement achieved by the CBAM introduced to YOLOv5s, the CBAM incorporated into the backbone network of YOLOv5s is replaced by another attention mechanism module called the SENet for an ablation experiment. In this section, the experimental equipment, dataset, evaluation metrics, experimental results and comparative analysis are introduced. We have put the core code of the algorithm on GitHub. Interested readers can download it, and the access link is https://github.com/2530525322/object-model (accessed on 9 August 2022).

The SENet mainly includes squeeze and excitation operations. The module structure is shown in Figure 5.

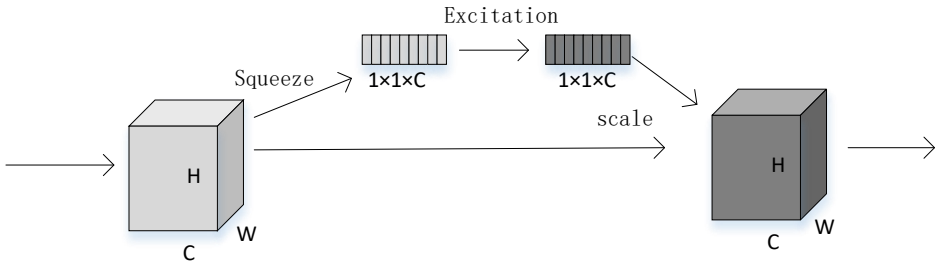

Figure 5. Structure of the SENet module.

The SENet mainly focuses the network's attention on specific channels by learning the connections between channels, thereby achieving improved accuracy. The general processing flow of the SENet is roughly divided into three steps.

Squeeze operation: The input W × H × C feature map is subjected to the global average pooling operation to obtain a 1 × 1 × C feature map.

Excitation operation: The result of the squeeze operation is transformed nonlinearly by using a fully connected layer.

Scale operation: The output obtained by the excitation operation is used as the weight and multiplied by the initial W × H × C input for the channel weights to obtain the final output.

3.1. Experimental Equipment and Training Parameters

The equipment used in the experiment is a Dell desktop computer, and its specific configuration is shown in Table 1.

Table 1. Computation system.

Name	Configuration
Processor	Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz
Running Memory	64 GB
Operating System	Linux
GPU	NVIDIA GeForce RTX 3080
GPU Memory	10 GB
Programming Tool	PyCharm
Programming Language	Python
Deep Learning Framework	PyTorch

Some of the training parameters in the experiment are shown in Table 2.

Table 2. Training parameters.

Parameter	Value
Learning Rate	0.01
Batch Size	32
Weight Decay	0.0005
Momentum	0.937
Epochs	300

3.2. Dataset

The datasets used in this experiment are the Pascal Visual Object Classes 2007 (VOC2007) dataset [28] and the Microsoft Common Objects in Context (COCO2014) dataset. The

COCO2014 dataset has a total of 123,287 images with 80 categories. The VOC2007 dataset contains a total of 9963 images. Twenty classes are included in the dataset, as shown in Figure 6; these classes include the airplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train and TV monitor categories, and the associated XML file provides the object class of the input image and the coordinates of the corresponding ground truth.

