applied sciences

MDPI

Article

A Lightweight YOLOv5 Optimization of Coordinate Attention

Jun Wu, Jiaming Dong, Wanyu Nie and Zhiwei Ye *

School of Computer Science, Hubei University of Technology, Wuhan 430068, China
* Correspondence: hgcsyzw@hbut.edu.cn; Tel.: +86-02759750444

Abstract: As Machine Learning technologies evolve, there is a desire to add vision capabilities to all devices within the IoT in order to enable a wider range of artificial intelligence. However, for most mobile devices, their computing power and storage space are affected by factors such as cost and the tight supply of relevant chips, making it impossible to effectively deploy complex network models to small processors with limited resources and to perform efficient real-time detection. In this paper, YOLOv5 is studied to achieve the goal of lightweight devices by reducing the number of original network channels. Then detection accuracy is guaranteed by adding a detection head and CA attention mechanism. The YOLOv5-RC model proposed in this paper is 30% smaller and lighter than YOLOv5s, but still maintains good detection accuracy. YOLOv5-RC network models can achieve a good balance between detection accuracy and detection speed, with potential for its widespread use in industry.

Keywords: object detection; lightweight; deep learning; reduce channels; coordinate attention

check for updates

Citation: Wu, J.; Dong, J.; Nie, W.; Ye, Z. A Lightweight YOLOv5 Optimization of Coordinate Attention. *Appl. Sci.* **2023**, *13*, 1746. https://doi.org/10.3390/app13031746

Academic Editor: Byung-Gyu Kim

Received: 31 December 2022
Revised: 18 January 2023
Accepted: 25 January 2023
Published: 30 January 2023

1. Introduction

With the advent of the Internet of Everything, IoT devices are developing at a rapid pace. We are entering the era of "Artificial Intelligence + Internet of Things" at a fast pace. At the same time, computer vision has been developed for many years and has now entered a phase of large-scale application. Hence, the first idea to enable AI for IoT devices is to add vision to IoT devices. However, for most of the current IoT mobile devices, such as logistics robots, floor sweepers, and commercial drones, the storage space of their devices is affected by cost factors, making it difficult to use high-precision models with large object detection networks. In many application scenarios, the devices in question have limited computing power and memory capacity. It is difficult to deploy large object detection network models with large weight parameters for applications. Compared to traditional machine learning object recognition methods, the current object detection algorithms include one-stage and two-stage categories. The two-stage methods are RCNN [1], Fast R-CNN [2], Faster R-CNN [3], Mask R-CNN [4], SPP-net [5], while the one-stage methods include YOLO [6–9] and SSD [10]. Therefore, it is important to make the mainstream large object detection network models somewhat lighter, which is also a popular and important focus in the field of object detection research.

In today's booming world of deep learning, it is particularly important to build object detection networks with attention mechanisms that are capable of autonomous learning. Attention mechanisms are currently widely used in the field of computer vision. On the one hand, such neural networks can learn the attention mechanism autonomously through back propagation, and on the other hand, the attention mechanism can in turn help us to understand the world seen by the neural network.

To address these issues, this paper selects the YOLOv5 network for the object lightweight improvement. The main idea of the lightweight network is to reduce the number of parameters per layer of convolution by reducing the channel width of the YOLO v5 network model. At the same time, the CA attention mechanism module is added to ensure the accuracy of object detection. By experimentally comparing the position of the CA

attention module, the optimal position is determined, which ultimately compensates for the disadvantage of reduced accuracy of object detection due to the reduced number of channels and network width.

This paper proposes the optimized lightweight YOLOv5-RC (YOLOv5-Reduced channel and Coordinated attention) model. The following innovations are proposed:

(1) To reduce the number of channels of YOLOv5, it is necessary to add a new network to extract different resolution feature maps,
(2) To add a matching detection head based on the feature maps in (1),
(3) To add a coordinated attention module in the best place.

The rest of this paper is as follows: Section 2 describes some existing work and ideas on a lightweight detection model based on deep learning. Section 3 describes how to adjust the number of channels and model replacement in the baseline YOLOv5 network. Section 4 describes the experimental environment, experimental procedures, and result analysis. Finally, Section 5 concludes the entire text.

2. Related Work

2.1. Lightweight Detection Based on Deep Learning

This section reviews some classical approaches to lightweight networks and the contributions made by previous authors to lightweight networks. Recently, a lot of research has focused on multi-class detection based on deep learning and convolutional neural networks. Firstly we would summarize the lightweight detection model of two-stage models. Arun R. A et al [11] demonstrated that Faster R-CNN ResNet101 V1 which developed as a lightweight model as 9MB size outperformed every other model and achieved mAP of 74.77%. Rossi Leonardo et al. [12] designed the Self-Balanced R-CNN (SBR-CNN), an evolved version of the Hybrid Task Cascade (HTC) model, which brought brand new loop mechanisms of bounding box and mask refinements. It shows the same or even better improvements if adopted in conjunction with other state-of-the-art models—in fact, with a lightweight ResNet-50 as backbone. Park J [13] presented a lightweight Mask RCNN model which had the former backbone replaced with MobileNetV2, while the convolution operation of the RPN was replaced with Depthwise Separable Convolution operation. This lightweight Mask RCNN model showed a 64% lower number of parameters compared to the base model, achieving similar mAP to the other base models. Park [14] proposed a lightweight network efficient shot detector (ESDet) based on deep training with small parameters which used depthwise and pointwise convolution to minimize the computational complexity during the feature extraction process.

