applied sciences

MDPI

Article

YOLO-DSD: A YOLO-Based Detector Optimized for Better Balance between Accuracy, Deployability and Inference Time in Optical Remote Sensing Object Detection

Hengxu Chen, Hong Jin and Shengping Lv *

College of Engineering, South China Agricultural University, Guangzhou 510642, China; hengxuchen@stu.scau.edu.cn (H.C.); hjin@scau.edu.cn (H.J.)
* Correspondence: lvshengping@scau.edu.cn; Tel.: +86-187-1937-3880

Abstract: Many deep learning (DL)-based detectors have been developed for optical remote sensing object detection in recent years. However, most of the recent detectors are developed toward the pursuit of a higher accuracy, but little toward a balance between accuracy, deployability and inference time, which hinders the practical application for these detectors, especially in embedded devices. In order to achieve a higher detection accuracy and reduce the computational consumption and inference time simultaneously, a novel convolutional network named YOLO-DSD was developed based on YOLOv4. Firstly, a new feature extraction module, a dense residual (DenseRes) block, was proposed in a backbone network by utilizing a series-connected residual structure with the same topology for improving feature extraction while reducing the computational consumption and inference time. Secondly, convolution layer–batch normalization layer–leaky ReLu (CBL) ×5 modules in the neck, named S-CBL×5, were improved with a short-cut connection in order to mitigate feature loss. Finally, a low-cost novel attention mechanism called a dual channel attention (DCA) block was introduced to each S-CBL×5 for a better representation of features. The experimental results in the DIOR dataset indicate that YOLO-DSD outperforms YOLOv4 by increasing $mAP^{0.5}$ from 71.3% to 73.0%, with a 23.9% and 29.7% reduction in Params and Flops, respectively, but a 50.2% improvement in FPS. In the RSOD dataset, the $mAP^{0.5}$ of YOLO-DSD is increased from 90.0~94.0% to 92.6~95.5% under different input sizes. Compared with the SOTA detectors, YOLO-DSD achieves a better balance between the accuracy, deployability and inference time.

Keywords: optical remote sensing; object detection; feature extraction; attention mechanism

check for updates

Citation: Chen, H.; Jin, H.; Lv, S. YOLO-DSD: A YOLO-Based Detector Optimized for Better Balance between Accuracy, Deployability and Inference Time in Optical Remote Sensing Object Detection. *Appl. Sci.* **2022**, *12*, 7622. https://doi.org/10.3390/app12157622

Academic Editors: Weitao Chen, Ailong Ma and Guohua Wu

Received: 22 June 2022
Accepted: 25 July 2022
Published: 28 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Object detection in optical remote sensing images (ORSIs) is a crucial but challenging task for remote sensing technology and has been widely applied in many fields, such as military, natural resources exploration, urban construction, agriculture and mapping [1,2]. The development of a cost-effective detector considering the characteristic of ORSIs is the persistently pursued direction, and has attracted a large amount of attention from scholars and practitioners.

The approaches for object detection can be roughly divided into traditional detectors and deep learning (DL)-based detectors. DL-based detectors, especially convolutional neural network (CNN) detectors, have gradually replaced traditional detectors since they possess better adaptability and generalization in different application scenarios. There are two categories of DL-based detectors: one-stage [3–9] and two-stage [10–13]. The one-stage detectors directly regress bounding boxes and probabilities for each object simultaneously without region proposals; thus, they perform well regarding inference speed. Two-stage detectors employ the region proposals to improve the location and detection accuracy, with the sacrifice of the inference speed. With the emergence of large-scale natural scene images (NSIs) datasets for object detection tasks such as Pascal VOC [14] and MS COCO [15],

DL-based detectors have been further developed for a better tradeoff between accuracy and cost, including Faster-RCNN [12], single shot multibox detector (SSD) [3], the series of You Only Look Once (YOLO) [4–6,8], CenterNet [7], EfficientDet [9] and RetinaNet [16]. These detectors with continuous improvement have been widely applied in various natural scene visual detection tasks.

