remote sensing

MDPI

Article

A Lightweight Object Detection Algorithm for Remote Sensing Images Based on Attention Mechanism and YOLOv5s

Pengfei Liu, Qing Wang *, Huan Zhang, Jing Mi and Youchen Liu

School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China; 230208375@seu.edu.cn (P.L.); 230198295@seu.edu.cn (H.Z.); 230218950@seu.edu.cn (J.M.); 230228958@seu.edu.cn (Y.L.)

* Correspondence: wq_seu@seu.edu.cn

Abstract: The specific characteristics of remote sensing images, such as large directional variations, large target sizes, and dense target distributions, make target detection a challenging task. To improve the detection performance of models while ensuring real-time detection, this paper proposes a lightweight object detection algorithm based on an attention mechanism and YOLOv5s. Firstly, a depthwise-decoupled head (DD-head) module and spatial pyramid pooling cross-stage partial GSConv (SPPCSPG) module were constructed to replace the coupled head and the spatial pyramid pooling-fast (SPPF) module of YOLOv5s. A shuffle attention (SA) mechanism was introduced in the head structure to enhance spatial attention and reconstruct channel attention. A content-aware reassembly of features (CARAFE) module was introduced in the up-sampling operation to reassemble feature points with similar semantic information. In the neck structure, a GSConv module was introduced to maintain detection accuracy while reducing the number of parameters. Experimental results on remote sensing datasets, RSOD and DIOR, showed an improvement of 1.4% and 1.2% in mean average precision accuracy compared with the original YOLOv5s algorithm. Moreover, the algorithm was also tested on conventional object detection datasets, PASCAL VOC and MS COCO, which showed an improvement of 1.4% and 3.1% in mean average precision accuracy. Therefore, the experiments showed that the constructed algorithm not only outperformed the original network on remote sensing images but also performed better than the original network on conventional object detection images.

Keywords: object detection; remote sensing; YOLOv5s; DD-head; SPPCSPG; content-aware reassembly of features; GSConv

Citation: Liu, P.; Wang, Q.; Zhang, H.; Mi, J.; Liu, Y. A Lightweight Object Detection Algorithm for Remote Sensing Images Based on Attention Mechanism and YOLOv5s. *Remote Sens.* **2023**, *15*, 2429. https://doi.org/10.3390/rs15092429

Academic Editors: Weifeng Liu, Igor García Olaizola and Bingfeng Zhang

Received: 4 April 2023
Revised: 29 April 2023
Accepted: 4 May 2023
Published: 5 May 2023

1. Introduction

In the era of big data, due to the development and progress of machine learning technology represented by deep learning, artificial intelligence can better cope with complex environments and tasks through continuous learning and adaptation. By processing and analyzing large-scale and high-dimensional data [1], it can reveal the patterns and patterns behind the data, supporting more accurate predictions and decisions [2–4]. By exploring potential patterns in data, new knowledge and innovation can be discovered, supporting the development of various sciences and technologies. Therefore, in recent years, artificial intelligence has been gradually applied to speech recognition, natural language processing, computer vision, and other fields. In the field of artificial intelligence, computer vision technology has found widespread application in diverse fields such as intelligent security [5], autonomous driving [6], remote sensing monitoring [7,8], medical and pharmaceuticals [9,10], agriculture [11], intelligent transportation [12], and information security [13]. Computer vision tasks can be categorized into image classification [14], object detection [15], and image segmentation [16]. The core task of object detection is to determine the categories and positions of multiple objects in the image and to give

corresponding detection boxes and object categories for each object. Remote sensing images contain a wealth of detailed information, which can intuitively reflect the shape, color, and texture of ground targets. Remote sensing target detection, as a fundamental technique, is widely applied in various fields, such as urban planning [17], land use [18], traffic guidance [19], and military surveillance [20]. With the development of ground observation technology, the scale of high-resolution remote sensing image data has been continuously increasing. High-resolution remote sensing images provide higher image quality and more abundant, detailed information, which presents greater opportunities for the development of target detection in the field of remote sensing. Target detection can be divided into two main categories: traditional object detection algorithms and deep learning-based object detection algorithms. Traditional object detection algorithms mainly rely on traditional feature extractors [21] and use sliding windows to generate object candidate regions. Representative algorithms include the Viola–Jones detector (VJ-Det) [22], the Histogram of Oriented Gradient (HOG) detector [23], and the deformable part model detector (DPM) [24]. With the development of deep learning, convolutional neural networks (CNNs) have gradually been applied to object detection tasks. Based on deep learning, object detection technology can use multi-structured network models and powerful training algorithms to adaptively learn high-level semantic information from images. Image features are extracted and fed into a classification network to complete the tasks of object classification and localization, thereby effectively improving the accuracy and efficiency of object detection tasks.

According to the detection principle, target detection algorithms based on deep learning can be divided into two categories: (1) Two-stage target detection algorithms based on candidate regions. Representative algorithms include R-CNN [25], Fast R-CNN [26], and Faster R-CNN [27]. This algorithm first generates sample candidate boxes [28–30], then encodes the extracted feature vectors using deep convolutional neural networks [31–33], and finally performs regression on the class and location of the target object within the candidate box [34]. By employing a two-stage operation, the target detection algorithm achieves high detection accuracy at the expense of slower speed and non-real-time detection. (2) One-stage object detection algorithms based on direct regression, represented by algorithms such as SSD [35] and the YOLO series [36–38]. This algorithm abandons the stage of generating candidate bounding boxes and directly outputs the position and category of the target through regression, which improves the detection speed.

The YOLO series is currently a classical one-stage object detection algorithm. Redmon et al. [39] proposed the YOLO algorithm, which represented a significant breakthrough in real-time object detection. However, the training of each component in YOLO needs to be conducted separately, leading to a slow inference speed. A strategy of jointly training the components was proposed, which not only improved the inference speed but also reduced the complexity of network training while enhancing the detection performance of YOLO. The YOLOv3 algorithm [37] utilizes Darknet-53 as the backbone network and fuses up-sampled feature maps with shallow feature maps to retain the semantic information of small objects and enhance the detection performance of such objects. The YOLOv4 algorithm [38] adopts CSPDarknet53 [40] as the backbone network and introduces Spatial Pyramid Pooling (SPP) [41] to optimize the receptive field of deep feature maps, thereby further improving detection accuracy. Ultralytics released the YOLOv5 algorithm, which incorporates CSPNet as its backbone network. The neck component employs a feature pyramid network (FPN) [42] to enable top-down semantic information transmission, leveraging both low-level features with high resolution and high-level features with semantic information. In addition, the algorithm utilizes a path aggregation network (PAN) [43] for bottom-up localization transmission, which facilitates the propagation of low-level information to the top level. The YOLOv5 model proposes five types—YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x—based on the differences in network structure depth and width. While the parameters and performance of the models increase sequentially, the detection speed gradually decreases. The YOLOv5n/s models have a small backbone

feature extraction network and are lightweight, but their target bounding box regression is not sufficiently accurate for practical applications. The YOLOv5m/l/x models have better detection and recognition performance with increasing network depth and width, but they struggle to meet real-time detection requirements on hardware-limited embedded devices. To address this problem, this paper proposes a lightweight object detection algorithm based on the YOLOv5s network, which combines the shuffle attention (SA) module, the depthwise-decoupled head (DD-head) module, the content-aware reassembly of features (CARAFE) module, the GSConv module, and the spatial pyramid pooling cross-stage partial GSConv (SPPCSPG) module to improve the detection accuracy of the model while meeting real-time requirements. The main contributions and innovations of the constructed model can be summarized as follows:

(1) In this paper, depthwise convolution is used to replace the standard convolution in the decoupled head module to construct a new detection head, the DD-head, which can improve the negative impact of classification and regression task conflicts while reducing the parameter volume of the decoupled head.
(2) Based on the SPPCSPC module, this paper utilizes the design principle of the GS bottleneck and replaces the CBS module in the SPPCSPC module with the GSConv module to design a lightweight SPPCSPG module, which is introduced into the backbone structure to optimize the YOLOv5s network model.
(3) The effect of embedding the SA module in the network backbone, neck, and head regions is studied, and the SA module is ultimately embedded in the head region to enhance the spatial attention and channel attention of the feature map, thereby improving the accuracy of multi-scale object detection.
(4) The CARAFE module is used to replace the nearest neighbor interpolation up-sampling module to reassemble feature points with similar semantic information in a content-aware manner and aggregate features in a larger receptive field, achieving the up-sampling operation.
(5) In this study, the Conv module in the variety of view-GS cross-stage partial (VoV-GSCSP) module is replaced by the GSConv module to reconstruct a new VoV-GSCSP module to further reduce the model's parameters. The GSConv module and the improved VoV-GSCSP module are embedded in the neck structure to maintain the model's detection accuracy while reducing the parameter volume.

Experimental results show that the proposed algorithm outperforms the original YOLOv5s algorithm in multi-scale object detection performance while meeting real-time requirements.

2. Related Work

2.1. Object Detection Algorithms for Remote Sensing Images

Traditional remote sensing image object detection algorithms are based on handcrafted feature design. The detection process typically includes candidate region extraction [44], feature extraction [45], classifier design [46], and post-processing. First, potential target regions are extracted from the input image using candidate region extraction. For each region, features are extracted; then, the extracted features are classified. Finally, post-processing, such as filtering and merging, is applied to all candidate boxes to obtain the final detection results. Candidate region extraction requires the setting of a large number of sliding windows, which results in a high time complexity and a significant amount of redundant computation. Handcrafted features are mainly extracted based on target visual information (such as color [47], texture [48], edges [49], context [50], etc.), giving them strong interpretability. However, handcrafted features have weak feature expression capability, poor robustness, limited adaptability, and are difficult to apply in complex and changing environments.

With the development of deep learning, the deep features extracted by neural networks have stronger semantic representation and discriminative ability. However, due to the characteristics of remote sensing images, such as large image size, significant directional changes, large-scale small targets, dense target distribution, significant scale variations,

target blurring, and complex backgrounds, existing detection algorithms cannot achieve satisfactory performance on remote sensing images. To address the problem of large image size, R^2-CNN [51] was designed using a lightweight backbone network, Tiny-Net, for feature extraction and used the approach of judging first and then locating to filter out sub-image blocks without targets, thereby reducing the computational burden of subsequent detection and recognition. For the problem of significant directional changes, the approaches of data augmentation [52] or adding rotation-invariant sub-modules [53] are typically used to solve this problem. Cheng et al. [54] explicitly increased rotation-invariant regularizers on CNN features by optimizing a new objective function to force the feature representation of training samples before and after rotation to be closely mapped to achieve rotational invariance. For the small target problem, Yang et al. [55] increased the number and scale of shallow feature pyramids to improve the detection accuracy of small targets and used a dense connection structure to enhance the feature expression ability of small targets. Zhang et al. [56] improved small target detection by up-sampling and enlarging the feature map size of each candidate region in the first stage of the two-stage Faster R-CNN. To address the problem of dense target distribution, DAPNet [57] used an adaptive region generation strategy based on the density of targets in the image. For the problem of significant scale variations, Guo et al. [58] and Zhang et al. [59] directly used a multi-scale candidate region network and a multi-scale detector to detect targets of different scales. For the problem of target blurring, Li et al. [60] proposed a dual-channel feature fusion network that can learn local and contextual attributes along two independent paths and fuse features to enhance discriminative power. Finally, for the problem of complex backgrounds, Li et al. [61] extracted multi-scale features and used an attention mechanism to enhance each feature map individually, thereby eliminating the influence of background noise.

2.2. Attention Mechanism

To extract effective information from massive and complex data, researchers have proposed attention mechanisms to obtain the importance differences of each feature map. In the visual system, attention mechanisms are considered as dynamic selection processes that adaptively weigh the features based on their importance differences in the input [62]. Currently, attention mechanisms have achieved good performance in tasks such as image classification [63], object detection [64], semantic segmentation [65], medical image processing [66], super-resolution [67], and multimodal tasks [68]. Attention mechanisms can be classified into the following six categories. (1) Channel attention: In deep neural networks, different channels in various feature maps typically represent different objects. Channel attention adaptively adjusts the weight of each channel to increase the importance of focused objects. Hu et al. [69] were the first to propose the concept of channel attention and to introduce SENet. The SENet module collects global information [70] using squeeze and excitation modules, captures channel relationships, and improves the representation power of the network. (2) Spatial attention: Spatial attention is an adaptive mechanism for selecting spatial regions. The representative algorithms of spatial attention include RAM [71], based on the RNN method; STN, which uses subnetworks to explicitly predict relevant regions [72]; GENet [73], which uses subnetworks implicitly to predict soft masks for selecting important regions; and GCNet [74], which uses a self-attention mechanism [75]. (3) Temporal attention: Temporal attention can be regarded as a dynamic temporal selection mechanism for determining when to focus attention, typically achieved by capturing short-term and long-term cross-frame feature dependencies [76,77]. Li et al. [76] proposed a global–local temporal representation (GLTR) to utilize multi-scale temporal information in video sequences. GLTR consists of a dilated temporal pyramid (DTP) for local temporal context learning and a temporal self-attention module for capturing global temporal interaction. (4) Branch attention: Branch attention can be regarded as a dynamic branch selection mechanism; it is often used in conjunction with multi-branch structures. Representative networks include highway networks [78], Selective Kernel (SK) convolution [79], CondCov operator [80], and dynamic convolution [81]. (5) Channel and spatial attention: Channel

and spatial attention combines the advantages of channel and spatial attention and can adaptively select important objects and regions [82]. Based on the ResNet network [83], the Residual Attention Network [84] pioneers the research of channel and spatial attention by combining attention mechanism and residual connection, emphasizing the importance of information features in spatial and channel dimensions. Woo et al. [63] proposed the convolutional block attention module (CBAM) by concatenating channel and spatial attention; it decouples channel and spatial attention to improve computational efficiency. (6) Spatial and temporal attention: Combining the advantages of spatial and temporal attention, it can adaptively select important regions and key frames. Song et al. [85] proposed a joint spatial and temporal attention network based on LSTM [86], which enables the adaptive discovery of discriminative features and key frames.