Although the above two-stage models had be lightweight, they maybe still unsuitable for detecting multi-scale elements in time. Hence, let us focus on the one-stage model. Bouderbal Imene et al. [15] proposed a CS_SSD lightweight model which focused the Single-Shot Detector (SSD) network redesigned to operate in CS networks function, containing a compression-reconstruction approach as an encoder-decoder neural network. Their CS_SSD achieves a compelling accuracy while being 30% faster than its original counterpart on small GPUs. Panigrahi [16] worked on MultiScale MultiLevel SqueezeNetYOLOv3 which is a modified lightweight YOLOv3. Two Squeeze & Expand blocks are embodied in SqueezeNet at specific levels of the network to extract a hierarchical feature representation. Cheng, Rao [17] designed SAS-YOLOv3-tiny detection algorithm which has a light Sandglass-Residual (SR) module based on depthwise separable convolution; the channel attention mechanism is constructed to replace the original convolution layer, and the convolution layer of stride two is used to replace the max-pooling layer for obtaining more informative features and promoting detection performance while reducing the number of parameters and computation. Li [18] proposed a lightweight YOLOv3 with Mobilenetv2 as the backbone which used depthwise separable convolution to replace 3×3 convolutional kernels in the detection head. Gu Y [19] researched a lightweight real-time traffic sign detection integration framework based on YOLOv4 by combining deep learning methods in the field of intelligent transportation. Its framework optimizes

the latency concern by reducing the computational overhead of the network, and facilitates information transfer and sharing at diverse levels. Ma [20] solved the problem of deploying on edge devices with limited computing resources and memory, so that they worked on a lightweight detector named Light-YOLOv4. They performed sparsity training by applying L1 regularization to the channel scaling factors, so the less important channels and layers could be found. Channel pruning and layer pruning would be enforced on the network to prune the less important parts, which could significantly reduce the network's width and depth. Moreover, compared with other state-of-the-art methods, such as SSD and FPN, Light-YOLOv4 is more suitable for edge devices.

2.2. Lightweight YOLOv5

As we know, traditional object detection has resulted in large model size and slow detection speed; as such, the application of object detection technologies under different application environments needs better real-time and lightweight performance. Because of this, a lightweight object detection method based on the You Only Look Once (YOLO) v5 algorithm and attentional feature fusion have been proposed to address this problem, and to produce a harmonious balance between accuracy and speediness for target detection in different environments. Liu W et al. [21] proposed YOLOv5-tassel to detect tassels in UAV-based RGB imagery. A bidirectional feature pyramid network was adopted for the path-aggregation neck to effectively fuse cross-scale features. The robust attention module of SimAM was introduced to extract the features of interest before each detection head. It is better than well-known object detection approaches, such as FCOS, RetinaNet, and YOLOv5. Wan F [22] presented a lightweight model YOLO-LRDD for road damage identification by enhancing the YOLOv5s approach. Its backbone network Shuffle-ECANet worked by adding an ECA attention module into the lightweight model ShuffleNetV2. Furthermore, it employed BiFPN rather than the original feature pyramid network since it improves the network's capacity to describe features. So that YOLO-LRDD provided a good balance of detection precision and speed.

Currently, the standard convolution is the main part of the convolution neural network, which is also the most computationally complex part of the whole network model. The FLOPs and parameters of the convolution can be basically considered as the following equation:

$$FLOPs = k^2 * H_{\text{out}} * W_{\text{out}} * C_{\text{in}} * C_{\text{out}} \quad (1)$$

$$Parameters = k^2 * C_{in} * C_{out} \quad (2)$$

where FLOPs are the number of floating point operations, and can be seen as the amount of computation, k^2 is the size of the convolution kernel, C_{in} and C_{out} are the number of channels of input and output, H_{out} and W_{out} are the width and length of the output feature map.

Many approaches have been used to modify the convolution network in terms of changing the convolution strategy. For example, Howard A et al. proposed MobileNet. They believe that this network pushes the state of the art for mobile-tailored computer vision models; the core is depth-wise separable convolutions and it is a key building block for many efficient neural network architectures [23,24]. Replacing normal convolution with depth-wise separable convolution can significantly reduce the computational effort. In addition, there are similar examples of the ShuffleNet. Thanks to pointwise group convolution with channel shuffle, all components in a ShuffleNet unit can be computed efficiently [25,26]; the group convolution mentioned in the paper is also a convolution strategy to reduce the FLOPs and the number of parameters. Similarly, there are examples in GhostNet. The paper mentions the supposition that the output feature maps are "ghosts" of a handful of intrinsic feature maps with some cheap transformation [27]. It uses the unique ghost convolution. The ghost convolution uses normal convolution to obtain only some features, and then uses a linear operation to obtain some redundant features.

These methods have been shown to be effective, and the use of these convolution strategies to replace the YOLOv5 network could achieve a reduction in FLOPs and the number of parameters, but it also leads to some degradation in detection accuracy. We believe that using these lightweight modules to modify the YOLOv5 network would undermine the integrity of the network model, and in using these strategies to reduce the feature maps obtained by convolution, a portion of the feature maps that are useful for the final detection would inevitably be removed, bringing a reduction in detection accuracy. However, the number of channels of the network model could be adjusted directly according to the size of the data set. Although this would result in obtaining fewer underlying features, since we would adjust the number of channels in a way that matches the size of the data set, it would allow for smoother removal of features that are not very helpful to the results. This would keep the feature maps that are more helpful to the detection results as much as possible. Finally, for a lightweight network, the degradation of detection accuracy could be kept within acceptable limits by using methods that increase the detection head and attention module.