Since ORSIs are photographed from an overhead perspective at different heights, whereas NSIs are shot from a horizontal perspective at relatively close distance, three main differences have emerged as follows: first, the available feature of most detected objects in ORSIs is less obvious than that in NSIs and leads to greater inter-class similarity. Second, the intra-class difference is more prominent since object scales of the same category in ORSIs usually vary greater. Third, the background in ORSIs is more complex and abundant than that in NSIs. Differences between ORSIs and NSIs with instances are shown in Figure 1. These differences make object detection in ORSIs more difficult, and most of the well-designed detectors for NSIs are not elaborately optimized for ORSIs. For the problems of a greater intra-class difference and inter-class similarity caused by the characteristic of objects in ORSIs, the detector needs to extract more abundant object features with high-level semantics to overcome it. However, the feature of objects in ORSIs are easily submerged by the redundant and complex background information and thus will decrease or even disappear when transmitted in the detector. Thus, DL-based detectors also require a stronger feature extraction and transmission ability.

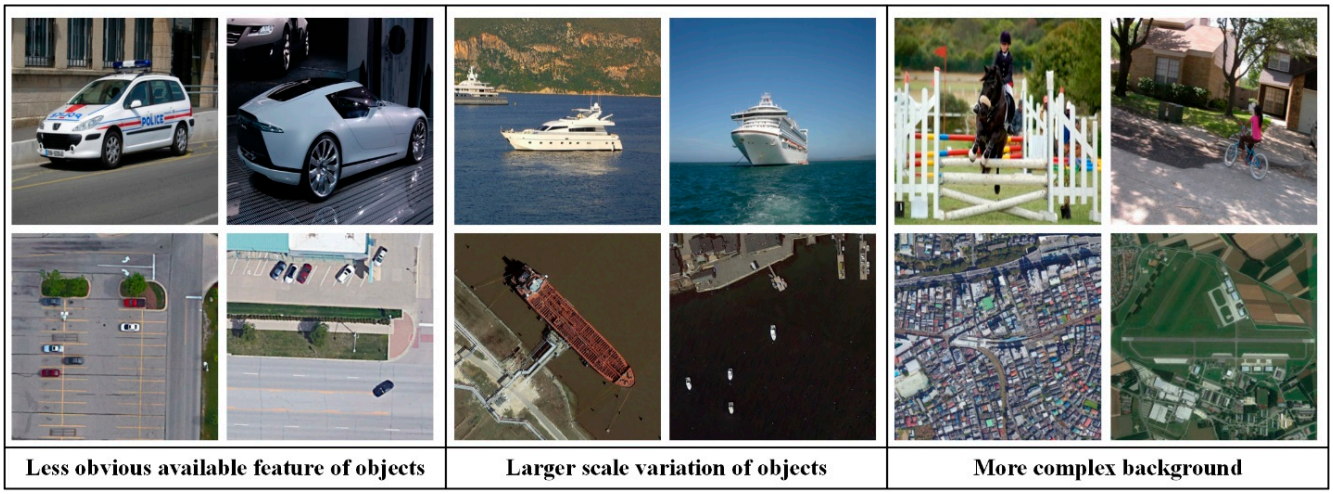

Figure 1. Three main differences between RSIs and NSIs. The first and second lines show instances from NSIs and RSIs, respectively.

With the popularity and wide application of embedded devices such as unmanned aerial vehicles (UAVs), the demand for real-time optical remote sensing object detection deployed on edge devices has increased rapidly. UAVs with far less computing resource and storage space than computers involve wide application scenarios such as rescue, military and surveying tasks, which require a high detection accuracy, flexible equipment deployment and less inference time for detectors [17].

In recent years, several outstanding achievements have been made by researchers in fields related to ORSIs, and can be roughly divided into heavyweight [18–21] and lightweight detectors [22–25]. Most of the heavyweight detectors usually have a high accuracy but require a large computational cost, and thus hinder their real-time response and the deployment on UAVs, whereas lightweight detectors have practical deployability and a fast inference speed but it is difficult for them to achieve as high a competitive accuracy as heavyweight detectors, especially for large multi-category object detection tasks [23,24,26]. Therefore, optimizing the structure of heavyweight detectors toward a better balance between accuracy, deployability and inference time is an issue well worth