2.3. Multi-Scale Feature Fusion

In object detection tasks, feature maps at different levels represent varying information about the detection targets. High-level feature maps encode semantic information about the objects which can be used for classification, while low-level feature maps encode positional information about the objects which can be used for regression [87]. The YOLO algorithms fuse multiple features obtained from neural networks to extract more information about small targets to improve detection accuracy. The Feature Pyramid Network (FPN) [42] enhances semantic feature representation through a top-down pathway and fuses features with more precise location information. However, FPN fails to propagate accurate localization information from lower-level feature maps to higher-level semantic feature maps, and the feature transfer between non-adjacent layers is limited. In addition, for masks generated for large targets, the redundant and lengthy spatial transfer path hinders the effective integration of high-level and low-level information, leading to information loss. Liu et al. [43] proposed the PANet network, which incorporates a bottom-up pathway enhancement structure and integrates shallow network features with FPN features. To improve upon the suboptimal fusion performance of manually designed feature pyramids, Ghiasi et al. [88] introduced the neural architecture search-feature pyramid network (NAS-FPN), which utilizes neural network architecture search methods to automatically design the feature network. The bidirectional feature pyramid network (BiFPN) [89] improves upon the PANet by introducing contextual [90] and weight information to balance features of different scales, resulting in a larger receptive field and richer semantic information. To address the issue of inconsistent feature scales in pyramid-based methods, Liu et al. [91] proposed a data-driven pyramid feature fusion strategy called adaptive spatial feature fusion (ASFF), which enables the network to learn how to directly filter out features from other levels in space to preserve useful information for combination. Subsequent research has shown the effectiveness of BiFPN [92–94] and ASPP [95–97] in improving the detection performance of YOLO algorithms.

3. YOLOv5 Algorithm

Among the existing object detection algorithms, the YOLOv5 algorithm has gained wide popularity in various applications due to its fast detection speed, high accuracy, and good flexibility. Based on differences in network depth and width, the proposed YOLOv5 model can be categorized into five types: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. These models demonstrate a progressive increase in number of parameters and level of performance, but with a corresponding decrease in detection speed. Considering both detection performance and speed, this research selected the YOLOv5s model. This model comprises four main components: input, backbone, neck, and head. Figure 1 illustrates the network structure diagram.

3.1. Input

YOLOv5 performs adaptive image scaling and Mosaic data augmentation, as well as optimized anchor box calculations at the input end for image data [38].

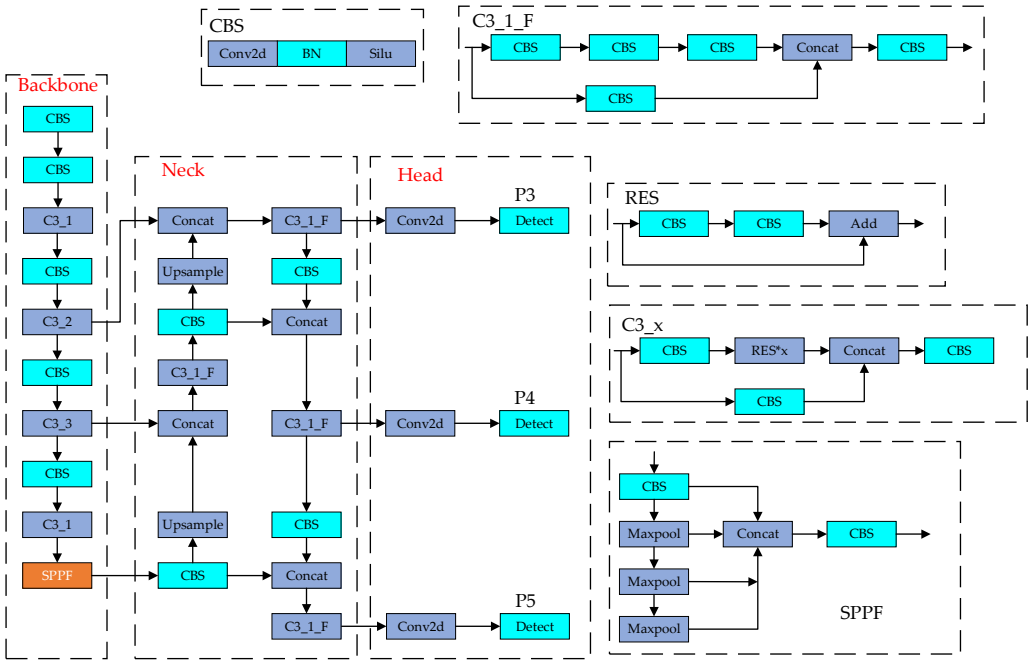

Figure 1. YOLOv5 network structure.

Mosaic is a data augmentation method based on Cutmix [98]. By combining four training images into one image and scaling the resulting image to the standard size before training, Mosaic can effectively improve object detection beyond normal backgrounds. During training, each batch of data contains a large number of images, and data augmentation increases the number of images in each batch fourfold, which reduces the requirement for large quantities when estimating mean and variance. Adaptive image scaling is only performed during the model inference stage. First, the scaling ratio is calculated based on the original image and the input network image size. Then, the scaled image size is determined by multiplying the original image size by the scaling ratio. Finally, the image is scaled to fit the input size of the network. In the YOLOv2–4 algorithms [36–38], prior box dimensions need to be extracted using K-means clustering [99]. To train on different datasets, a separate program is required to obtain the initial anchor boxes to meet specific size requirements. YOLOv5 embeds adaptive anchor box calculation into its code, which automatically calculates the optimal anchor boxes during each training session based on the dataset.

3.2. Backbone

The backbone network primarily extracts feature information from input images. The C3 module and SPPF module are mainly used in the YOLOv5 network. The C3 module reduces model computation and improves inference speed, while the SPPF module extracts multi-scale information from feature maps, which is beneficial for improving model accuracy.

The C3 module consists of three standard convolutional layers and multiple bottleneck modules, the number of which is determined by the parameters specified in the configuration file. The C3 module is the main module for learning residual features. Its structure consists of two branches: one branch uses the specified multiple bottleneck modules stacked and standard convolutional layers, while the other branch only passes through standard convolutional layers. Finally, the two branches are concatenated and passed through standard convolutional layers to output the final feature map.

SPP [41] can fuse feature maps of different scales and sizes by performing fixed-size pooling on feature maps of any scale to obtain a fixed number of features. Then, each pooled feature is concatenated to obtain a fixed-length feature map. The principle of the

SPPF module is similar to SPP, with a slightly different structure. In YOLOv5, SPP uses three scales of features [5,9,13] to fuse with the input feature. The results further improve the scale invariance of input images with different scales and aspect ratios. On the other hand, SPPF only uses a 5×5 pooling kernel. After the input image passes through a standard convolutional layer, it goes through three stacked 5×5 pooling kernels. Each scale feature after pooling is fused with the scale feature after passing through the standard convolutional layer to obtain the final feature map. Compared with the SPP module, the computational complexity of the SPPF module is greatly reduced, and the model speed is improved.

3.3. Neck

The neck network of YOLOv5 consists of a feature pyramid network (FPN) [42] and a path aggregation network (PAN) [43]. FPN always uses the semantic information of high-level features and high-resolution location information of low-level features simultaneously by using a top-down approach to propagate semantic information. On the other hand, PAN uses a bottom-up approach to facilitate the propagation of low-level information to the top level for better localization. The three sizes of feature maps output by the backbone network are aggregated by the neck network to enhance semantic information and localization features, which helps to improve the ability to detect objects of different sizes.

3.4. Head

As the detection component of the object detection model, the head predicts objects of different sizes by processing multi-scale feature maps. The anchor box mechanism at the head extracts prior box scales through clustering and constrains the predicted box positions.

The model outputs three scale tensors, with the first scale having an eight-fold down-sampling compared to the input image, resulting in a smaller receptive field that preserves high-resolution features from the bottom layers and is beneficial for detecting small objects. The second scale has a 16-fold down-sampling, resulting in a moderately sized receptive field that is beneficial for detecting medium-sized objects. The third scale has a 32-fold down-sampling, resulting in a larger receptive field that is beneficial for detecting large objects.

3.5. Loss Function

The loss function ($\mathcal{L}_{total}$) of YOLOv5 consists of three components defined as Equation (1), which covers several necessary loss function modules in object detection such as confidence loss function, class prediction loss function, and bounding box prediction loss function [100].

$$\mathcal{L}_{total} = \mathcal{L}_{obj} + \mathcal{L}_{class} + \mathcal{L}_{bbox} \tag{1}$$

where $\mathcal{L}_{obj}$ represents the target confidence loss of the model, $\mathcal{L}_{class}$ represents the target class prediction loss of the model, and $\mathcal{L}_{bbox}$ represents the bounding box loss of the model.

$$\begin{aligned} \mathcal{L}_{obj} = \quad & \lambda_{obj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{obj} \left[-\hat{C}_i^j \ln\left(C_i^j\right) - \left(1 - \hat{C}_i^j\right) \ln\left(1 - C_i^j\right) \right] \\ & + \lambda_{nobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{nobj} \left[-\hat{C}_i^j \ln\left(C_i^j\right) - \left(1 - \hat{C}_i^j\right) \ln\left(1 - C_i^j\right) \right] \end{aligned} \tag{2}$$

$$\mathcal{L}_{class} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \sum_{c \subset classes} \left[-\hat{p}_i(c) \ln(p_i(c)) - (1 - \hat{p}_i(c)) \ln(1 - p_i(c)) \right] \tag{3}$$

where S represents the grid size, B represents the number of predicted boxes per grid, and I_{ij}^{obj} represents whether the i_{th} predicted box in the j_{th} grid contains an object. If the overlap between the predicted box and the ground truth box exceeds the threshold, I_{ij}^{obj} is set to 1, indicating the presence of an object to be predicted, and it is included in the calculation of

the loss function; otherwise, I_{ij}^{obj} is set to 0. I_{ij}^{nobj} represents whether the i_{th} predicted box in the j_{th} grid contains a background object. If the overlap between the predicted box and the ground truth box is less than the threshold, I_{ij}^{nobj} is set to 1; otherwise, I_{ij}^{nobj} it is set to 0. λ_{obj} and λ_{nobj} are balance coefficients used to adjust the balance between the confidence loss in the presence and absence of objects. C_i^j represents the confidence of the predicted box, $\hat{C}_i^j$ represents the confidence of the ground truth box, $p_i(c)$ represents the predicted probability of the c class when the i_{th} network detects an object, and $\hat{p}_i(c)$ represents the true probability of the c class when the i_{th} network detects an object.

The YOLOv5 bounding box prediction loss function utilizes the CIOU loss function ($\mathcal{L}_{CIOU}$) [101], and its definition is as follows:

$$\mathcal{L}_{CIOU} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1 - IOU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \alpha v \tag{4}$$

$$IOU == \frac{\left|B \cap B^{gt}\right|}{\left|B \cup B^{gt}\right|} \tag{5}$$

where $\rho^2\left(b, b^{gt}\right)$ represents the Euclidean distance between the centers of the predicted box and the ground-truth box; c represents the diagonal distance of the minimum enclosing region that can simultaneously contain the predicted and ground-truth boxes; B,B^{gt}, and α respectively denote the predicted box, the ground-truth box, and the weight coefficient; v is used to measure the similarity of aspect ratios; and the formulas for α and v are given by Equations (6) and (7), respectively:

$$\alpha = \frac{v}{(1 - IOU) + v} \tag{6}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w^p}{h^p}\right)^2 \tag{7}$$

where w^{gt} and h^{gt} respectively represent the width and height of the ground-truth box; and w^p and h^p respectively represent the width and height of the predicted box.

4. Improved YOLOv5s Algorithm

4.1. Shuffle Attention Module

As the hierarchical depth of the network increases, the information extracted from the head of the YOLOv5s network becomes increasingly abstract which will lead to missed or false detection of small objects in the image. In this study, an attention mechanism was incorporated into the YOLOv5s network to address this issue.

The attention mechanism can be mainly divided into spatial attention and channel attention, which are used to capture pixel relationships in space and dependencies between channels, respectively. The combination of these two attention mechanisms, such as in CBAM, can achieve better results but inevitably increases the computational complexity of the model. The SGE attention mechanism module [102] is a classic attention module. Its core idea is to group feature maps, with each group of feature maps representing a semantic feature. By utilizing the similarity between local and global features, the attention mask is generated to guide the spatial distribution of enhanced semantic features. Based on the design concept of the SGE attention mechanism, the shuffle attention (SA) mechanism [103] introduces the channel shuffle operation, which uses both spatial and channel attention mechanisms in parallel, efficiently combining the two. As shown in Figure 2, the SA module first groups the $c \times h \times w$ feature map obtained by convolution, and the grouped feature map serves as the SA unit. Each SA unit is divided into two parts, with the upper part using the channel attention mechanism and the lower part using the spatial attention mechanism. The processed two parts are stacked by channel numbers to achieve information fusion

within the SA unit. Finally, the channel shuffle operation is applied to all SA units to realize information communication between different sub-features and obtain the final output feature map.

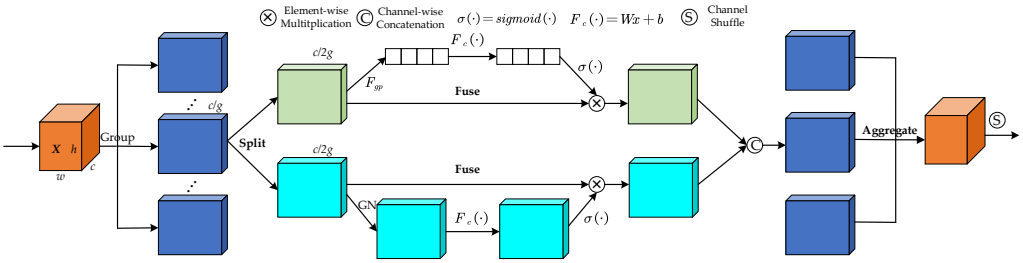

Figure 2. Shuffle attention network structure.

The SA module has the convenient feature of being plug-and-play and has been applied in some networks. However, there is currently no absolute theoretical basis for determining which part of the YOLOv5s network is best in terms of integration. The YOLOv5s network can be divided into four modules: input, backbone, neck, and head. The input module mainly performs preprocessing operations on images; it does not perform feature extraction processing on images. Therefore, in this study, fusion network models that incorporate the SA module into the backbone, neck, and head modules of YOLOv5s were designed and named YOLOv5s-SA-A, YOLOv5s-SA-B, and YOLOv5s-SA-C, respectively.