3. Proposal Model

This section will introduce the comparison of the parameters of YOLOv5-RC and YOLOv5s. Then we review how the coordinate attention works, and finally give the changes in the network structure brought about by the addition of the detection head and the attention.

3.1. Reduce the Number of Network Channels

Firstly, the YOLOv5-RC model is focused to reduce the network width of YOLOv5 by lowering the width factor from 0.5 to 0.2 and 0.3, from Equations (1) and (2), when the overall number of channels is reduced by 40% or 60%, and both by 40% or 60%, which should result in a 64% and 84% reduction in both the number of convolution parameters and FLOPs in the backbone.

From Table 1, it can be found that depending on the setting of the width factor, the number of most convolution parameters is reduced by approximately 64% and 84%. The first convolution is used to replace the previous Focus layer with a specific requirement for the number of channels, which made the number of parameters in the first three layers drop less. At the same time, reduction of channels is an intuitive method that inevitably loses some features, which also brings some accuracy loss. We would therefore add a detection head and insert attention in the backbone. Table 1 also reflects the additional layers of the new detection head in the backbone and the new coordinate attention.

Table 1. Comparison of the parameters of the modules in backbone after reducing the number of channels of YOLOv5s by 40% and 60% respectively.

	YOLOv5s			YOLOv5-RC-0.3			YOLOv5-RC-0.2		
Block	**Parameters**	C_{in}	C_{out}	**Parameters**	C_{in}	C_{out}	**Parameters**	C_{in}	C_{out}
Conv	3520	3	32	2640	3	24	1760	3	16
Conv	18,560	32	64	8720	24	40	4672	16	32
C3	18,816	64	64	7440	40	40	4800	32	32
Conv	73,984	64	128	28,960	40	80	16,240	32	56
C3	115,712	128	128	45,440	80	80	22,400	56	56
Conv	295,424	128	256	115,520	80	160	52,624	64	104
C3	625,152	256	256	244,800	160	160	103,792	104	104
Conv	/	/	/	334,544	160	232	150,080	104	160
C3	/	/	/	243,600	232	232	116,160	160	160
Conv	1,180,672	256	512	652,000	232	312	299,936	160	208
C3	1,182,720	512	512	439,920	312	312	195,936	208	208
CA	/	/	/	9075	312	312	5432	208	208
SPPF	656,896	512	512	244,296	312	312	108,784	208	208

3.2. Coordinate Attention

It is important to ensure that the reduction in detection accuracy is not significant. It3 may be useful to provide a brief review of how coordinated attention approach [28–31] works.The CA attention mechanism is achieved by averaging pooling in the horizontal and vertical directions respectively. The spatial information is then encoded using a converter, and finally the spatial information is fused into the channels in a weighted manner so that the CA attention mechanism takes into account the spatial and channel information in a comprehensive manner. Figure 1 shows the workflow of the coordinate attention mechanism. It can be easily found that as the coordinate attention acquires feature mapping in both vertical and horizontal directions separately, it will perform the following operations:

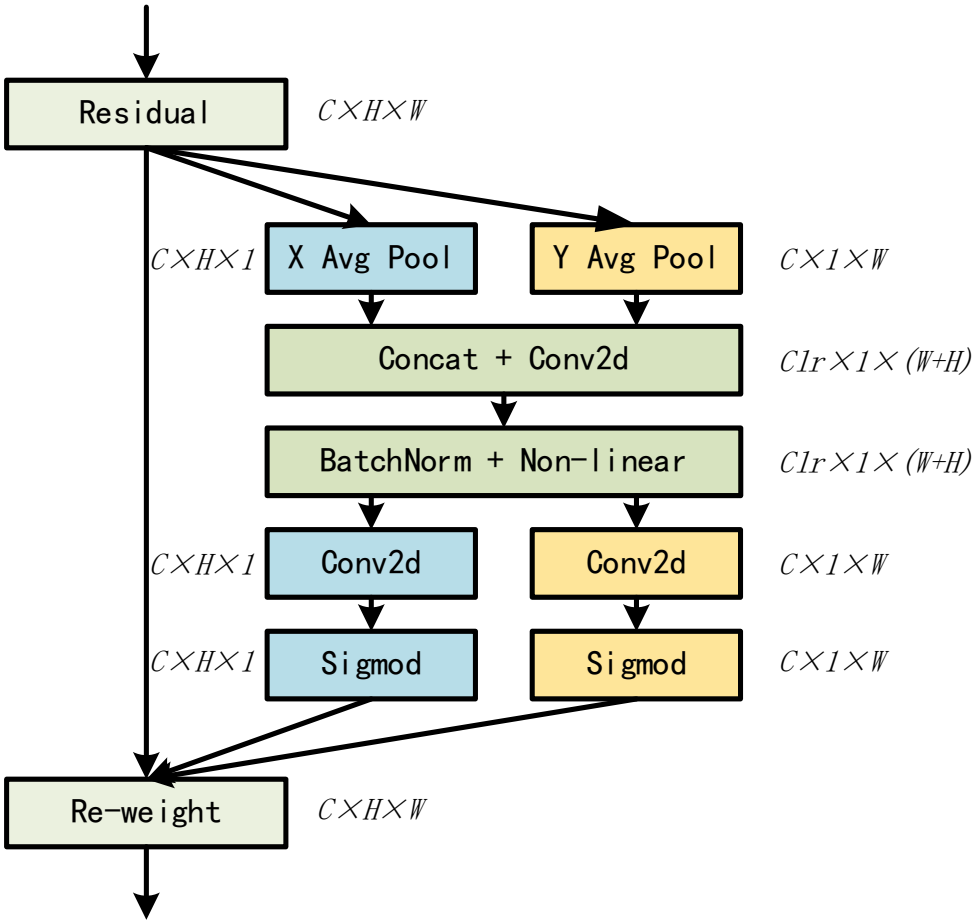

Figure 1. Coordinate Attention working diagram, where AVG pool refers to 1D global pooling, and the notes on both sides are the number of channels and the width and height of the characteristic diagram.