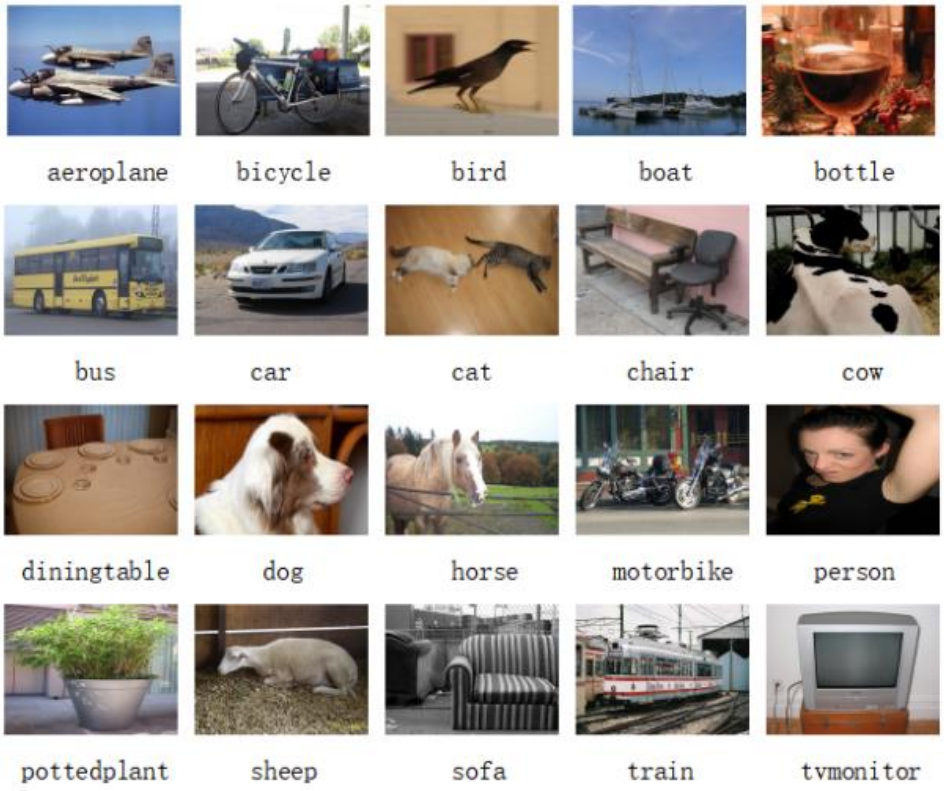

Figure 6. VOC2007 dataset.

3.3. Evaluation Metrics

To evaluate the performance of the proposed algorithm, the evaluation metrics in this paper are precision (P), recall (R), mean average precision (mAP), F-score and frames per second (FPS).

Precision calculates the proportion of the number of correctly predicted positive samples to the total number of samples predicted as positive samples, that is, the accuracy of the prediction for the evaluation object. Precision is defined as follows:

$$P = \frac{TP}{TP + FP}, \tag{5}$$

where TP represents true positives, that is, the number of positive samples predicted as positive samples; FP represents false positives, that is, the number of negative samples predicted as positive samples.

Recall calculates the proportion of the number of correctly predicted positive samples to the total number of actual positive samples, that is, whether the evaluation object is completely found or not. Recall is defined as follows:

$$R = \frac{TP}{TP + FN}, \tag{6}$$

where FN represents false negatives, that is, the number of positive samples predicted as negative samples.

The *mAP* calculates the mean of the average precision (*AP*) values of all classes and is used to evaluate the overall performance of the algorithm. The *mAP* is given as

$$AP = \int_0^1 P \, dR, \tag{7}$$

$$mAP = \frac{\sum_{j=1}^{c} AP_j}{c}. \tag{8}$$

F-score calculates the harmonic value of precision and recall, which can comprehensively measure these two indicators. The *F-score* is defined as follows:

$$F\text{-}score = \left(1 + \beta^2\right) \frac{P \cdot R}{\beta^2 \cdot P + R}, \tag{9}$$

where β is used to balance the weight of precision and recall in the *F-score*, and there are three values. When β is equal to 1, precision is as important as recall; when β is less than 1, precision is more important than recall; when β is greater than 1, recall is more important than precision.

3.4. Experimental Results and Comparative Analysis

During the experimental training process, the stochastic gradient descent (SGD) [29] optimization algorithm is used to update the model parameters. Table 3 shows the experimental results obtained on the VOC2007 dataset.

Table 3. Ablation experiment.

Dataset	Attention Mechanism		Precision	Recall	mAP@0.5	mAP@0.95	F1-Score	FPS
	SENet	CBAM						
VOC2007	×	×	75.68%	60.87%	66.35%	41.14%	67.47%	76
	✓	×	76.27%	62.09%	67.33%	42.03%	68.45%	57
	×	✓	80.20%	62.05%	69.44%	45.99%	69.97%	60

As seen in Table 3, compared with those of the original YOLOv5s algorithm that does not introduce an attention mechanism, the precision, recall and mAP of the proposed algorithm that introduces an attention mechanism are improved. Compared with the original YOLOv5s, the YOLOv5s version with the SENet module achieves a 0.59% improvement in precision, a 1.22% improvement in recall and a 0.98% improvement in mAP, while the YOLOv5s version with the CBAM yields larger improvements, with a 4.52% improvement in precision, a 1.18% improvement in recall and a 3.09% improvement in mAP. By conducting a comparative analysis on the experimental results, it can be concluded that the algorithm in this paper has better performance than the original algorithm and the algorithm with the SENet module. SENet only includes channel attention and can only obtain important feature information on the channel, while CBAM includes not only channel attention but also spatial attention. It can obtain important feature information in both channel and space, so that the network can better learn important features in the image. The more picture features the network learns, the better it can recognize the object, which will make the network's recognition accuracy higher.