The SA module was embedded into the backbone structure to form the YOLOv5s-SA-A network. The backbone extracts the feature information from images through a relatively deep convolutional network. As the network layers deepen, the resolution of the feature map decreases. The SA module can be used for spatial attention enhancement and channel attention reconstruction of feature maps at different locations. The C3 module aggregates features at different levels. In this study, the SA module was placed after the C3 module. The network structure is shown in Figure 3A. The SA module was embedded into the neck structure to form the YOLOv5s-SA-B network. The FPN and PAN structures in the neck module can transmit semantic information from top to bottom and positional information from bottom to top, thereby enhancing the aggregation of semantic information and positioning features. This module uses four Concat operations to fuse deep and shallow information. Therefore, the SA module was placed after the Concat operation to enhance the spatial attention and channel reconstruction of the fused feature map. The network structure is shown in Figure 3B. The SA module was embedded into the head structure to form the YOLOv5s-SA-C network. The YOLOv5s network predicts targets using three feature maps of different scales. Large targets are predicted on small feature maps, while small targets are predicted on large feature maps. In this study, the SA module was embedded before the prediction head to enhance the spatial attention and channel reconstruction of each feature map. The network structure is shown in Figure 3C.

4.2. DD-Head Module

Traditional YOLO algorithms use the coupled head, which utilizes the same convolutional layer for both classification and regression tasks at the head of the network. However, classification and regression tasks have different focuses. Classification is more concerned with the texture of each sample, while regression is more focused on the edge features of object images. Studies [104,105] have pointed out that there is a conflict between classification and regression tasks in object detection, and using a coupled head for both tasks may lower the model's performance.

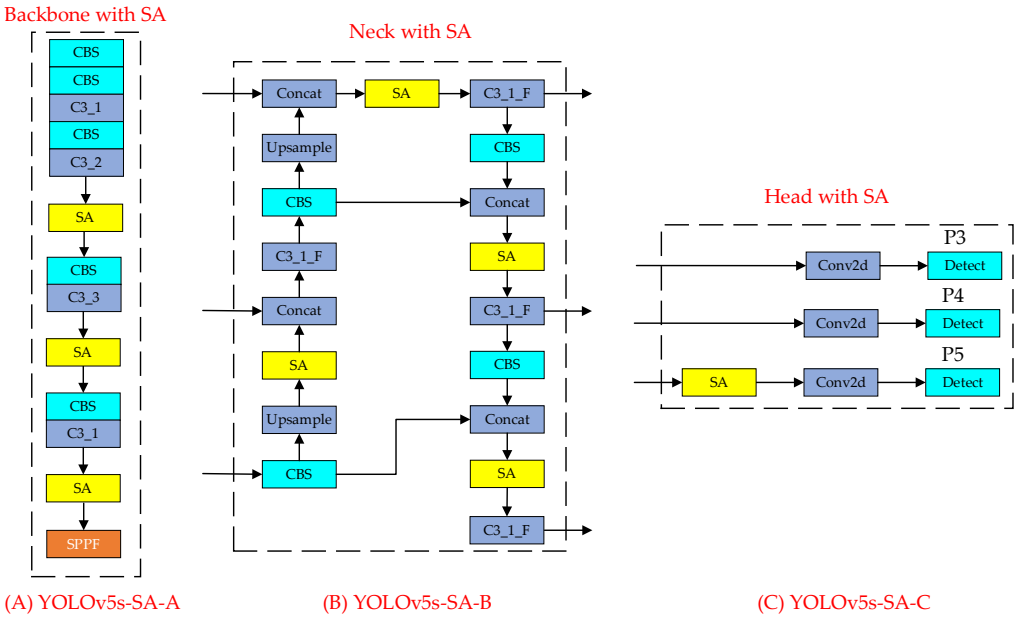

Figure 3. YOLOv5s network structure integrating shuffle attention module.

The YOLOX algorithm [106] is the first algorithm to apply the decoupled head module, achieving more significant results than the coupled head. The network structure of the decoupled head is shown in Figure 4. For the input feature map, the decoupled head first uses a 1×1 convolution to reduce its dimensionality, mapping the feature maps of P3, P4, and P5 with different dimensions of the feature fusion network output into feature maps with a unified number of channels. Then, two parallel channels are used to perform object regression and target box coordinate regression tasks. To reduce the complexity of the decoupled head and improve model convergence speed, each channel uses two 3×3 convolutions. *Cls.*, *Reg.*, and *Obj.* output values can be obtained through processing, where *Cls.* represents the category of the target box, *Reg.* represents the position information of the target box, and *Obj.* represents whether each feature point contains an object. The final prediction information is obtained by fusing the three output values. In summary, the decoupled head improves model performance by separately addressing classification and regression tasks.

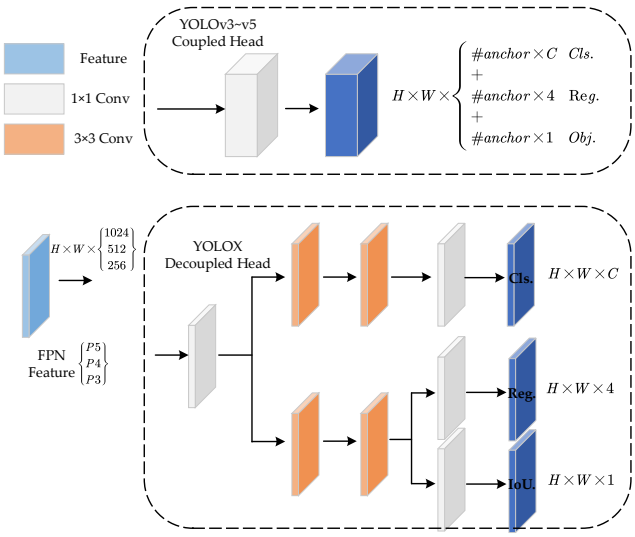

Figure 4. Decoupled head network structure.

Although introducing a decoupled head can effectively improve the detection performance of a network, it undoubtedly increases the model's parameter count and decreases the detection speed. In this study, the 3 × 3 convolution in each branch of the decoupled head network was replaced with a 3 × 3 depthwise convolution [107], reducing the parameter count. The network architecture is shown in Figure 5, which is named the DD-head. The original coupled head in the YOLOv5s model was replaced with the DD-head to mitigate the negative impact of the classification and regression task conflict.

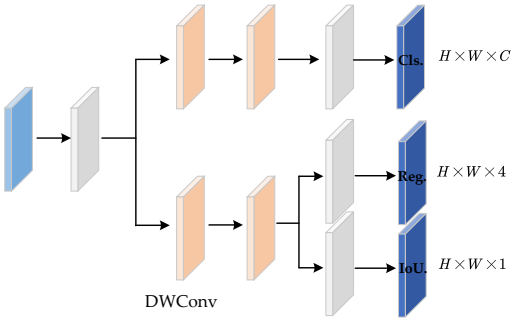

Figure 5. Depthwise-decoupled head network structure.

4.3. Content-Aware Reassembly of the Features Module

The YOLOv5s model utilizes the FPN module to achieve top-down semantic information transfer and multi-scale object detection through feature fusion. The multi-scale feature fusion is achieved by nearest-neighbor interpolation up-sampling to unify the feature map size. However, this up-sampling operation presents two limitations: (1) the interpolation up-sampling operation only considers the spatial information of the feature map and ignores its semantic information, resulting in simultaneous up-sampling of target and noise positions; (2) the receptive field of the interpolation up-sampling operation is usually small, leading to insufficient use of global feature information. Another adaptive up-sampling operation is the deconvolution operation; however, it also has two limitations: (1) the deconvolution operator uses the same convolution kernel across the entire feature map, regardless of the underlying information, which limits its ability to respond to local changes; (2) the parameter volume of the deconvolution operation is large, which reduces the detection speed of the network. To address the problem of low semantic correlation in up-sampling in object detection models, this study adopts the Content-Aware Reassembly of Features (CARAFE) module [108] to replace the nearest-neighbor interpolation up-sampling module. The CARAFE module recombines feature points with similar semantic information in a content-aware manner and aggregates features in a larger receptive field to achieve the up-sampling operation.

The up-sampling process of the CARAFE module mainly consists of two steps: up-sampling kernel prediction and content-aware reassembly. Firstly, channel compression is performed by the up-sampling kernel prediction module to reduce the number of input feature channels. Then, the compressed feature map is encoded with content and the reassembly kernel is predicted according to the content of each target location. Finally, the content-aware reassembly module performs a dot product between the reassembly kernel and the corresponding region of the original feature map to complete the up-sampling process.

4.4. GSConv Module

Although the introduction of the decoupled head and SPPCSPC modules can improve the detection performance of the YOLOv5s network model, these modules increase the parameter count of the model, which is unfavorable for creating lightweight networks. To design lightweight networks, deep separable convolution (DSC) modules are typically used instead of conventional convolutional modules. The advantage of DSC modules is their efficient computational capability, as their parameter count and computational workload are

approximately one-third of those of traditional convolutional modules. However, during the feature extraction process, the channel information of the input image is separated in the calculation process, which can result in lower feature extraction and fusion capabilities compared to standard convolutional modules. To effectively utilize the computational capability of DSC and ensure that its detection accuracy reaches the level of standard convolution (SC), the GSconv [109] module is proposed based on SC, DSC, and shuffle modules. The network structure is shown in Figure 6. Firstly, the feature map with C_1 channels is split into two parts, where half of the feature map is used for deep separable convolution and the remaining part is used for standard convolution. Then, the two-channel feature maps are combined for feature concatenation. Shuffle is a channel mixing technique that allows information from the SC module to completely mix with the DSC output by transmitting its feature information across various channels, thus achieving channel information interaction.

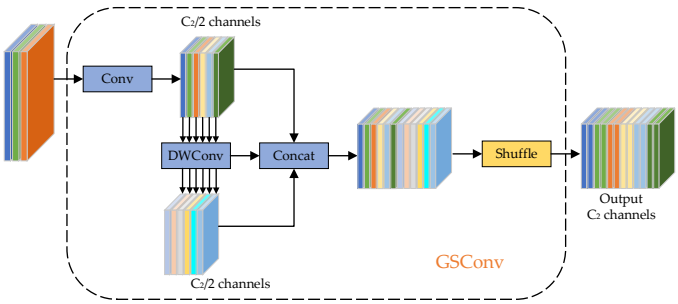

Figure 6. GSConv network structure.

During the convolution process, the spatial information of the feature map gradually shifts to channels, where the number of channels increases when the width and height of the feature map decrease, resulting in stronger semantic information. However, each spatial compression and channel expansion of the feature map can lead to partial loss of semantic information, which affects the accuracy of object detection. The SC module largely preserves the hidden connections between each channel, which can reduce information loss to some extent, but with high time complexity. In contrast, the DSC module cuts off these hidden connections, resulting in the complete separation of channel information during the calculation process. The GSconv module retains as many connections as possible while maintaining lower time complexity, thereby reducing information loss and achieving faster operations and thus unifying SC and DSC.

Based on the GSConv module, the network structure of the GS bottleneck and VoV-GSCSP module is shown in Figure 7. Compared with the bottleneck module in the original YOLOv5s network, the GS bottleneck replaces the two 1×1 convolutions in the bottleneck module with the GSConv module and adds new skip connections. Therefore, the two branches of the GS bottleneck perform separate convolutions without weight sharing, propagating channel information through different network paths by dividing the number of channels. As a result, the information propagated by the GS bottleneck shows greater correlation and diversity, resulting in more accurate information and reduced computational workload. The VoV-GSCSP module is designed by using the GS bottleneck instead of the bottleneck in the C3 module. In the VoV-GSCSP module, the input feature map is also divided into two parts based on channel numbers. The first part is processed by a convolutional module and features are extracted by stacking the GS bottleneck, while the other part serves as the residual connection and is convolved by a convolutional module. The two feature maps are then concatenated based on channel numbers and passed through a convolutional module for output. The VoV-GSCSP module inherits the advantages of both the GSConv module and the GS bottleneck. With the new skip connection branch, the VoV-GSCSP module has a stronger nonlinear representation, effectively addressing the problem of gradient vanishing. At the same time, the split-channel method of VoV-GSCSP achieves rich gradient combinations, solving the problem of redundant gradient

information and improving learning ability [109]. Experimental results have shown that the VoV-GSCSP module not only reduces computational workload but also improves model accuracy [110,111]. In this study, the Conv module in the VoV-GSCSP module was replaced with the GSConv module to further reduce model parameters. The GSConv module and the improved VoV-GSCSP module were embedded into the neck structure of the model, so as to maintain detection accuracy while reducing parameter count.

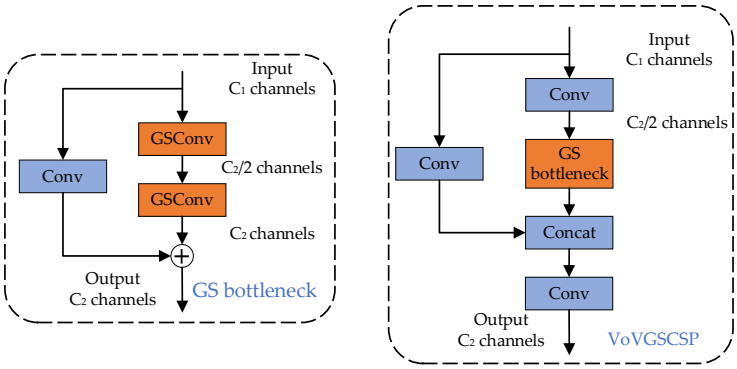

Figure 7. GS bottleneck and VoV-GSCSP network structure.

4.5. SPPCSPG Module

The SPPCSPC module [112] is built on the basis of the SPP module and the CSP structure, as shown in Figure 8. The module first divides the features extracted by the C3 module of the backbone into two parts: the SPP and conventional convolution operations. The SPP structure consists of four branches, corresponding to max-pooling operations with pool kernel sizes of 1, 5, 9, and 13. These four different pool kernels allow the SPPCSPC structure to handle objects with four different receptive fields, better distinguishing small and large targets. Finally, the SPP operation and conventional convolution operation are merged together by Concat to achieve faster speed and higher accuracy.

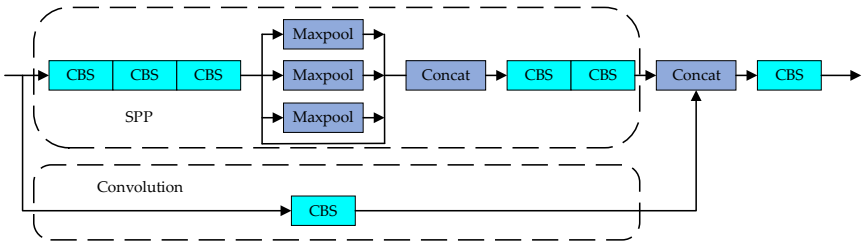

Figure 8. Spatial pyramid pooling cross-stage partial conv network structure.