Firstly it uses the pooling kernels of (H,1) and (1,W) to average pooling along the horizontal and vertical. Respectively, its output of the height and width can be expressed as:

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i < W} x_c(h, i) \tag{3}$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le i < H} x_c(j, w) \tag{4}$$

Then, a concatenate operation is performed on the obtained output, followed by a convolution operation and a nonlinear activation function, which is the intermediate feature map that encodes spatial information in both the horizontal direction and the vertical direction.

Finally, the feature mapping is again split along two spatial directions into two separate tensors. Then the resulting tensor is transformed by convolution and activation functions to a tensor that matches the number of channels in the output.

3.3. Modified Network Model

Our proposed YOLOv5-RC network is obtained after modifying the network structure by the previously mentioned method, and the network model changes as shown in Figure 2.

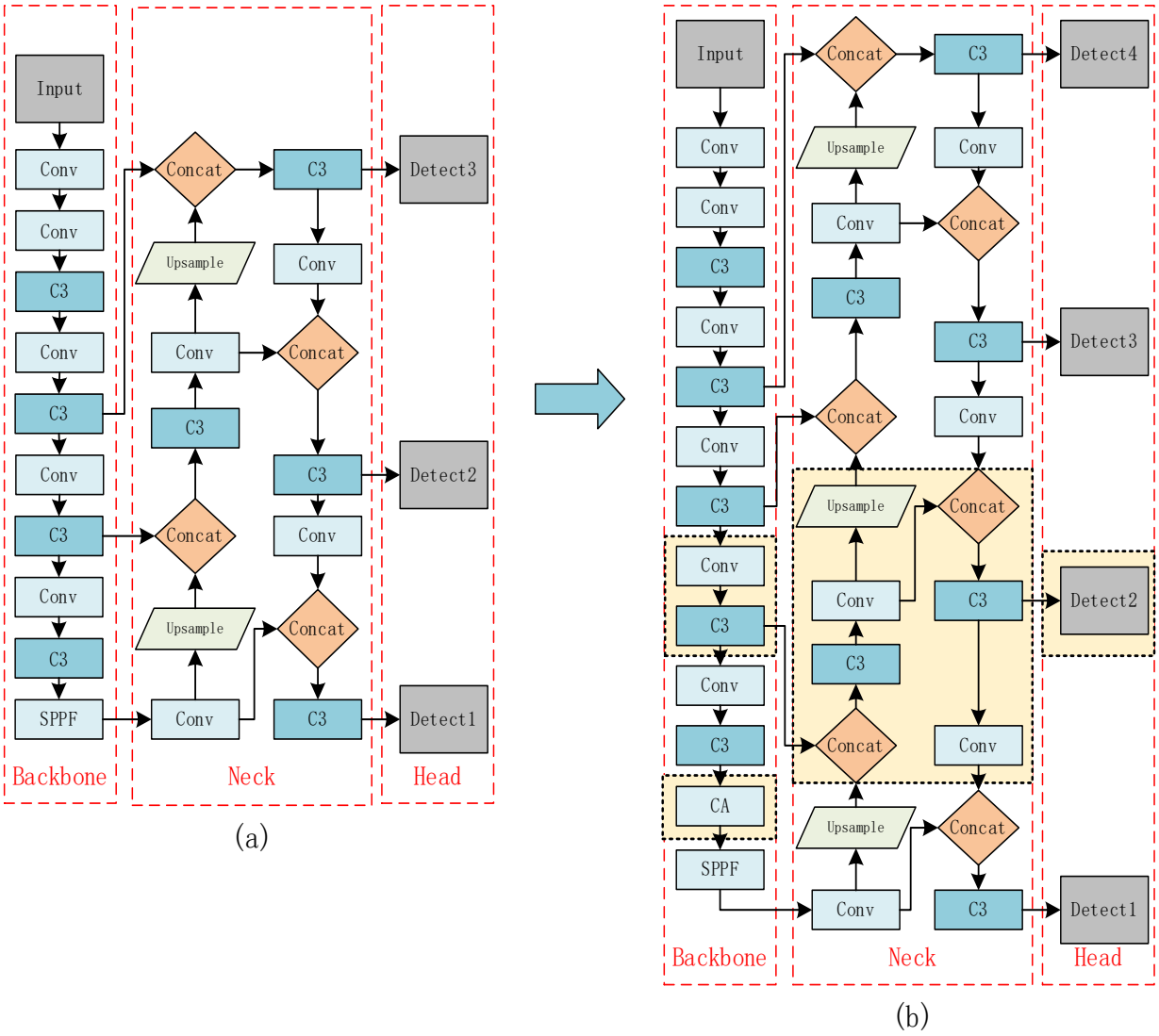

Figure 2. (**a**) YOLOv5, (**b**) YOLOv5-RC, the added structure is marked with light color box.