The experimental comparison results of the object bounding-box regression loss function are shown in Figure 7, where the horizontal axis is the number of epochs and the vertical axis is the value of the bounding-box loss. The experimental results show that the use of the CIoU loss as the bounding-box regression loss function results in faster convergence than the GIoU loss.

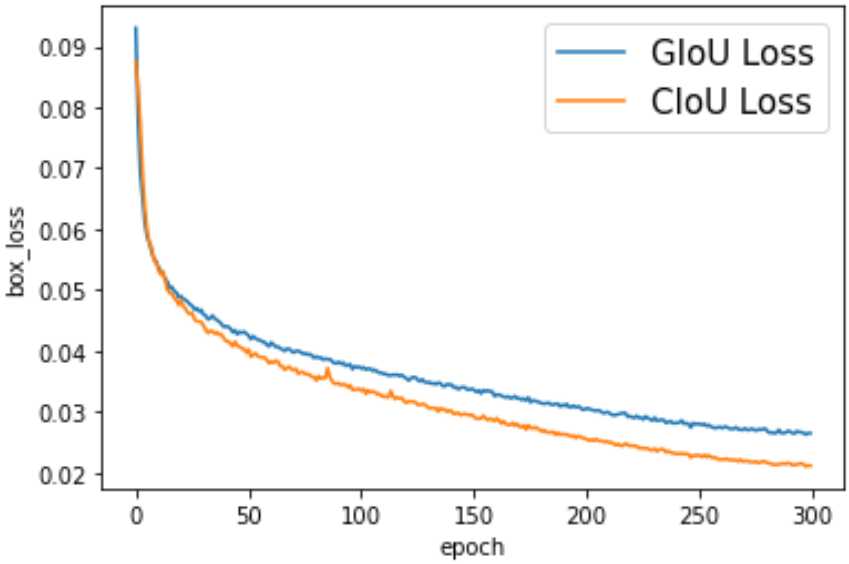

Figure 7. Variation in the two loss functions.

In order to further verify the effectiveness of the improved algorithm, this study includes comparative experiments on the COCO2014 dataset. The experimental results are shown in Table 4.

Table 4. Comparative experiment.

Dataset	Algorithm	Precision	Recall	mAP@0.5	mAP@0.95	F1-Score	FPS
COCO2014	YOLOv5s	64.48%	48.22%	52.72%	33.22%	55.18%	60
	CBAM-YOLOv5s	66.69%	49.10%	54.11%	33.98%	56.56%	58

As can be seen from Table 4, compared with the original YOLOv5s algorithm, the precision, recall and mAP of the CBAM-YOLOv5s algorithm are increased by 2.21%, 0.88% and 1.39%, respectively. Based on the experimental results in Tables 3 and 4, it can be concluded that the improved CBAM-YOLOv5s algorithm is better than the original YOLOv5s algorithm on the VOC2007 dataset and the COCO2014 dataset.

Figure 8 shows the detection effect of the CBAM-YOLOv5s algorithm on the VOC2007 dataset. It can detect different targets in the picture and frame them.

Figure 8. The detection effect of the algorithm.

To verify the effect of the algorithm proposed in this paper, this paper also compares it with other object detection algorithms, as shown in Table 5. The precision and the FPS are used as measurement indicators.

Table 5. Comparison with other algorithms.

Dataset	Algorithm	Backbone	Precision	FPS
VOC2007	SSD	VGG-16	77.5%	46
	ESSD	VGG-16	79.4%	25
	MDSSD	VGG-16	78.6%	28
	YOLOv3	Darknet-53	74.5%	36
	YOLOv4	CSPDarknet53	78.1%	35
	YOLOv5s	CSPDarknet53	75.6%	76
	CBAM-YOLOv5s	CSPDarknet53	80.2%	60

It can be seen from the results in Table 5 that the improved YOLOv5s performs better than the other detection algorithms in terms of precision and FPS. For example, it outperforms the YOLOv4 by 2.1% on the VOC2007 dataset with faster detection.