Although the SPPCSPC module can improve the detection performance of the model to a certain extent, it also increases the model's parameter count. Therefore, in this study, a lightweight SPPCSPG module was proposed based on the design principles of the GS bottleneck and the GSConv module. The SPPCSPG module was integrated into the backbone structure to optimize the YOLOv5s network model.

Incorporating the improvements to the YOLOv5s backbone, this study replaced the SPPF module with the SPPCSPG module to enhance detection accuracy. In the neck structure, all Conv modules were replaced with GSConv modules and the improved VoV-GSCSP module was introduced to reduce the parameters and computation brought about by feature pyramid structure upgrades. To address the issue of limited semantic information and receptive fields caused by nearest-neighbor interpolation up-sampling operations in the original network model, this study adopted the CARAFE module to replace the nearest-neighbor interpolation up-sampling module, which reorganizes feature points with similar semantic information in a content-aware way and aggregates features in a larger receptive field to perform up-sampling operations. To address the problem of

inaccurate target localization and weak feature expression capability in the original network model, this study introduced the SA attention module in the head module. Finally, to improve the performance metrics of the model, the DD-head was used in the detection layer of the YOLOv5s model to accomplish classification and regression tasks. The improved network model structure is shown in Figure 9.

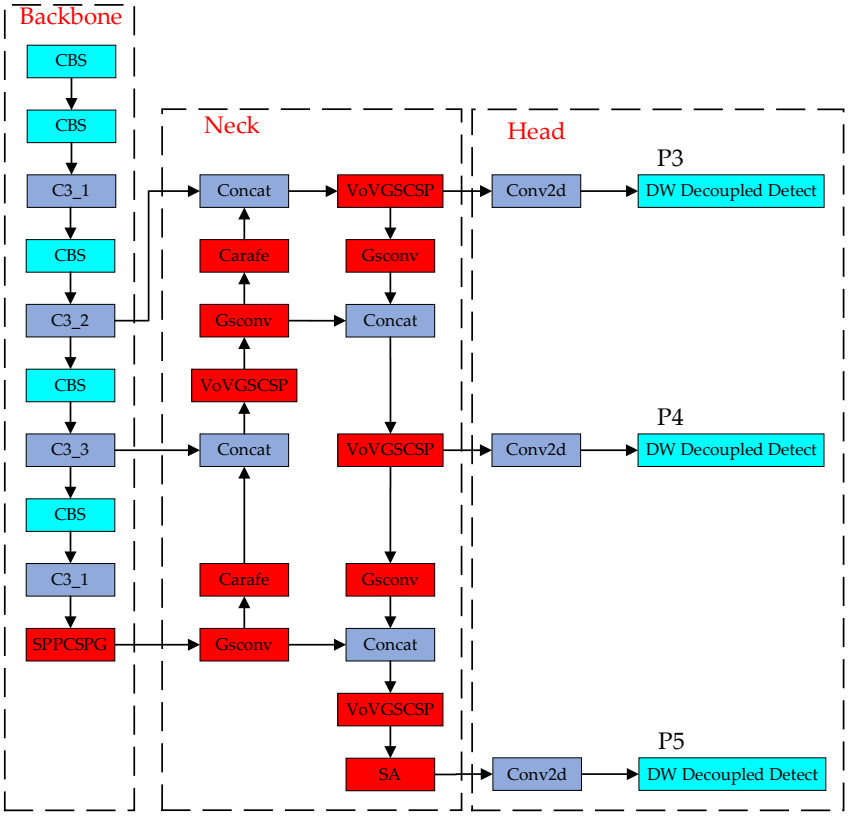

Figure 9. Network structure of improved YOLOv5s model.

5. Experimental Results and Analysis

5.1. Experimental Platform and Dataset

The dataset used in the experiment includes two remote sensing datasets, the RSOD dataset [113,114] and DIOR dataset [115], as well as two general target detection datasets, the PASCAL VOC dataset [116,117] and MS-COCO dataset [118].

The RSOD dataset includes four types of detection object categories, aircraft, oil tank, playground, and overpass, with a total of 976 images. The aircraft category includes 446 images, with a total of 4993 targets. The oil tank category includes 165 images, with a total of 1586 targets. The playground category includes 189 images, with a total of 191 targets. The overpass category contains 176 images and a total of 180 targets. The partition ratio between the training set to the test set is 4 to 1.

The DIOR dataset is a large-scale benchmark dataset for object detection in remote sensing images. The dataset includes 23,463 images of different seasons and weather patterns, with a total of 190,288 targets. The unified image size is 800 × 800, with a resolution of 0.5 m to 30 m. DIOR datasets include 20 categories: airplane (AL), airport (AT), baseball field (BF), basketball court (BC), bridge (B), chimney (C), dam (D), expressway service area (ESA), expressway toll station (ETS), golf course (GC), ground track field (GTF), harbor (HB), overpass (O), ship (S), stadium (SD), storage tank (ST), tennis court (TC), train station (TS), vehicle (V), and windmill (W). According to the original settings in the DIOR dataset, the number of images in the training, validation, and testing sets is 5863, 5862, and 11,738, respectively. This study combines the training set and validation set as the training set.

The PASCAL VOC dataset includes the PASCAL VOC 2007 and 2012 datasets, which can be used for tasks such as image classification, object detection, semantic segmentation, and motion detection. The PASCAL VOC dataset includes a total of 20 common objects in daily life. In this study, the training and validation sets of the PASCAL VOC 2007 and VOC 2012 datasets were used as the model's training set, while the testing set of VOC 2007 was used as the model's testing set. The MS-COO dataset is currently the most challenging target detection dataset which includes a total of 80 types of detection objects. The MS-COO dataset includes more small objects (with an area smaller than 1% of the image) and more dense localization objects than the PASCAL VOC dataset.

The experiments were conducted on a system with Ubuntu 18.04, CUDA 11.1, and a GeForce RTX A5000 graphics card. The network development framework used was Pytorch 1.9, and the integrated development environment was Pycharm. The training was uniformly set to 300 epochs, with a batch size of 64.

5.2. Evaluation Metrics

In this study, precision, mean average precision (*mAP*), and detection frames per second (FPS) were used as performance evaluation metrics for the object detection method. Precision (*P*) and recall (*R*) were calculated using Equations (8) and (9), respectively:

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

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

where *TP* represents the number of positive samples that are correctly identified as positive, *FP* represents the number of negative samples that are incorrectly identified as positive, and *FN* represents the number of positive samples that are incorrectly identified as negative. By selecting different precision and recall values, the precision-recall (*PR*) curve can be drawn, and the area under the *PR* curve is defined as the *AP*. The mean *AP* (*mAP*) is calculated by taking the mean of the *AP* for all detection categories. The calculation of the performance evaluation metrics *AP* and *mAP* is shown in Equations (10) and (11), respectively:

$$AP = \int_0^1 p(r)dr \tag{10}$$

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

where $p(r)$ represents the precision value at a certain recall value r, and n represents the number of detection categories.

5.3. Experimental Results and Analysis

The experimental results are divided into seven sections based on two different datasets: the comparison experiment of SA module embedding, the effect experiment of the DD-head module, the effect experiment of the SPPCSPG module, the RSOD dataset experiment, the DIOR dataset experiment, the PASCAL VOC dataset experiment, and the MS COCO dataset experiment.

5.3.1. Performance Evaluation of the SA Module Embedded Model

To explore the best SA module embedding model and investigate the detection performance changes brought by embedding the SA module in different structures of the YOLOv5s network, three proposed models (YOLOv5s-SA-A, YOLOv5s-SA-B, and YOLOv5s-SA-C) were evaluated using the RSOD dataset to achieve better optimization design of the network. The detection performance of the original YOLOv5s and improved models were compared, and the experimental results are shown in Table 1.

Table 1. Performance evaluation of shuffle attention module embedded model.

Method	Data Size	Param.	GFLOPs	Precision (%)	Recall (%)	mAP@0.5 (%)	FPS
YOLOv5s	640 × 640	7.02 M	15.8	0.942	0.932	0.950	119.0
YOLOv5s-SA-A	640 × 640	7.02 M	15.8	0.930	0.920	0.948	105.8
YOLOv5s-SA-B	640 × 640	7.02 M	15.8	0.937	0.935	0.948	112.3
YOLOv5s-SA-C	640 × 640	7.02 M	15.8	0.939	0.930	0.952	113.9

From Table 1, it can be seen that not all networks with integrated SA modules can improve detection performance. YOLOv5s-SA-A had significantly decreased precision, recall, mAP, and FPS compared to the original YOLOv5s network. YOLOv5s-SA-B showed improved recall, but the other three indicators were decreased compared to the original network. YOLOv5s-SA-C had increased mAP, but the other three indicators were decreased compared to the original network. The reason for the different experimental results when embedding attention mechanisms at different positions in the network is that the feature maps extracted by the backbone have rich semantic features, while the feature maps extracted by the neck and head have larger receptive fields, which play a crucial role in improving object detection performance. In the backbone module, the feature maps retain the shallow texture and contour information of the targets, with poor semantic information; thus, embedding attention mechanisms cannot effectively learn semantic information. The YOLOv5s-SA-C algorithm is superior to the YOLOv5s-SA-B algorithm in both detection accuracy and speed. Therefore, considering the principle of balancing accuracy and speed, the YOLOv5s-SA-C algorithm was finally chosen as the model for embedding SA modules.

5.3.2. Effect Experiment of the DD-Head Module

Experiments were conducted on the proposed DD-head module, decoupled head module, and coupled head module in the RSOD dataset to investigate the impact of the constructed DD-head module on the detection accuracy and speed of the model.

Table 2 shows that both the decoupled head and the proposed DD-head modules can improve the mean average precision accuracy of the model, with an increase of 0.6% and 0.5%, respectively. Although the DD-head module has a 0.1% lower mean average precision accuracy compared to the decoupled head module, it has a 7.05 M lower parameter count and an 11.9 increase in FPS value compared to that of the decoupled head module.

Table 2. Performance evaluation of depthwise-decoupled head module embedded model.

Method	Data Size	Param.	GFLOPs	Precision (%)	Recall (%)	mAP@0.5 (%)	FPS
YOLOv5s	640 × 640	7.02 M	15.8	0.942	0.932	0.950	119.0
YOLOv5s + Decoupled head	640 × 640	14.33 M	56.2	0.937	0.969	0.956	94.9
YOLOv5s + DD-head	640 × 640	7.28 M	16.7	0.960	0.929	0.955	106.8

5.3.3. Effect Experiment of the SPPCSPG Module

Experiments were conducted on the proposed SPPCSPG module, SPPCSPC module, and SPPF module in the RSOD dataset to study the impact of the constructed SPPCSPG module on the detection accuracy and speed of the model.

Table 3 indicates that both the SPPCSPC module and the proposed SPPCSPG module can improve the mean average precision accuracy of the model, with an increase of 0.3% and 0.5%, respectively. Moreover, the SPPCSPG module has a 0.2% higher mean average precision accuracy than the SPPCSPC module and has 3.5 M fewer parameters. Therefore, based on the evaluation of parameter count and detection accuracy, the proposed SPPCSPG module outperforms the SPPCSPC module.

Table 3. Performance evaluation of spatial pyramid pooling cross-stage partial with GSConv module embedded model.

Method	Data Size	Param.	GFLOPs	Precision (%)	Recall (%)	mAP@0.5 (%)	FPS
YOLOv5s	640 × 640	7.02 M	15.8	0.942	0.932	0.950	119.0
YOLOv5s + SPPCSPC	640 × 640	13.45 M	20.9	0.947	0.931	0.953	117.0
YOLOv5s + SPPCSPG	640 × 640	9.95 M	18.1	0.942	0.954	0.955	113.1

5.3.4. Performance Comparison in the RSOD Dataset

The detection performance of the original YOLOv5s model and the constructed model was compared on the RSOD dataset to verify the effectiveness of the constructed model. The results were shown in Table 4. The mean average precision accuracy and FPS of the model constructed in this article with the original YOLOv5s model were compared on the RSOD dataset, and the detection accuracy of each category was compared. It can be seen that the mean average precision accuracy of the model constructed in this article is 96.4%, which is an improvement of 1.4% compared to the original model, and the mean average precision accuracy on categories oil tank and playground were improved by 0.1% and 5.4%, respectively.

Table 4. RSOD test detection results.

Method	Param.	GFLOPs	mAP@0.5 (%)	Aircraft	Oil Tank	Playground	Overpass	FPS
YOLOv5s	7.02 M	15.8	0.950	0.983	0.986	0.839	0.994	119.0
Ours	9.67 M	18.4	0.964	0.981	0.987	0.893	0.992	88.8

In order to further visually demonstrate the effectiveness of the constructed models, the detection results of each model were shown in Figure 10.

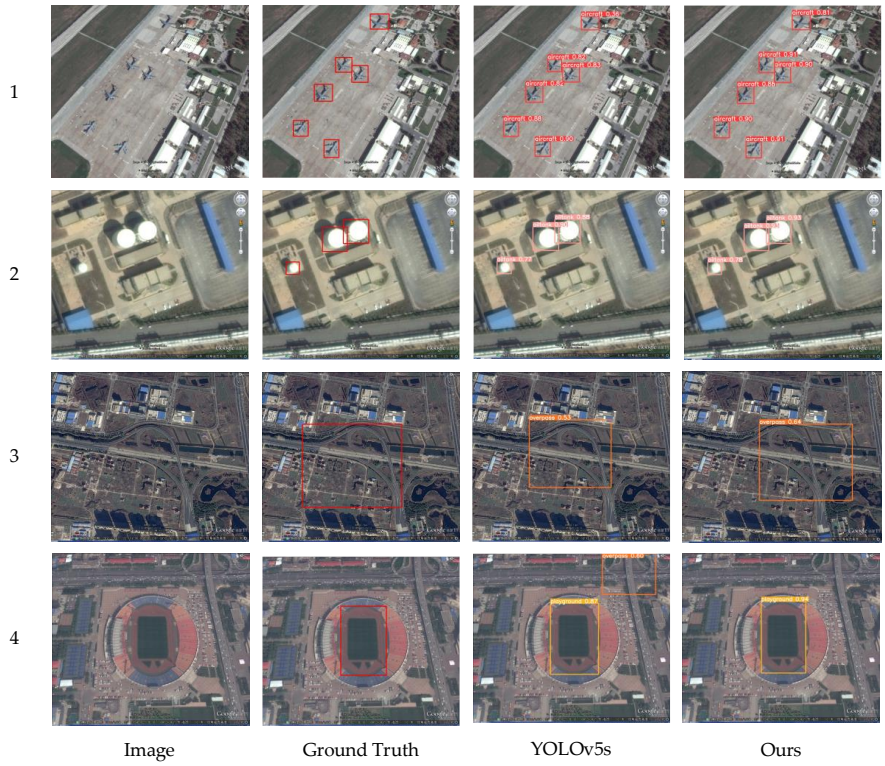

Figure 10. Visualization results of each model on the RSOD test dataset. The model constructed in this article has a lower false alarm rate (row **4**) and a higher recall rate (rows **1**, **2**, and **3**).