In this paper we have made four improvements:

(1) To reduce the number of channels of YOLOv5, it is necessary to add a new network to extract different resolution feature maps,
(2) To add a matching detection head based on the feature maps in (1),
(3) To add a coordinated attention module in the best place.

In Figure 2, Conv module is a basic convolution module, which includes a normal convolution layer, a batch normalization layer and an activation function. In particular, the first Conv structure is to split the input image into four copies as a way to improve the receptive field and reduce the loss of original information. C3 is a simplified version of BottleneckCSP that serves to fuse features and reduce computational effort. SPPF serially uses three convolutions of the same convolution kernel and is designed to fuse more features of different resolutions. Relative to Figure 2a, the marked box portion of Figure 2b shows an additional detection head, and the network structure matches the additional detection head.

YOLOv5-RC in Figure 2b adds a new set of Conv and C3 structures in Backbone, which can obtain additional feature maps of different resolutions for input to the feature pyramid composed of FPN+PAN structures. Due to the input of this different set of feature maps, the number of layers of the feature pyramid in Neck is increased from 3 to 4, and the feature maps of the new layer are input into Head. Finally, a new Detect Head with a different object detection size range is added. The best location of attention insertion will be discussed in the next section. In summary, it is obvious that to reduce the number of channels and then add a detection head later seems to be an optimization for lightweight.

4. Experiment and Discussion

This section introduces the data set used for the experiment, and describe the experimental environment and experimental hardware. Then it gives a comparison of YOLOv5-RC with YOLOv5s for two width factors by experimental results and analysis.

4.1. Data Sets and Experiment Environment

The operating environment of the experiment is introduced as shown in Table 2.

Table 2. Experimental Environment.

Hardware and Software	Models and Versions
CPU	Intel(R) Xeon(R) CPU E5-2678 v3 @ 2.50GHz
GPU	NVIDIA GTX 1070Ti
OS	Ubuntu 16.04
Development Language	Python 3.8
Deep Learning Framework	Pytorch 1.8.0

The data sets used include PASCAL VOC2007 and PASCAL VOC2012 and GlobalWheat2020. The VOC data set is one of the commonly used data sets in object detection, where there are 20 categories of object, all of which are common objects in daily life. The training data set is the official training set of VOC2012 and VOC2007 with 16,551 images, and the validation data set and test data set are the official test data set of VOC2007 with 4952 images. The GlobalWheat2020 data set is the data used for the kaggle wheat detection competition, with only 1 object category wheat_head, the training data set including 3422 images, the validation data set with 748 images, and the test data set with 1276 images.

(1) The use of VOC data set is due to the moderate size of the data set, while the detection categories are more appropriate to the actual application. Based on this size of the data set, we reduce the number of channels for light weighting to produce a more intuitive end result.
(2) The GlobalWheat2020 data set has a smaller size and fewer object categories, and performing lightweight work on such a data set can visualize the advantages of the approach in this paper, for which we use smaller channel number factors, thus verifying the generalizability of this approach.

4.2. Result Evaluation

For the lightness of the network, FLOPs and Parameters can be evaluated more intuitively, the smaller these two metrics are, the simpler and lighter the network structure is. In addition, the FPS of the test is used as the evaluation criterion, and the FPS in this paper is the inference speed obtained when the batch size is set to 32; the FPS is more focused on the evaluation of the speed performance in actual use than the first two evaluations of the network model. As for the detection accuracy, mAP50 refers to the AP calculation when the average iou threshold is set to 0.5 under different categories, and a larger mAP50 indicates that the detection accuracy of the model is better. The goal of this paper is to keep the decrease in mAP50 small and significantly reduce both GFLOPs and Parameters while improving FPS.

4.3. Better Use of Coordinate Attention

Attention often has better results, but little attention is paid to how to use it better. Specifically, coordinate attention cannot be used anywhere in the network without restrictions, but should have certain methods. On the question of how to better use attention, we conducted ablation experiments by inserting the coordinate attention into different positions of the YOLOv5s network backbone, so as to find the best working location of the coordinate attention, and applied this law to our YOLOv5-RC. This experiment was conducted based on a slightly earlier version of YOLOv5.

In Table 3, the first column shows the insertion position of the coordinate attention in the backbone of YOLOv5s, the second column shows the mAP50 of the corresponding experiment, and the third column shows the difference in mAP50 between the corresponding experiment and the previous position. It can be found that the coordinate attention works better in the backward position of the Backbone, and we believe this is because the attention is better able to guide the network to learn valuable features at the larger number of channels. To show this trend more visually, a line graph was drawn after summarizing the data in Table 3.

Table 3. Ablation Experiment of the best working position of attention.

Model	mAP50	mAP50 Change
YOLOv5s-CA-1	71.3	-
YOLOv5s-CA-2	66.6	−4.7
YOLOv5s-CA-3	73.5	+6.9
YOLOv5s-CA-4	69.4	−4.1
YOLOv5s-CA-5	77.1	+7.7
YOLOv5s-CA-6	73.5	−3.6
YOLOv5s-CA-7	80.3	+6.8
YOLOv5s-CA-9	79.9	−0.4
YOLOv5s-CA-10	81.8	+1.9

As shown in Figure 3, the *x*-axis indicates the location of the CA module inserted into YOLOv5's backbone. It is easy to find that the accuracy of the network gradually increases as the coordinate attention insertion location is moved backwards. Finally, from the above results, the best location of coordinate attention insertion is the 10th which is nearly the end of YOLOv5's Backbone. We believe that coordinate attention has dual attention characteristics of space and channel, and the spatial attention aspect will be influenced by the Conv and C3 structures. In YOLOv5's Backbone, Conv will extract features to make the feature map smaller, while C3 will fuse features to make them richer. Therefore, when inserting coordinate attention in Backbone in turn, detection accuracy will show a wave-like trend after each Conv and C3. In YOLOv5's Backbone, the number of channels gradually increases, and the channel attention part of coordinate attention will play a bigger role, making the overall trend of detection accuracy rise.