4. Conclusions

In this paper, a CBAM is incorporated into the backbone network of YOLOv5s to optimize its network structure, and the CIoU loss is used as the object bounding-box regression loss function to accelerate the speed of the regression process. To verify the performance of the proposed algorithm, extensive experiments are conducted on the VOC2007 dataset. The experimental results show that compared with those of the original YOLOv5s, the precision, recall and mAP of the proposed algorithm are significantly improved; furthermore, the CIoU loss is used because the bounding-box regression loss function is faster than the GIoU loss in terms of convergence. The algorithm in this paper solves the problem regarding the low detection accuracy of the original YOLOv5s algorithm to a certain extent, but the algorithm still exhibits certain detection errors and missed detection problems for complex images with dense objects. Future research will involve continuously optimizing the network structure of the proposed algorithm to further improve its detection accuracy.

Author Contributions: Conceptualization, T.J., C.L., M.Y. and Z.W.; data curation, T.J., C.L., M.Y. and Z.W.; methodology, T.J. and C.L.; writing—original draft, T.J. and C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61871258.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, W.; Zhang, Y.; Ge, G.; Jiang, Q.; Wang, Y.; Hu, L. A Hybrid Spatial Indexing Structure of Massive Point Cloud Based on Octree and 3D R*-Tree. *Appl. Sci.* **2021**, *11*, 9581. [CrossRef]
2. Liu, K.; Mulky, R. Enabling autonomous navigation for affordable scooters. *Sensors* **2018**, *18*, 1829. [CrossRef] [PubMed]
3. Conte, G.; Scaradozzi, D.; Mannocchi, D.; Raspa, P.; Panebianco, L.; Screpanti, L. Experimental testing of a cooperative ASV-ROV multi-agent system. *IFAC-PapersOnLine* **2016**, *49*, 347–354. [CrossRef]
4. Kang, T.; Yi, J.B.; Song, D.; Yi, S.J. High-speed autonomous robotic assembly using in-hand manipulation and re-grasping. *Appl. Sci.* **2020**, *11*, 37. [CrossRef]
5. Garcia, A.; Mittal, S.S.; Kiewra, E.; Ghose, K. A convolutional neural network feature detection approach to autonomous quadrotor indoor navigation. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019; pp. 74–81.
6. Levinson, J.; Askeland, J.; Becker, J.; Dolson, J.; Heldet, D.; Kammel, S.; Kolter, J.Z.; Langer, D.; Pink, O.; Pratt, V.; et al. Towards fully autonomous driving: Systems and algorithms. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 163–168.

7. Behrens, T.; Rohr, K.; Stiehl, H.S. Robust segmentation of tubular structures in 3-D medical images by parametric object detection and tracking. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2003**, *33*, 554–561. [CrossRef] [PubMed]
8. Zhou, T.; Qi, S.; Wang, W.; Shen, J.; Zhu, S.C. Cascaded parsing of human-object interaction recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 2827–2840. [CrossRef] [PubMed]
9. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
10. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
11. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
12. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
13. GitHub. YOLOV5-Master. 2021. Available online: https://github.com/ultralytics/yolov5.git/ (accessed on 1 March 2021).
14. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
15. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.
16. Zheng, L.; Fu, C.; Zhao, Y. Extend the shallow part of single shot multibox detector via convolutional neural network. In Proceedings of the Tenth International Conference on Digital Image Processing (ICDIP 2018), Shanghai, China, 11–14 May 2018; International Society for Optics and Photonics: Shanghai, China, 2018; Volume 10806, p. 1080613.
17. Cui, L.; Ma, R.; Lv, P.; Jiang, X.; Gao, Z.; Zhou, B.; Xu, M. MDSSD: Multi-scale deconvolutional single shot detector for small objects. *arXiv* **2018**, arXiv:1805.07009. [CrossRef]
18. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
19. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
20. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
21. Zhou, T.; Li, J.; Wang, S.; Tao, R.; Shen, J. Matnet: Motion-attentive transition network for zero-shot video object segmentation. *IEEE Trans. Image Process.* **2020**, *29*, 8326–8338. [CrossRef] [PubMed]
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
23. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the 2018 IEEE conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
24. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
25. Kim, D.; Park, S.; Kang, D.; Paik, J. Improved Center and Scale Prediction-Based Pedestrian Detection Using Convolutional Block. In Proceedings of the 2019 IEEE 9th International Conference on Consumer Electronics, Berlin, Germany, 8–11 September 2019; pp. 418–419.
26. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
27. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
28. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
29. Bottou, L. Stochastic Gradient Descent Tricks. In *Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 421–436.