As shown in Figure 10, from rows 1 to 4, it could be seen that the detection boxes of the model constructed in this article were closer to the actual detection boxes of targets, with a higher recall rate. From row 4, it can be seen that the model constructed in this article had a lower false alarm rate on the overpass. Therefore, the effectiveness of the model constructed in this article is superior to the original model.

5.3.5. Performance Comparison in the DIOR Dataset

In order to further validate and evaluate the effectiveness of the improved YOLOv5s in improving the detection accuracy of the model, experimental comparisons were conducted on the DIOR dataset with other methods specified in the literature. The experimental results are shown in Table 5, with bold values indicating the optimal results in each column.

Table 5. DIOR test detection results.

Method	mAP	AL	AT	BF	BC	B	C	D	ESA	ETS	GC
HawkNet [119]	72.0	65.7	84.2	76.1	87.4	45.3	79.0	64.5	82.8	72.4	82.5
CANet [120]	74.3	70.3	82.4	72.0	87.8	55.7	79.9	67.7	83.5	77.2	77.3
Yao et al. [121]	75.8	91.0	74.5	93.3	83.2	47.4	**91.9**	63.3	68.0	61.4	80.0
MFPNet [122]	71.2	76.6	83.4	80.6	82.1	44.3	75.6	68.5	85.9	63.9	77.3
FSoD-Net [123]	71.8	88.9	66.9	86.8	90.2	45.5	79.6	48.2	86.9	75.5	67.0
ASDN [124]	66.9	63.9	73.8	71.8	81	46.3	73.4	56.3	73.4	66.2	74.7
MSFC-Net [125]	70.1	85.8	76.2	74.4	90.1	44.2	78.1	55.5	60.9	59.5	76.9
Xue et al. [126]	80.5	**95.2**	84.2	**94.8**	85.2	54.0	90.5	71.0	75.3	70.7	82.0
DFPN-YOLO [127]	69.33	80.2	76.8	72.7	89.1	43.4	76.9	72.3	59.8	56.4	74.3
AC-YOLO [128]	77.1	93.1	80.9	79.9	84.4	**76.0**	81.7	77.1	67.6	70.0	66.7
SCRDet++ [129]	75.1	71.9	85.0	79.5	88.9	52.3	79.1	77.6	89.5	77.8	84.2
MSSDet [130]	76.9	70.7	88.6	81.8	90.4	56.5	82.5	73.0	90.1	78.6	86.6
Gao et al. [131]	72.5	78.1	83.9	73.0	89.0	48.2	79.4	65.6	63.9	61.9	80.6
MDCT [132]	80.5	92.5	85.0	93.5	84.7	53.7	90.2	74.3	79.9	68.2	68.6
YOLOv5s	80.4	87.2	86.9	86.2	**92.3**	55.5	83.0	72.6	91.1	**83.0**	81.6
Ours	**81.6**	87.9	**91.1**	84.9	91.7	55.8	80.7	**78.9**	**92.8**	82.6	**86.6**
Method	**mAP**	**GTF**	**HB**	**O**	**S**	**SD**	**ST**	**TC**	**TS**	**V**	**W**
HawkNet [119]	72.0	74.7	50.2	59.6	89.7	66.0	70.8	87.2	61.4	52.8	88.2
CANet [120]	74.3	83.6	56.0	63.6	81.0	79.8	70.8	88.2	67.6	51.2	89.6
Yao et al. [121]	75.8	82.8	57.4	65.8	80.0	92.5	81.1	88.7	63.0	73.0	78.1
MFPNet [122]	71.2	77.2	62.1	58.8	77.2	76.8	60.3	86.4	64.5	41.5	80.2
FSoD-Net [123]	71.8	77.3	53.6	59.7	78.3	69.9	75.0	91.4	52.3	52.0	90.6
ASDN [124]	66.9	75.2	51.1	58.4	76.2	67.4	60.2	81.4	58.7	45.8	83.1
MSFC-Net [125]	70.1	73.7	49.6	57.2	89.6	69.2	76.5	86.7	51.8	55.2	84.3
Xue et al. [126]	80.5	82.1	70.6	67.3	**95.0**	**94.3**	**83.8**	91.6	61.2	79.8	81.8
DFPN-YOLO [127]	69.33	71.6	63.1	58.7	81.5	40.1	74.2	85.8	73.6	49.7	86.5
AC-YOLO [128]	77.1	75.7	**75.5**	76.7	87.0	65.8	70.1	88.7	63.5	**81.2**	80.5
SCRDet++ [129]	75.1	83.1	64.2	65.6	71.3	76.5	64.5	88.0	70.9	47.1	85.1
MSSDet [130]	76.9	85.6	63.5	66.5	82.5	82.0	63.3	88.7	71.7	46.7	89.2
Gao et al. [131]	72.5	76.6	63.5	61.6	89.6	68.7	76.4	87.0	66.4	57.0	78.7
MDCT [132]	80.5	**92.9**	68.4	**83.8**	92.9	77.4	83.0	92.8	64.7	77.4	83.0
YOLOv5s	80.4	86.4	66.5	67.3	91.8	81.0	80.4	**93.2**	69.7	60.3	**92.2**
Ours	**81.6**	86.4	68.7	67.3	91.7	81.5	80.3	93.0	**77.1**	60.9	91.4

Note: In the table, 20 categories are divided into 2 rows, with 10 detection results for each row. The boldface values represent the maximum value in the column.

From the experimental results in Table 6, it can be seen that compared with HawkNet [119], CANet [120], MFPNet [122], FSoD-Net [123], ASDN [124], MSFC-Net [125], DFPN-YOLO [127], AC-YOLO [128], SCRNet++ [129], MSSNet [130], MDCT [132], and YOLOv5s, the improved algorithm proposed in this paper based on YOLOv5s significantly improves detection accuracy. On the DIOR dataset, the mean average precision accuracy of the improved YOLOv5s network is 1.2% higher than that of the original YOLOv5s network, indicating that the network constructed in this article not only outperforms the original YOLOv5s

network on the RSOD dataset but also achieves better performance than the original network on the more complex large-scale remote sensing dataset, DIOR.

Table 6. Object detection results on PASCAL VOC2007 test dataset.

Method	Backbone	mAP@0.5 (%)	FPS	GPU
Faster R-CNN [27]	VGGNet	73.2	7	Titan X
SSD 300 [35]	VGGNet	74.1	46	Titan X
ASSD 300 [133]	VGGNet	79.1	39.6	GTX 1080Ti
MFFAMM 300 [134]	VGG16	80.7	26	-
Zhe et al. 300 [135]	VGG16	80.1	42.2	RTX 2080Ti
FESSD 300 [136]	ResNet-50	82.2	41.3	RTX 3090
YOLOv3 320 [37]	Darknet53	74.5	45.5	Titan X
GC-YOLOv3 320 [137]	Darknet53	81.3	39	GTX 1080Ti
DSP-YOLO 416 [138]	Darknet53	82.2	56	Titan Xp
Zhang et al. 416 [139]	MobileNetv2	81.67	44.18	RTX 2080Ti
He et al. 416 [140]	ECA-CSPNet	78.6	94	RTX 2080Ti
SSD 512 [35]	VGGNet	76.8	19.0	Titan X
ASSD 512 [133]	VGGNet	81.0	20.8	GTX 1080Ti
PDS-Net 512 [141]	CSPDarknet-53	84.9	32.2	RTX 2070
SLMS-SSD 512 [142]	VGG16	81.2	17.4	RTX 2080Ti
YOLOv3 544 [37]	Darknet53	78.6	40	Titan X
GC-YOLOv3 544 [137]	Darknet53	83.7	31	GTX 1080Ti
RON384++ [143]	VGG16	77.6	-	Titan X
STDN 513 [144]	DenseNet169	80.9	28.6	Titan Xp
Zhong et al. [145]	BottleneckCSP	84.3	85.2	RTX 2080Ti
YOLO-T 640 [146]	CSPDarknet-53	85.2	65.7	RTX 3090
BFBG-YOLO [147]	CSPDarknet-53	80.3	99.0	RTX 3090
SL-YOLO [148]	ShuffleNet v2	81.2	17.8	Tesla P40
YOLOv5s 640	CSPDarknet-53	83.7	**143**	RTX A5000
Ours 640	CSPDarknet-53	**85.1**	90.2	RTX A5000

The boldface values represent the maximum value in the column.

In order to further visually demonstrate the effectiveness of the constructed models, the detection results of each model were shown in Figure 11.

As shown in Figure 11, from rows 1, 2, 5 and 6, it could be seen that the detection boxes of the model constructed in this article were closer to the actual detection boxes of targets, with a higher recall rate. From rows 1, 3, 4, and 6, it can be seen that the model constructed in this article had a lower false alarm rate. Therefore, the effectiveness of the model constructed in this article is superior to the original model.

5.3.6. Performance Comparison in the PASCAL VOC Dataset

To further validate and evaluate the effectiveness of the proposed algorithm to YOLOv5s in improving detection accuracy, the proposed algorithm was compared with several advanced object detection algorithms that have emerged in recent years. The training set and test set used in the experiments were consistent with those used in the study. The experimental results are shown in Table 6, where the bold values indicate the best results in each column.

From Table 6, it can be seen that the algorithm proposed in this article meets the real-time requirements, with a mean average precision accuracy of 85.1%, which is 1.4% higher than the original YOLOv5s algorithm. Compared to one-stage target detection algorithms, such as the SSD series algorithm and its improved algorithm, it has advantages in terms of precision and detection speed. Compared to two-stage target detection algorithms, such as the Fast R-CNN algorithm, it has a much higher detection accuracy and detection rate.

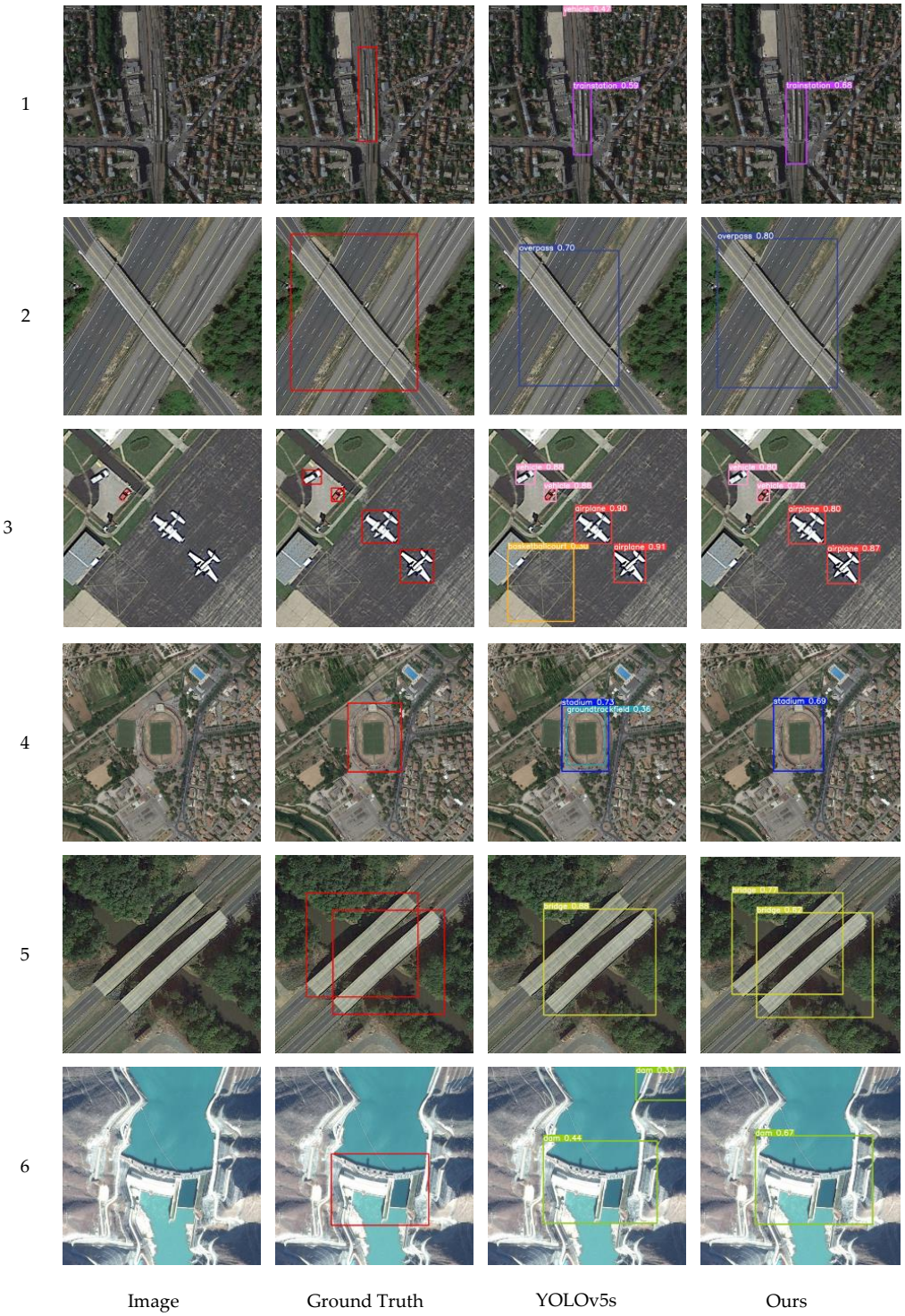

Figure 11. Visualization results of each model on the DIOR test dataset. The model constructed in this article has a lower false alarm rate (rows **1**, **3**, **4** and **6**) and a higher recall (rows **1**, **2**, **5** and **6**).