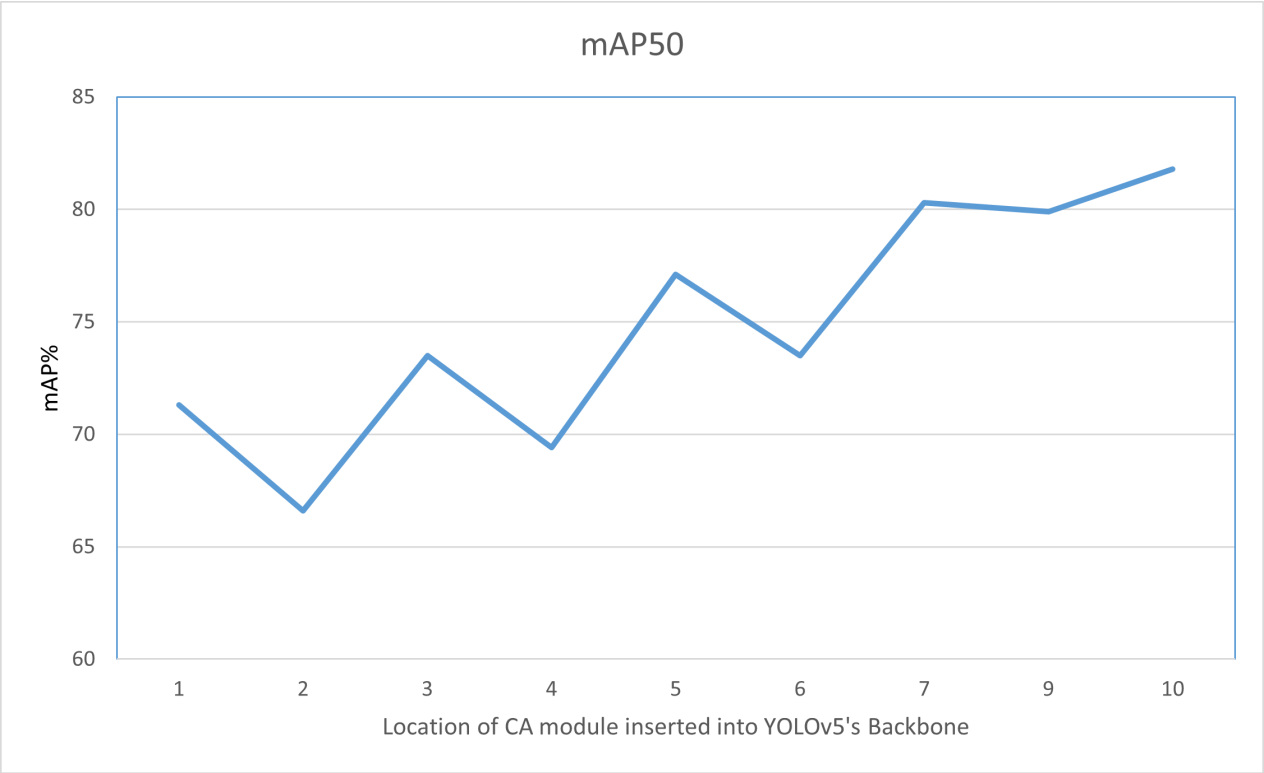

Figure 3. The impact of coordinate attention on accuracy when inserted into YOLOv5's Backbone.

4.4. Result Analysis

This section takes YOLOv5s as the baseline, gives the three lightweighting methods we mentioned, and then compares it with our YOLOv5-RC, where YOLOv5s-ghost refers to the use of ghost convolution instead of the original Backbone in convolution.

In Table 4, the first column indicates the name of the modified network, the second column is the size of the image used, and the third and fourth columns are the previously mentioned network model evaluation criteria Parameters and GFLOPs, which can visualize the degree of lightness of the network structure. The table includes YOLOv5s as the baseline, and YOLOv5 networks modified using some special convolution strategies, as well as YOLOv5-RC networks with width factors of 0.2 and 0.3. It can be observed that the YOLOv5-RC network Parameters with a width factor of 0.2 is only 2.14M and the FLOPs are 3.4G, which are 30% and 20% of the baseline, respectively. In addition, these two figures are also lower than the YOLOv5 network modified with three special strategies, which can indicate that the proposed method in this paper can reduce the complexity of the network very well. As opposed to the intuitive criteria given in Table 3, the results given in Table 4 are the actual data obtained from the network model during training and validation. Similarly, the evaluation criteria mAP50 and FPS mentioned previously are used to measure the real detection accuracy and detection speed of YOLOv5-RC.

Table 4. Parameters and GFLOPs in YOLOv5 lightweight work under different methods.

Model	Image Size	Parameters (M)	GFLOPs
YOLOv5s(baseline)	640	7.11	16.5
YOLOv5-ghost	640	3.73	8.2
YOLOv5-mobilenet	640	3.59	6.4
YOLOv5-shufflenet	640	4.8	4.8
YOLOv5-RC-0.2	640	2.14	3.4
YOLOv5-RC-0.3	640	4.67	6.7

Table 5 uses different evaluation criteria. The third column Speed refers to the time used to detect each graph, which mainly includes Inference and NMS time consumption. Speed can be calculated from the fourth column corresponding to the FPS. FPS is an important symbol to evaluate the speed of a network detection. The mAP50 in the fifth column responds to the detection accuracy of the network model. The results are similar to those in Table 3. In terms of lightweight, it can be observed that the FPS of the first three methods does improve, but at the cost of a larger decrease in mAP50, which is due to the use of some lightweight convolution strategies causing the inevitable loss of some features during feature extraction. In comparison, YOLOv5-RC-0.2 shows a significant improvement in FPS, but at the same time the mAP50 drops by only 4.8, while YOLOv5-RC-0.3 shows a mAP50 drop of only 2.9. The use of two different width factors reflects the flexibility to adjust to the size of the data set and the actual problem requirements.

Table 5. FPS and mAP50 in YOLOv5 lightweight work under different methods.

Model	Epoch	Speed (ms)	FPS	mAP50
YOLOv5s(baseline)	300	7.2	138.9	84.7
YOLOv5-ghost	300	6.0	166.7	75.9
YOLOv5-mobilenet	300	5.2	192.3	70.9
YOLOv5-shufflenet	300	4.9	204.1	76.2
YOLOv5-RC-0.2	300	4.8	208.3	79.9
YOLOv5-RC-0.3	300	5.9	169.5	81.8

The ablation experiments with the insertion of the coordinate attention were mentioned in Section 4.3 of this paper. From the accuracy point of view, the mAP50 with the insertion of the coordinate attention at the optimal position is still lower than that of YOLOv5s as the baseline. In the more lightweight YOLOv5-RC, however, the coordinate attention has a certain improvement on the mAP50, which could be found in Table 6. In this case, it may be because the attention mechanism makes the network model fit faster and also exacerbates the over-fitting problem, which is more prominent for large and complex network models and has less impact on lightweight network models. In addition, we observed that the main time-consuming Inference part of the detection speed YOLOv5-RC has a significant reduction. The NMS time-consuming of YOLOv5-RC is more compared to YOLOv5s, for which we did not modify the hyper parameters meticulously due to the limitation of the device, which indicates that YOLOv5-RC still has room for improvement, and this is one of the directions for subsequent research.

Table 6. Result table of changing the number of channels according to the size of data set.

Model	Data Set	Parameters (M)	GFLOPs	FPS	mAP50
YOLOv5s	VOC	7.11	16.5	138.9	84.7
YOLOv5-RC-0.2	VOC	2.14	3.4	208.3	79.9
YOLOv5-RC-0.3	VOC	4.67	6.7	169.5	81.8

To verify the generalized ability of YOLOv5-RC, we tested YOLOv5-RC on the GlobalWheat2020, and made a comparison with other lightweight YOLO models which are shown in Table 7.

(1) First, in the comparison of the indicators for the Parameters(M), it is to find significant YOLOv5-RC using fewest parameters and GFLOPs.
(2) In order to apply the lightweight object detection model in mobile equipment, we need higher FPS than traditional application scenarios. Our YOLOv5-RC gets the highest FPS in the comparison of relevant detection networks.
(3) Finally, the accuracy is also a key evaluation indicator. Because of the reduction of channels, YOLOv5-RC gets less mAP than the original YOLOv5, but is better than other lightweight object detection models.

Table 7. Results for the data set GlobalWheat2020.

Model	Data Set	Parameters (M)	GFLOPs	FPS	mAP50
YOLOv3	GlobalWheat2020	61.49	154.7	22.6	94.3
YOLOv3-tiny	GlobalWheat2020	8.67	12.9	69.4	89.7
YOLOv5	GlobalWheat2020	142.98	120.9	16.3	94.5
YOLOv5s	GlobalWheat2020	7.1	16.5	34.8	95
YOLOv5-Ghost	GlobalWheat2020	3.68	8.1	45.5	92.4
YOLOv5s-transformer	GlobalWheat2020	7.05	16.1	9.9	93.7
YOLOv5-RC	GlobalWheat2020	0.56	1.0	86.2	94.2

5. Conclusions

In this paper, we propose the following steps: (1) Reduce the number of channels, (2) add a new network to extract different resolution feature maps, (3) add a matching detection head based on the feature maps in (2), and (4) add an attention module.

With two width factors, the above method reduces the number of parameters to 30% and 66% and FLOPs to 20% and 40%, resulting in a reduction in the number of parameters to 30% and 66% and FLOPs to 20% and 40%, but with a small decrease in mAP. In addition, for the use of attention in YOLOv5, we also mentioned: (1) coordinate attention can be inserted in the last position of Backbone, which is the position with the largest number of channels and works best, (2) attention helps network fitting, which works better on lightweight networks and may have the opposite effect in complex networks. The method proposed in this paper ensures the integrity of the network and is more robust than previous convolution modification of YOLOv5 networks using various special strategies, and is also simpler and easier to use in practical applications. This approach provides ideas and references for the construction of lightweight networks. In addition, the reduction of the number of channels should relate to the size of the data set and the number of target categories. Two different data sets, VOC and GlobalWheat2020, were used in the experiments to verify the generalizability of this method. However, since we only verified the feasibility of this method in YOLO series networks, whether this method has a similar relevance on other network models is the direction of subsequent research.