To further verify whether the algorithm can effectively improve the accuracy of small target detection, Table 7 compares the accuracy of the improved YOLOv5s and other advanced target detection algorithms in the 20 categories in the PASCAL VOC2007 test set. The results show that compared to the original YOLOv5s algorithm, the improved YOLOv5s algorithm improves the detection accuracy of the model in almost every category, especially for small target categories such as birds.

Table 7. PASCAL VOC2007 test detection results.

Method	mAP	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow
Faster R-CNN [27]	73.2	76.5	79	70.9	65.5	52.1	83.1	84.7	86.4	52	81.9
SSD 300 [35]	74.1	74.6	80.2	72.2	66.2	47.1	82.9	83.4	86.1	54.4	78.5
ASSD 300 [133]	79.1	85.4	84.1	78.7	71.8	54.0	86.2	85.3	89.5	60.4	87.4
FESSD 300 [136]	82.2	89.4	86.2	**84.3**	**78.2**	57.8	91.6	91.5	**91.7**	62.2	**90.4**
Zhe et al. 300 [135]	80.1	84.6	87.6	80.1	73.0	50.4	89.3	88.3	90.9	60.2	87.8
He et al. 416 [140]	78.6	86.1	86.2	76.5	66.5	66.4	86.6	91.3	80.7	64.3	84.4
Zhang et al. 416 [139]	81.6	88.5	87.5	83.1	75.2	67.1	85.3	90.2	88.9	60.9	89.7
DSP-YOLO 416 [138]	82.2	88.5	89.5	79.1	74.0	68.7	89.7	90.6	89.9	66.7	84.4
SSD 512 [35]	76.8	82.4	84.7	78.4	73.8	53.2	86.2	87.5	86.0	57.8	83.1
ASSD 512 [133]	81.0	86.8	85.2	84.1	75.2	60.5	88.3	88.4	89.3	63.5	87.6
PDS-Net 512 [141]	84.2	**93.3**	**98.0**	80.2	73.8	70.2	90.9	**96.3**	87.1	65.0	87.3
SLMS-SSD 512 [142]	81.2	88.5	87.1	83.2	76.4	59.2	88.3	88.4	89.0	66.6	86.9
STDN513 [144]	80.9	86.1	89.3	79.5	74.3	61.9	88.5	88.3	89.4	67.4	86.5
DSP-YOLO 608 [138]	83.1	91.0	90.7	81.8	75.6	73.8	91.3	92.7	91.2	66.9	86.9
RON384++ [143]	77.6	86. 0	82.5	76.9	69.1	59.2	86.2	85.5	87.2	59.9	81.4
SFGNet [149]	81.2	82.2	83.9	80.3	71.5	78.2	89.6	86.9	90.0	65.7	87.9
SL-YOLO [148]	81.2	86.4	85.7	77.9	75.5	72.5	85.4	87.8	86.2	**85.9**	72.1
YOLOv5s 640	83.7	91.6	91.9	81.1	75.0	**78.5**	91.2	92.9	87.3	67.4	88.0
Ours 640	**85.1**	92.2	92.0	82.9	74.4	78.1	**92.6**	93.7	91.1	68.8	89.3
Method	**mAP**	**Table**	**Dog**	**Horse**	**Mbike**	**Person**	**Plant**	**Sheep**	**Sofa**	**Train**	**Tv**
Faster R-CNN [27]	73.2	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83	72.6
SSD 300 [35]	74.1	73.9	84.4	84.5	82.4	76.1	48.6	74.3	75.0	84.3	74.0
ASSD 300 [133]	79.1	77.1	87.4	86.8	84.8	79.5	57.8	81.5	80.1	87.4	76.9
FESSD 300 [136]	82.2	74.4	89.4	90.5	87.7	83.7	52.4	88.6	**81.6**	**91.0**	81.7
Zhe et al. 300 [135]	80.1	**81.4**	87.1	89.1	87.9	82.1	54.6	80.4	80.5	89.2	78.1
He et al. 416 [140]	78.6	73.7	77.6	85.3	85.9	86.1	52.4	80.8	75.5	84.5	80.7
Zhang et al. 416 [139]	81.6	78.4	89.5	89.5	84.9	84.8	55.1	86.9	74.3	90.8	82.0
DSP-YOLO 416 [138]	82.2	75.0	89.2	89.3	89.8	85.8	56.6	84.4	81.1	89.1	81.6
SSD 512 [35]	76.8	70.2	84.9	85.2	83.9	79.7	50.3	77.9	73.9	82.5	75.3
ASSD 512 [133]	81.0	76.6	88.2	86.7	85.7	82.8	59.2	83.6	80.5	87.5	80.8
PDS-Net 512 [141]	84.2	74.5	86.5	91.8	91.9	89.7	59.9	**92.8**	79.3	89.1	84.6
SLMS-SSD 512 [142]	81.2	74.6	87.3	88.6	86.5	82.2	54.8	85.5	80.9	87.9	81.0
STDN513 [144]	80.9	79.5	86.4	89.2	88.5	79.3	53.0	77.9	81.4	86.6	**85.5**
DSP-YOLO 608 [138]	83.1	75.5	89.0	90.4	88.6	87.3	55.1	87.3	80.0	86.9	80.0
RON384++ [143]	77.6	73.3	85.9	86.8	82.8	79.6	52.4	78.2	76.0	86.2	78.0
SFGNet [149]	81.2	72.4	**90.3**	89.9	83.5	82.5	67.8	79.0	81.6	86.7	75.7
SL-YOLO [148]	81.2	79.5	78.8	88.2	86.5	81.1	**71.2**	84.4	79.2	82.7	76.3
YOLOv5s 640	83.7	79.1	86.0	91.4	89.3	89.5	60.5	86.1	76.0	86.8	84.9
Ours 640	**85.1**	76.7	88.1	**92.6**	**92.2**	**90.0**	61.2	89.4	79.0	90.6	84.7

Note: In the table, 20 categories are divided into 2 rows, with 10 detection results for each row. The boldface values represent the maximum value in the column.

5.3.7. Performance Comparison in the MS COCO Dataset

In order to further demonstrate the advantages of this method in detecting small and dense targets, an experimental comparison between this method and other methods in the literature was conducted on the MS COCO test dataset. From the experimental results in Table 8, it can be seen that compared with R-FCN [150], SSD [35], FESSD [136], YOLOv3 [37], GC-YOLOv3 [137], Mini-YOLOv4-tiny [151], TRC-YOLO [152], Trident-YOLO [153], SLMS-SSD [142], BANet_S [154], STDN [144], SFGNet [149], YOLO-T [146], SL-YOLO [148], and YOLOv5s, the improved algorithm proposed in this article based on YOLOv5s significantly improved detection accuracy. Compared to the original YOLOv5s algorithm, the overall detection accuracy of the algorithm proposed in this article was improved by 3.1%, and the detection accuracy of small, medium, and large targets was improved by 0.2%, 1.9%, and 3.9%, respectively. Experimental results show that the proposed algorithm outperforms the

original YOLOv5s algorithm in small target detection, medium target detection, and large target detection.

Table 8. MS COCO test-dev detection results.

Method	AP	AP_{50}	AP_{75}	AP Small	AP Medium	AP Large
R-FCN [150]	29.2	51.5	-	10.3	32.4	43.3
SSD 300 [35]	25.1	43.1	25.8	6.6	25.9	41.4
FESSD 300 [136]	28.3	-	29.6	-	-	-
Zhe et al. 300 [135]	29.9	49.9	31.3	10.6	24.5	47.6
YOLOv3 416 [37]	31.0	55.3	32.3	15.2	33.2	42.8
GC-YOLOv3 416 [137]	-	55.5		-	-	-
Mini-YOLOv4-tiny 416 [151]	23.4	42.2	23.4	-	-	-
TRC-YOLO 416 [152]	18.4	38.4	15.6	6.3	17.6	27.2
Trident-YOLO 416 [153]	18.8	37.0	17.3	20.9	25.1	29.3
He et al. 416 [140]	23.6	43.8	26.8	8.4	27.1	42.3
SSD 512 [35]	28.8	48.5	30.3	10.9	31.8	43.5
SLMS-SSD 512 [142]	30.8	52.4	32.0	16.1	33.7	44.0
BANet_S 640 [154]	40.2	58.6		**23.5**	**44.6**	**53.2**
STDN 513 [144]	31.8	51.0	33.6	14.4	36.1	43.4
YOLOv3 608 [37]	33.0	57.9	34.4	18.3	35.4	41.9
SFGNet [149]	32.3	54.1	-	-	-	-
Zhong et al. [145]	38.4	56.2	42.1	21.6	43.0	52.4
YOLO-T 640 [146]	**42.0**	58.3	**44.1**	-	-	-
SL-YOLO [148]	36.8	51.3	37.3	11.7	37.7	48.4
YOLOv5s 640	37.4	56.8	40.7	21.2	42.3	49.0
Ours 640	40.8	**59.9**	43.7	21.4	44.2	52.9

The boldface values represent the maximum value in the column.

6. Discussion

In this section, the contribution of the constructed module to the proposed network was explored through ablation experiments.

Ablation experiments were conducted on the RSOD dataset to study the effects of CARAFE, SA attention, SPPCSPG, GSConv, and DD-head modules on both model accuracy and detection speed. These models were trained on the RSOD dataset and tested on an RTXA5000 GPU. The input size of the test images for the ablation experiment was 640 × 640, and the experimental results are presented in Table 9.

Table 9. Ablation experiment on RSOD dataset.

Model	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10
CARAFE		√					√	√	√	√
SA			√				√	√	√	√
SPPCSPG				√				√	√	√
GSConv					√				√	√
DD-head						√				√
Params(M)	7.02 M	7.15 M	7.02 M	9.95 M	6.35 M	7.28 M	7.15 M	10.09 M	9.41 M	9.67 M
FLOPs(G)	15.8	16.3	15.8	18.1	14.6	16.7	16.3	18.6	17.5	18.4
mAP@0.5 (%)	0.950	0.952	0.952	0.955	0.953	0.955	0.955	0.958	0.961	0.964
mAP@0.5:0.95 (%)	0.653	0.659	0.652	0.676	0.649	0.674	0.671	0.672	0.669	0.648
FPS	119.0	110.9	113.9	113.1	107.8	106.8	109.9	96.2	91.2	88.8

In Table 9, CARAFE represents the replacement of the nearest-neighbor interpolation up-sampling module in the original YOLOv5s network with the CARAFE module. SA represents the embedding of the SA attention module in the head structure of the YOLOv5s network. SPPCSPG represents the replacement of the SPPF module in the original YOLOv5s network with the SPPCSPG module. GSConv represents the replacement of the Conv module in the neck structure of the YOLOv5s network with the GSConv module and the replacement of the C3 module with the improved VoV-GSCSP module. DD-head represents

the replacement of the coupled head module in the original YOLOv5s network with the DD-head module. The presence or absence of a checkmark (√) indicates whether the proposed improvement module was incorporated into the YOLOv5s network.

Model 1 is the original YOLOv5s network, while models 2–10 are the corresponding improved YOLOv5s networks. Analysis of the results in Table 2 shows that embedding the CARAFE, SA attention, SPPCSPG, GSConv, and DD-head modules separately into the original YOLOv5s network can improve the detection accuracy of the network. The detection performance measured by mAP@0.5 is improved by 0.2%, 0.2%, 0.5%, 0.5%, and 0.3%, respectively, compared with the original YOLOv5s network. Analysis of models 7–10 reveals that the combination of multiple improved modules performs better than individual improved modules, indicating that each introduced module contributes to the effective improvement of the model's detection performance.

In order to verify the improvement effect of the proposed model on the accuracy of general object detection, ablation experiments were conducted on the PASCAL VOC dataset, and the experimental results are shown in Table 10.

Table 10. Ablation experiment on PASCAL VOC dataset.

Model	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10
CARAFE		√					√	√	√	√
SA			√				√	√	√	√
SPPCSPG				√				√	√	√
GSConv					√				√	√
DD-head						√				√
Params(M)	7.06 M	7.20 M	7.06 M	10.00 M	6.39 M	7.32 M	7.20 M	10.13 M	9.46 M	9.71 M
FLOPs(G)	15.9	16.4	15.9	18.3	14.8	16.9	16.4	18.8	17.6	18.6
mAP@0.5 (%)	0.837	0.840	0.839	0.847	0.841	0.842	0.841	0.848	0.850	0.851
mAP@0.5:0.95 (%)	0.585	0.595	0.588	0.606	0.603	0.596	0.595	0.615	0.619	0.619
FPS	143.0	125.4	141.9	122.5	125.2	128.6	120.8	110.8	98.4	90.2

Similarly, analyzing the results of models 2–6 in Table 10, it can be seen that embedding the CARAFE module, SA attention module, SPPCSPG module, GSConv module, and DD-head module separately in the original YOLOv5s network can improve the mean average precision accuracy of the network to levels higher than the detection indicators of the original YOLOv5s network; mAP@0.5 increased by 0.3%, 0.2%, 1.0%, 0.4%, and 0.5% respectively. Analyzing models 7–10, it is apparent that the results of multiple improved module combinations are better than those of a single improved module, indicating that the introduced improved models have effectively improved the detection performance of the model.

7. Conclusions

This paper proposes a lightweight target detection algorithm based on YOLOv5s to improve the detection performance of the model while meeting the real-time detection requirements. Specifically, this article constructs a DD-head to replace the coupled head of YOLOv5s based on a decoupled head and depthwise convolution to improve the negative impact of classification and regression task conflicts. An SPPCSPG module based on the SPPCSPC module and GSConv module is constructed to replace the SPPF module of YOLOv5s, which improves the utilization of multi-scale information. An SA attention mechanism is introduced in the head structure to enhance spatial attention and reconstruct channel attention. A CARAFE module is introduced in the up-sampling operation to reassemble feature points with similar semantic information in a content-aware manner and aggregate features in a larger receptive field to fully fuse semantic information. In the neck structure, the GSConv module and the reconstructed VoV-GSCSP module are introduced to maintain detection accuracy while reducing the number of parameters. The experiments show that the constructed algorithm performs better than the original network not only for remote sensing images but also for conventional object detection images. The model built in this research is for target detection in remote sensing datasets, and the

equipment used is a high-performance GPU. The detection performance of the model built on resource-constrained edge computing devices has not been tested. The next research work will focus on how to build a real-time detection model for edge computing devices.