Author Contributions: Conceptualization, J.W. and J.D.; methodology, J.D.; software, J.D.; validation, J.D., J.W. and W.N.; formal analysis, J.D. and Z.Y.; investigation, J.D.; resources, J.D.; data curation, J.D.; writing—original draft preparation, J.W. and J.D.; writing—review and editing, J.D. and Z.Y.; visualization, J.D.; supervision, Z.Y.; project administration, J.D.; funding acquisition, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (Grant No. 61602161, 61772180), Hubei Province Science and Technology Support Project (Grant No: 2020BAB012), and The Fundamental Research Funds for the Research Fund of Hubei University of Technology (HBUT: 2021046).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets that support this study are openly available online.

Acknowledgments: The authors would like to thank the anonymous reviewers for their constructive suggestions, which comprehensively improved the quality of the paper.

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

References

1. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 142–158. [CrossRef] [PubMed]
2. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]

3. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 1440–1448. [CrossRef]
4. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [CrossRef] [PubMed]
5. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
6. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
7. Bochkovskiy, A.; Wang, C.; Liao, H. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934. [CrossRef]
8. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525. [CrossRef]
9. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767. [CrossRef]
10. 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, 11–14 October 2016; pp. 21–37. [CrossRef]
11. Arun, R.A.; Umamaheswari, S. Effective and efficient multi-crop pest detection based on deep learning object detection models. *J. Intell. Fuzzy Syst.* **2022**, *43*, 5185–5203. [CrossRef]
12. Rossi, L.; Karimi, A.; Prati, A. Self-Balanced R-CNN for instance segmentation. *J. Vis. Commun. Image Represent.* **2022**, *87*, 103595. [CrossRef]
13. Park, J.; Moon, H. Lightweight Mask RCNN for Warship Detection and Segmentation. *IEEE Access* **2022**, *10*, 24936–24944. [CrossRef]
14. Park, C.; Lee, S.; Han, H. Efficient Shot Detector: Lightweight Network Based on Deep Learning Using Feature Pyramid. *Appl. Sci.* **2021**, *11*, 8692. [CrossRef]
15. Bouderbal, I.; Amamra, A.; Djebbar, M.E.-A.; Benatia, M.A. Towards SSD accelerating for embedded environments: A compressive sensing based approach. *J. Real-Time Image Process.* **2022**, *19*, 1199–1210. [CrossRef]
16. Panigrahi, S.; Rajub, U. MS-ML-SNYOLOv3: A robust lightweight modification of SqueezeNet based YOLOv3 for pedestrian detection. *Optik* **2022**, *260*, 169061. [CrossRef]
17. Cheng, R.; He, X.; Zheng, Z.; Wang, Z. Multi-Scale Safety Helmet Detection Based on SAS-YOLOv3-Tiny. *Appl. Sci.* **2021**, *11*, 3652. [CrossRef]
18. Li, H.; Liu, L.; Du, J.; Jiang, F.; Guo, F.; Hu, Q.; Fan, L. An Improved YOLOv3 for Foreign Objects Detection of Transmission Lines. *IEEE Access* **2022**, *10*, 45620–45628. [CrossRef]
19. Gu, Y.; Si, B. A Novel Lightweight Real-Time Traffic Sign Detection Integration Framework Based on YOLOv4. *Entropy* **2022**, *24*, 487. [CrossRef]
20. Ma, X.; Ji, K.; Xiong, B.; Zhang, L.; Feng, S.; Kuang, G. Light-YOLOv4: An Edge-Device Oriented Target Detection Method for Remote Sensing Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2021**, *14*, 10808–10820. [CrossRef]
21. Liu, W.; Quijano, K.; Crawford, M.M. YOLOv5-Tassel: Detecting Tassels in RGB UAV Imagery with Improved YOLOv5 Based on Transfer Learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2022**, *15*, 8085–8094. [CrossRef]
22. Wan, F.; Sun, C.; He, H.; Lei, G.; Xu, L.; Xiao, T. YOLO-LRDD: A lightweight method for road damage detection based on improved YOLOv5s. *EURASIP J. Adv. Signal Process.* **2022**, *2022*, 98. [CrossRef]
23. Howard, A.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.-C.; Tan, M.; Chu, G.; Vasudevan, V.; Zhu, Y.; Pang, R.; et al. Searching for MobileNetV3. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324. [CrossRef]
24. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]
25. Zhang, X.Y.; Zhou, X.Y.; Lin, M.X.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856. [CrossRef]
26. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131. [CrossRef]
27. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features from Cheap Operation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1577–1586. [CrossRef]
28. Hou, Q.B.; Zhou, D.Q.; Feng, J.S. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13708–13717. [CrossRef]
29. Liu, X.; Zhang, B.; Liu, N. CAST-YOLO: An Improved YOLO Based on a Cross-Attention Strategy Transformer for Foggy Weather Adaptive Detection. *Appl. Sci.* **2023**, *13*, 1176. [CrossRef]

30. Tian, Z.; Huang, J.; Yang, Y.; Nie, W. KCFS-YOLOv5: A High-Precision Detection Method for Object Detection in Aerial Remote Sensing Images. *Appl. Sci.* **2023**, *13*, 649. [CrossRef]
31. Kim, M.; Kim, Y. Parcel Classification and Positioning of Intelligent Parcel Storage System Based on YOLOv5. *Appl. Sci.* **2023**, *13*, 437. [CrossRef]