Author Contributions: P.L. conducted the research. Q.W. revised the paper and guided the research. J.M. and P.L. were responsible for data collection, creating the figures, and revising the paper. H.Z., J.M., P.L., Q.W. and Y.L. revised and improved the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (No.42074039) and Post graduate Research and Practice Innovation Program of Jiangsu Province (No. SJCX21_0040).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments: This research was supported by Foundation items: National Natural Science Foundation of China and Post graduate Research and Practice Innovation Program of Jiangsu Province.

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

References

1. Haq, M.A.; Ahmed, A.; Khan, I.; Gyani, J.; Mohamed, A.; Attia, E.; Mangan, P.; Pandi, D. Analysis of environmental factors using AI and ML methods. *Sci. Rep.* **2022**, *12*, 13267. [CrossRef] [PubMed]
2. Haq, M.A.; Jilani, A.K.; Prabu, P. Deep Learning Based Modeling of Groundwater Storage Change. *CMC Comput. Mat. Contin.* **2022**, *70*, 4599–4617.
3. Haq, M.A. CDLSTM: A Novel Model for Climate Change Forecasting. *CMC Comput. Mat. Contin.* **2022**, *71*, 2363–2381.
4. Haq, M.A. SMOTEDNN: A Novel Model for Air Pollution Forecasting and AQI Classification. *CMC Comput. Mat. Contin.* **2022**, *71*, 1403–1425.
5. Ning, Z.; Sun, S.; Wang, X.; Guo, L.; Wang, G.; Gao, X.; Kwok, R.Y.K. Intelligent resource allocation in mobile blockchain for privacy and security transactions: A deep reinforcement learning based approach. *Sci. China Inf. Sci.* **2021**, *64*, 162303. [CrossRef]
6. Xu, Y.; Wang, H.; Liu, X.; He, H.R.; Gu, Q.; Sun, W. Learning to See the Hidden Part of the Vehicle in the Autopilot Scene. *Electronics* **2019**, *8*, 331. [CrossRef]
7. Abdollahi, A.; Pradhan, B.; Shukla, N.; Chakraborty, S.; Alamri, A. Deep Learning Approaches Applied to Remote Sensing Datasets for Road Extraction: A State-Of-The-Art Review. *Remote Sens.* **2020**, *12*, 144. [CrossRef]
8. Liu, P.; Wang, Q.; Yang, G.; Li, L.; Zhang, H. Survey of Road Extraction Methods in Remote Sensing Images Based on Deep Learning. *PFG—J. Photogramm. Remote Sens. Geoinf. Sci.* **2022**, *90*, 135–159. [CrossRef]
9. Jia, D.; He, Z.; Zhang, C.; Yin, W.; Wu, N.; Li, Z. Detection of cervical cancer cells in complex situation based on improved YOLOv3 network. *Multimed. Tools Appl.* **2022**, *81*, 8939–8961. [CrossRef]
10. Shaheen, H.; Ravikumar, K.; Lakshmipathi Anantha, N.; Uma Shankar Kumar, A.; Jayapandian, N.; Kirubakaran, S. An efficient classification of cirrhosis liver disease using hybrid convolutional neural network-capsule network. *Biomed. Signal. Process. Control.* **2023**, *80*, 104152. [CrossRef]
11. Yang, J.; Guo, X.; Li, Y.; Marinello, F.; Ercisli, S.; Zhang, Z. A survey of few-shot learning in smart agriculture: Developments, applications, and challenges. *Plant Methods* **2022**, *18*, 28. [CrossRef] [PubMed]
12. Lv, Z.; Zhang, S.; Xiu, W. Solving the Security Problem of Intelligent Transportation System with Deep Learning. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 4281–4290. [CrossRef]
13. Shaik, A.S.; Karsh, R.K.; Islam, M.; Laskar, R.H. A review of hashing based image authentication techniques. *Multimed. Tools Appl.* **2022**, *81*, 2489–2516. [CrossRef]
14. Li, Y.; Zhang, H.; Xue, X.; Jiang, Y.; Shen, Q. Deep learning for remote sensing image classification: A survey. *WIREs Data Min. Knowl. Discov.* **2018**, *8*, e1264. [CrossRef]
15. Fan, D.; Ji, G.; Cheng, M.; Shao, L. Concealed Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 6024–6042. [CrossRef] [PubMed]
16. Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3523–3542. [CrossRef]
17. Liu, X.; He, J.; Yao, Y.; Zhang, J.; Liang, H.; Wang, H.; Hong, Y. Classifying urban land use by integrating remote sensing and social media data. *Int. J. Geogr. Inf. Sci.* **2017**, *31*, 1675–1696. [CrossRef]

18. Liu, R.; Tao, F.; Liu, X.; Na, J.; Leng, H.; Wu, J.; Zhou, T. RAANet: A Residual ASPP with Attention Framework for Semantic Segmentation of High-Resolution Remote Sensing Images. *Remote Sens.* **2022**, *14*, 3109. [CrossRef]
19. Li, S.; Lyu, D.; Huang, G.; Zhang, X.; Gao, F.; Chen, Y.; Liu, X. Spatially varying impacts of built environment factors on rail transit ridership at station level: A case study in Guangzhou, China. *J. Transp. Geogr.* **2020**, *82*, 102631. [CrossRef]
20. Hu, S.; Fong, S.; Yang, L.; Yang, S.; Dey, N.; Millham, R.C.; Fiaidhi, J. Fast and Accurate Terrain Image Classification for ASTER Remote Sensing by Data Stream Mining and Evolutionary-EAC Instance-Learning-Based Algorithm. *Remote Sens.* **2021**, *13*, 1123. [CrossRef]
21. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
22. Tang, X.; Zhou, P.; Wang, P. Real-time image-based driver fatigue detection and monitoring system for monitoring driver vigilance. In Proceedings of the 2016 35th Chinese Control Conference (CCC), Chengdu, China, 27–29 July 2016; pp. 4188–4193.
23. Alexe, B.; Deselaers, T.; Ferrari, V. Measuring the Objectness of Image Windows. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2189–2202. [CrossRef]
24. Yap, M.H.; Pons, G.; Martí, J.; Ganau, S.; Sentís, M.; Zwiggelaar, R.; Davison, A.K.; Martí, R. Automated Breast Ultrasound Lesions Detection Using Convolutional Neural Networks. *IEEE J. Biomed. Health Inform.* **2018**, *22*, 1218–1226. [CrossRef]
25. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
26. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
27. 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]
28. Kong, T.; Yao, A.; Chen, Y.; Sun, F. HyperNet: Towards Accurate Region Proposal Generation and Joint 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. 845–853.
29. Cho, M.; Chung, T.Y.; Lee, H.; Lee, S. N-RPN: Hard Example Learning for Region Proposal Networks. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 3955–3959.
30. Rao, Y.; Cheng, Y.; Xue, J.; Pu, J.; Wang, Q.; Jin, R.; Wang, Q. FPSiamRPN: Feature Pyramid Siamese Network with Region Proposal Network for Target Tracking. *IEEE Access* **2020**, *8*, 176158–176169. [CrossRef]
31. Zhong, Q.; Li, C.; Zhang, Y.; Xie, D.; Yang, S.; Pu, S. Cascade region proposal and global context for deep object detection. *Neurocomputing* **2020**, *395*, 170–177. [CrossRef]
32. Cai, C.; Chen, L.; Zhang, X.; Gao, Z. End-to-End Optimized ROI Image Compression. *IEEE Trans. Image Process.* **2020**, *29*, 3442–3457. [CrossRef] [PubMed]
33. Shaik, A.S.; Karsh, R.K.; Islam, M.; Singh, S.P.; Wan, S. A Secure and Robust Autoencoder-Based Perceptual Image Hashing for Image Authentication. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 1645658. [CrossRef]
34. Seferbekov, S.; Iglovikov, V.; Buslaev, A.; Shvets, A. Feature Pyramid Network for Multi-class Land Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 272–273.
35. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37.
36. 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.
37. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
38. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
39. 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.
40. Wang, C.; Liao, H.M.; Wu, Y.; Chen, P.; Hsieh, J.; Yeh, I. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 13–19 June 2020; pp. 1571–1580.
41. 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]
42. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
43. Li, H.; Xiong, P.; An, J.; Wang, L. Pyramid Attention Network for Semantic Segmentation. *arXiv* **2018**, arXiv:1805.10180.
44. Xu, J.; Sun, X.; Zhang, D.; Fu, K. Automatic Detection of Inshore Ships in High-Resolution Remote Sensing Images Using Robust Invariant Generalized Hough Transform. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 2070–2074.
45. Cucchiara, R.; Grana, C.; Piccardi, M.; Prati, A.; Sirotti, S. Improving shadow suppression in moving object detection with HSV color information. In Proceedings of the ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585), Oakland, CA, USA, 25–29 August 2001; pp. 334–339.

46. Corbane, C.; Najman, L.; Pecoul, E.; Demagistri, L.; Petit, M. A complete processing chain for ship detection using optical satellite imagery. *Int. J. Remote Sens.* **2010**, *31*, 5837–5854. [CrossRef]
47. Li, Z.; Itti, L. Saliency and Gist Features for Target Detection in Satellite Images. *IEEE Trans. Image Process.* **2011**, *20*, 2017–2029.
48. Brekke, C.; Solberg, A.H.S. Oil spill detection by satellite remote sensing. *Remote Sens. Environ.* **2005**, *95*, 1–13. [CrossRef]
49. Cheng, G.; Han, J.; Guo, L.; Qian, X.; Zhou, P.; Yao, X.; Hu, X. Object detection in remote sensing imagery using a discriminatively trained mixture model. *ISPRS J. Photogramm. Remote Sens.* **2013**, *85*, 32–43. [CrossRef]
50. Hinz, S.; Stilla, U. Car detection in aerial thermal images by local and global evidence accumulation. *Pattern Recognit. Lett.* **2006**, *27*, 308–315. [CrossRef]
51. Pang, J.; Li, C.; Shi, J.; Xu, Z.; Feng, H. R2-CNN: Fast Tiny Object Detection in Large-Scale Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5512–5524. [CrossRef]
52. Fu, Y.; Wu, F.; Zhao, J. Context-Aware and Depthwise-based Detection on Orbit for Remote Sensing Image. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 1725–1730.
53. Cheng, G.; Zhou, P.; Han, J. Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7405–7415. [CrossRef]
54. Cheng, G.; Han, J.; Zhou, P.; Xu, D. Learning Rotation-Invariant and Fisher Discriminative Convolutional Neural Networks for Object Detection. *IEEE Trans. Image Process.* **2019**, *28*, 265–278. [CrossRef]
55. Yang, X.; Sun, H.; Sun, X.; Yan, M.; Guo, Z.; Fu, K. Position Detection and Direction Prediction for Arbitrary-Oriented Ships via Multitask Rotation Region Convolutional Neural Network. *IEEE Access* **2018**, *6*, 50839–50849. [CrossRef]
56. Zhang, W.; Wang, S.; Thachan, S.; Chen, J.; Qian, Y. Deconv R-CNN for Small Object Detection on Remote Sensing Images. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 2483–2486.
57. Li, L.; Cheng, L.; Guo, X.; Liu, X.; Jiao, L.; Liu, F. Deep Adaptive Proposal Network in Optical Remote Sensing Images Objective Detection. In Proceedings of the IGARSS 2020—2020 IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA, 26 September–2 October 2020; pp. 2651–2654.
58. Guo, W.; Yang, W.; Zhang, H.; Hua, G. Geospatial Object Detection in High Resolution Satellite Images Based on Multi-Scale Convolutional Neural Network. *Remote Sens.* **2018**, *10*, 131. [CrossRef]
59. Zhang, X.; Zhu, K.; Chen, G.; Tan, X.; Zhang, L.; Dai, F.; Liao, P.; Gong, Y. Geospatial Object Detection on High Resolution Remote Sensing Imagery Based on Double Multi-Scale Feature Pyramid Network. *Remote Sens.* **2019**, *11*, 755. [CrossRef]
60. Li, K.; Cheng, G.; Bu, S.; You, X. Rotation-Insensitive and Context-Augmented Object Detection in Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 2337–2348. [CrossRef]
61. Li, Q.; Mou, L.; Jiang, K.; Liu, Q.; Wang, Y.; Zhu, X. Hierarchical Region Based Convolution Neural Network for Multiscale Object Detection in Remote Sensing Images. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 4355–4358.
62. Guo, M.; Xu, T.; Liu, J.; Liu, Z.; Jiang, P.; Mu, T.; Zhang, S.; Martin, R.R.; Cheng, M.; Hu, S. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [CrossRef]
63. Woo, S.; Park, J.; Lee, J.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 3–19.
64. Hao, Z.; Wang, Z.; Bai, D.; Tao, B.; Tong, X.; Chen, B. Intelligent Detection of Steel Defects Based on Improved Split Attention Networks. *Front. Bioeng. Biotechnol.* **2022**, *9*, 810876. [CrossRef]
65. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual Attention Network for Scene Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 3141–3149.
66. Guan, Q.; Huang, Y.; Zhong, Z.; Zheng, Z.; Zheng, L.; Yang, Y. Diagnose like a Radiologist: Attention Guided Convolutional Neural Network for Thorax Disease Classification. *arXiv* **2018**, arXiv:1801.09927.
67. Dai, T.; Cai, J.; Zhang, Y.; Xia, S.T.; Zhang, L. Second-Order Attention Network for Single Image Super-Resolution. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 11057–11066.
68. Xu, T.; Zhang, P.; Huang, Q.; Zhang, H.; Gan, Z.; Huang, X.; He, X. AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1316–1324.
69. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef] [PubMed]
70. Shaik, A.S.; Karsh, R.K.; Suresh, M.; Gunjan, V.K. LWT-DCT Based Image Hashing for Tampering Localization via Blind Geometric Correction. In *ICDSMLA 2020*; Kumar, A., Senatore, S., Gunjan, V.K., Eds.; Springer: Singapore, 2022; pp. 1651–1663.
71. Mnih, V.; Heess, N.; Graves, A.; Kavukcuoglu, K. Recurrent Models of Visual Attention. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2204–2212.
72. Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial Transformer Networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 2017–2025.

73. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Vedaldi, A. Gather-Excite: Exploiting Feature Context in Convolutional Neural Networks. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 9423–9433.
74. Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; Liu, W. CCNet: Criss-Cross Attention for Semantic Segmentation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 603–612.
75. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
76. Li, J.; Zhang, S.; Wang, J.; Gao, W.; Tian, Q. Global-Local Temporal Representations for Video Person Re-Identification. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3957–3966.
77. Liu, Z.; Wang, L.; Wu, W.; Qian, C.; Lu, T. TAM: Temporal Adaptive Module for Video Recognition. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 13688–13698.
78. Srivastava, R.K.; Greff, K.; Schmidhuber, J.U.R. Training Very Deep Networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 2377–2385.
79. Li, X.; Wang, W.; Hu, X.; Yang, J. Selective Kernel Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 510–519.
80. Yang, B.; Bender, G.; Le, Q.V.; Ngiam, J. CondConv: Conditionally Parameterized Convolutions for Efficient Inference. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 1307–1318.
81. Chen, Y.; Dai, X.; Liu, M.; Chen, D.; Yuan, L.; Liu, Z. Dynamic Convolution: Attention Over Convolution Kernels. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11027–11036.
82. Chen, L.; Zhang, H.; Xiao, J.; Nie, L.; Shao, J.; Liu, W.; Chua, T.S. SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6298–6306.
83. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
84. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual Attention Network for Image Classification. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6450–6458.
85. Song, S.; Lan, C.; Xing, J.; Zeng, W.; Liu, J. An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4263–4270.
86. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.Y.; Liu, J. LSTM network: A deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.* **2017**, *11*, 68–75. [CrossRef]
87. Wang, L.; Ouyang, W.; Wang, X.; Lu, H. Visual Tracking with Fully Convolutional Networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 3119–3127.
88. Ghiasi, G.; Lin, T.Y.; Le, Q.V. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7029–7038.
89. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787.
90. Merugu, S.; Tiwari, A.; Sharma, S.K. Spatial–Spectral Image Classification with Edge Preserving Method. *J. Indian Soc. Remote Sens.* **2021**, *49*, 703–711. [CrossRef]
91. Liu, S.; Di, H.; Wang, Y. Learning Spatial Fusion for Single-Shot Object Detection. *arXiv* **2019**, arXiv:1911.09516.
92. Zhang, Y.; Wang, W.; Li, Z.; Shu, S.; Lang, X.; Zhang, T.; Dong, J. Development of a cross-scale weighted feature fusion network for hot-rolled steel surface defect detection. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105628. [CrossRef]
93. Qiu, M.; Huang, L.; Tang, B. Bridge detection method for HSRRSIs based on YOLOv5 with a decoupled head. *Int. J. Digit. Earth* **2023**, *16*, 113–129. [CrossRef]
94. Liang, M.; Liu, X.; Hu, X. Small target detection algorithm for train operating environment image based on improved YOLOv3. *J. Comput. Appl.* **2023**, 1–12.
95. Li, W.; Chen, L.; Xe, X.; Hao, X.; Li, H. An Algorithm for Detecting Prohibited Items in X-ray Images Based on Improved YOLOv5. *Comput. Eng. Appl.* **2023**, *42*, 2675–2683.
96. Zhao, W.; Syafrudin, M.; Fitriyani, N.L. CRAS-YOLO: A Novel Multi-Category Vessel Detection and Classification Model Based on YOLOv5s Algorithm. *IEEE Access* **2023**, *11*, 11463–11478. [CrossRef]
97. Luo, X.; Wu, Y.; Zhao, L. YOLOD: A Target Detection Method for UAV Aerial Imagery. *Remote Sens.* **2022**, *14*, 3240. [CrossRef]
98. Yun, S.; Han, D.; Chun, S.; Oh, S.J.; Yoo, Y.; Choe, J. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE: Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6022–6031.

99. Ran, X.; Zhou, X.; Lei, M.; Tepsan, W.; Deng, W. A Novel K-Means Clustering Algorithm with a Noise Algorithm for Capturing Urban Hotspots. *Appl. Sci.* **2021**, *11*, 11202. [CrossRef]
100. Li, Z.; Yang, S.; Deshuai, S.; Liu, X.; Zheng, Y. Yield estimation method of apple tree based on improved lightweight YOLOv5. *Smart Agric.* **2021**, *3*, 100–114.
101. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation. *IEEE Trans. Cybern.* **2022**, *52*, 8574–8586. [CrossRef] [PubMed]
102. Li, X.; Hu, X.; Yang, J. Spatial Group-wise Enhance: Improving Semantic Feature Learning in Convolutional Networks. *arXiv* **2019**, arXiv:1905.09646.
103. Zhang, Q.; Yang, Y. SA-Net: Shuffle Attention for Deep Convolutional Neural Networks. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 2235–2239.
104. Song, G.; Liu, Y.; Wang, X. Revisiting the Sibling Head in Object Detector. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11560–11569.
105. Wu, Y.; Chen, Y.; Yuan, L.; Liu, Z.; Wang, L.; Li, H.; Fu, Y. Rethinking Classification and Localization for Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10183–10192.
106. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
107. Gao, H.; Yang, Y.; Li, C.; Gao, L.; Zhang, B. Multiscale Residual Network with Mixed Depthwise Convolution for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 3396–3408. [CrossRef]
108. Wang, J.; Chen, K.; Xu, R.; Liu, Z.; Loy, C.C.; Lin, D. CARAFE: Content-Aware ReAssembly of FEatures. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3007–3016.
109. Zhang, M.; Gao, F.; Yang, W.; Zhang, H. Wildlife Object Detection Method Applying Segmentation Gradient Flow and Feature Dimensionality Reduction. *Electronics* **2023**, *12*, 377. [CrossRef]
110. Yang, Z.; Li, L.; Luo, W.; Ning, X. PDNet: Improved YOLOv5 Nondeformable Disease Detection Network for Asphalt Pavement. *Comput. Intell. Neurosci.* **2022**, *2022*, 5133543. [CrossRef] [PubMed]
111. Wu, F.; Duan, J.; Ai, P.; Chen, Z.; Yang, Z.; Zou, X. Rachis detection and three-dimensional localization of cut off point for vision-based banana robot. *Comput. Electron. Agric.* **2022**, *198*, 107079. [CrossRef]
112. Wang, C.; Mark, A.B.; Liao, M.H. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
113. Long, Y.; Gong, Y.; Xiao, Z.; Liu, Q. Accurate Object Localization in Remote Sensing Images Based on Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2486–2498. [CrossRef]
114. Xiao, Z.; Liu, Q.; Tang, G.; Zhai, X. Elliptic Fourier transformation-based histograms of oriented gradients for rotationally invariant object detection in remote-sensing images. *Int. J. Remote Sens.* **2015**, *36*, 618–644. [CrossRef]
115. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [CrossRef]
116. Everingham, M.; Eslami, S.M.A.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
117. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
118. Lin, T.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 740–755.
119. Lin, H.; Zhou, J.; Gan, Y.; Vong, C.; Liu, Q. Novel up-scale feature aggregation for object detection in aerial images. *Neurocomputing* **2020**, *411*, 364–374. [CrossRef]
120. Li, Y.; Huang, Q.; Pei, X.; Chen, Y.; Jiao, L.; Shang, R. Cross-Layer Attention Network for Small Object Detection in Remote Sensing Imagery. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2021**, *14*, 2148–2161. [CrossRef]
121. Yao, Y.; Cheng, G.; Xie, X.; Han, J. Optical remote sensing image object detection based on multiresolution feature fusion. *Natl. Remote Sens. Bull.* **2021**, *25*, 1124–1137.
122. Yuan, Z.; Liu, Z.; Zhu, C.; Qi, J.; Zhao, D. Object Detection in Remote Sensing Images via Multi-Feature Pyramid Network with Receptive Field Block. *Remote Sens.* **2021**, *13*, 862. [CrossRef]
123. Wang, G.; Zhuang, Y.; Chen, H.; Liu, X.; Zhang, T.; Li, L.; Dong, S.; Sang, Q. FSoD-Net: Full-Scale Object Detection from Optical Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–18. [CrossRef]
124. Liu, N.; Mao, Z.; Wang, Y.; Shen, J. Remote Sensing Images Target Detection Based on Adjustable Parameter and Receptive field. *Acta Photonica Sin.* **2021**, *50*, 302–313.
125. Zhang, T.; Zhuang, Y.; Wang, G.; Dong, S.; Chen, H.; Li, L. Multiscale Semantic Fusion-Guided Fractal Convolutional Object Detection Network for Optical Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–20. [CrossRef]
126. Xue, J.; Zheng, Y.G.; Dong, C.; Wang, P.; Yasir, M. Improved YOLOv5 network method for remote sensing image-based ground objects recognition. *Soft Comput.* **2022**, *26*, 10879–10889. [CrossRef]

127. Sun, Y.; Liu, W.; Gao, Y.; Hou, X.; Bi, F. A Dense Feature Pyramid Network for Remote Sensing Object Detection. *Appl. Sci.* **2022**, *12*, 4997. [CrossRef]
128. Liu, H.; Zhang, L.; Wang, F.; He, R. Object detection algorithm based on attention mechanism and context information. *J. Comput. Appl.* **2022**, 1–9.
129. Yang, X.; Yan, J.; Liao, W.; Yang, X.; Tang, J.; He, T. SCRDet++: Detecting Small, Cluttered and Rotated Objects via Instance-Level Feature Denoising and Rotation Loss Smoothing. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 2384–2399. [CrossRef]
130. Chen, W.; Han, B.; Yang, Z.; Gao, X. MSSDet: Multi-Scale Ship-Detection Framework in Optical Remote-Sensing Images and New Benchmark. *Remote Sens.* **2022**, *14*, 5460. [CrossRef]
131. Gao, P.; Cao, X.; Li, K.; You, X. Object Detection in Remote Sensing Images by Fusing Multi-neuron Sparse Features and Hierarchical Depth Features. *J. Geo Inf. Sci.* **2023**, *25*, 638–653.
132. Chen, J.; Hong, H.; Song, B.; Guo, J.; Chen, C.; Xu, J. MDCT: Multi-Kernel Dilated Convolution and Transformer for One-Stage Object Detection of Remote Sensing Images. *Remote Sens.* **2023**, *15*, 371. [CrossRef]
133. Zhao, H.; Li, Z.; Zhang, T. Attention Based Single Shot Multibox Detector. *J. Electron. Inf. Technol.* **2021**, *43*, 2096–2104.
134. Qu, Z.; Han, T.; Yi, T. MFFAMM: A Small Object Detection with Multi-Scale Feature Fusion and Attention Mechanism Module. *Appl. Sci.* **2022**, *12*, 8940. [CrossRef]
135. Yang, Z.; Bu, Z.; Liu, C. SSD Optimization Model Based on Shallow Feature Fusion. *Int. J. Pattern Recognit. Artif. Intell.* **2022**, *36*, 2259033. [CrossRef]
136. Qian, H.; Wang, H.; Feng, S.; Yan, S. FESSD: SSD target detection based on feature fusion and feature enhancement. *J. Real Time Image Process.* **2023**, *20*, 2. [CrossRef]
137. Yang, Y.; Deng, H. GC-YOLOv3: You Only Look Once with Global Context Block. *Electronics* **2020**, *9*, 1235. [CrossRef]
138. Zhang, X.; Gao, Y.; Wang, H.; Wang, Q. Improve YOLOv3 using dilated spatial pyramid module for multi-scale object detection. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1738093438. [CrossRef]
139. Zhang, M.; Xu, S.; Song, W.; He, Q.; Wei, Q. Lightweight Underwater Object Detection Based on YOLO v4 and Multi-Scale Attentional Feature Fusion. *Remote Sens.* **2021**, *13*, 4706. [CrossRef]
140. He, X.; Song, X. Improved YOLOv4-Tiny lightweight target detection algorithm. *J. Front. Comput. Sci. Technol.* **2023**, 1–17.
141. Junayed, M.S.; Islam, M.B.; Imani, H.; Aydin, T. PDS-Net: A novel point and depth-wise separable convolution for real-time object detection. *Int. J. Multimed. Inf. Retr.* **2022**, *11*, 171–188. [CrossRef]
142. Wang, K.; Wang, Y.; Zhang, S.; Tian, Y.; Li, D. SLMS-SSD: Improving the balance of semantic and spatial information in object detection. *Expert Syst. Appl.* **2022**, *206*, 117682. [CrossRef]
143. Kong, T.; Sun, F.; Yao, A.; Liu, H.; Lu, M.; Chen, Y. RON: Reverse Connection with Objectness Prior Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5244–5252.
144. Zhou, P.; Ni, B.; Geng, C.; Hu, J.; Xu, Y. Scale-Transferrable Object Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 528–537.
145. Qu, Z.; Gao, L.; Wang, S.; Yin, H.; Yi, T. An improved YOLOv5 method for large objects detection with multi-scale feature cross-layer fusion network. *Image Vis. Comput.* **2022**, *125*, 104518. [CrossRef]
146. Tu, X.; Bao, X.; Wu, B.; Jin, Y.; Zhang, Q. Object detection algorithm for 3D coordinate attention path aggregation network. *J. Front. Comput. Sci. Technol.* **2023**, 1–16.
147. Yang, J.; Hong, L.; Du, Y.; Mao, Y.; Liu, Q. A Lightweight Object Detection Algorithm Based on Improved YOLOv5s. *Electron. Opt. Control* **2023**, *30*, 24–30.
148. Song, Z.; Xiao, B.; Ai, Y.; Zheng, L.; Tie, J. Improved lightweight YOLOv4 target detection algorithm. *Electron. Meas. Technol.* **2022**, *45*, 142–152.
149. Hu, J.; Wang, Y.; Cheng, S.; Liu, J.; Kang, J.; Yang, W. SFGNet detecting objects via spatial fine-grained feature and enhanced RPN with spatial context. *Syst. Sci. Control Eng.* **2022**, *10*, 388–406. [CrossRef]
150. Dai, J.F.; Li, Y.; He, K.M.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; Volume 29.
151. Bacea, D.; Oniga, F. Single stage architecture for improved accuracy real-time object detection on mobile devices. *Image Vis. Comput.* **2023**, *130*, 104613. [CrossRef]
152. Wang, G.; Ding, H.; Yang, Z.; Li, B.; Wang, Y.; Bao, L. TRC-YOLO: A real-time detection method for lightweight targets based on mobile devices. *IET Comput. Vis.* **2022**, *16*, 126–142. [CrossRef]
153. Wang, G.; Ding, H.; Li, B.; Nie, R.; Zhao, Y. Trident-YOLO: Improving the precision and speed of mobile device object detection. *IET Image Process.* **2022**, *16*, 145–157. [CrossRef]
154. Xiao, J.; Guo, H.; Zhou, J.; Zhao, T.; Yu, Q.; Chen, Y.; Wang, Z. Tiny object detection with context enhancement and feature purification. *Expert Syst. Appl.* **2023**, *211*, 118665. [CrossRef]