investigating. To establish a detector with a better balance between accuracy, deployability and inference time, a novel detector called YOLO-DSD for real-time optical remote sensing object detection based on YOLOv4 was developed in this study. The main contributions are as follows: (1) a new feature extraction module named a dense residual (DenseRes) Block was designed for better feature extraction and to reduce the computational cost and inference time in the backbone network. (2) Convolution layer–batch normalization layer–leaky ReLu (CBL) ×5 modules in the neck were improved with a short-cut connection and named S-CBL×5 to strengthen the transmission of object features. (3) A novel low-cost attention mechanism called a dual channel attention (DCA) Block was proposed to enhance the representation of the object feature. The experimental results in the DIOR dataset indicate that YOLO-DSD outperforms YOLOv4 by increasing $mAP^{0.5}$ from 71.3% to 73.0%, with a 23.9% and 29.7% reduction in Params and Flops, respectively, but a 50.2% improvement in FPS. In the RSOD dataset, the $mAP^{0.5}$ of YOLO-DSD is increased from 90.0~94.0% to 92.6~95.5% under different input sizes. Compared with the SOTA detectors, YOLO-DSD achieves a better balance between accuracy, deployability and inference time.

2. Related Works

2.1. DL-Based Detectors for Optical Remote Sensing Object Detection

DL-based detectors have been widely applied in natural sense visual tasks. However, detectors established on NSIs need to further improve their feature extraction ability for optical remote sensing object detection tasks due to the problems of a greater intra-class difference, inter-class similarity and feature loss in ORSIs. Therefore, some heavyweight detectors have been improved and applied in ORSIs by many scholars. Xu et al. [18] modified YOLOv3 with a multi-receptive field to take full advantage of the feature information and to detect optical remote sensing objects effectively. Cheng et al. [19] designed an end-to-end cross-scale feature fusion framework for ORSIs object detection based on Faster R-CNN with a feature pyramid network (FPN) [16]. Yin et al. [20] proposed a multi-scale feature extraction network based on RetinaNet, which strengthens the detection performance of irregular objects in ORSIs. Yuan et al. [21] established a multi-FPN that performs well in object detection with a complex background. The above research has successfully made obvious improvements in detection accuracy, but come with the non-ignorable sacrifice of the deployability or inference speed, and thus further hinder the application of detectors in edge devices. As a consequence, some lightweight DL-based detectors have been elaborately designed and improved to facilitate the application in edge devices. Li et al. [22] designed a lightweight detector by taking advantage of YOLOv3 and DenseNet [27]. Lang et al. [23] employed the backbone network of ThunerNet [28] and constructed a six-layer feature fusion pyramid to enhance the detection performance. The improved YOLOv4-tiny proposed by Lei et al. [24] was constructed with an efficient channel attention mechanism to enhance the information sensitivity in each channel. Li et al. [25] established a lightweight detector for vehicle and ship detection through using a semantic transfer block and the distillation loss. Although these lightweight detectors have a better accuracy after improvement, there is still an obvious gap in the detection accuracy compared with heavyweight detectors.

Our motivation is to propose an end-to-end detector that can achieve a higher detection accuracy, better deployability and less inference time in order to meet the requirements of edge device real-time detection. YOLOv4 [8] is one of the widely used one-stage detectors, with an impressive performance in accuracy, deployability and inference time. It has been improved and applied in various fields, such as agriculture, industry and transportation [29–32], which verify its excellent generalization. In this study, YOLOv4 was utilized as the basic framework, while it was optimized from feature extraction modules, structures of the neck and the attention mechanism for a better application in optical remote sensing object detection.

2.2. Feature Extraction Modules in Backbone

The backbone that is utilized to extract high-level semantic features of images is the first part of the DL-based detector. It comprises several feature extraction modules. VGG [33] is one of the earliest backbones for object detection and utilizes 3×3 convolution layers as the feature extraction module. However, its heavy computation burden and shallow depth hinder the deployability and performance of detectors.

To solve this problem, He et al. [34] introduced a new feature extraction module named a Res Block to deepen the depth of backbones by adding short-cut connections. ResNet based on the Res Block achieves a better accuracy than VGG in the natural scene dataset, with a lower computation burden and deeper depth. The backbone DarkNet53 of YOLOv3 [6] also uses the Res Block as the main feature extraction module. Since then, many feature extraction modules based on the Res Block, such as a ResNeXt Block [35], Res2 Block [36], Dense Block [27] and CSP Block [37], have been improved and developed. The trunk of the ResNeXt Block is split into 32 paths that transform the input from high to low dimensions and back to high dimensions using the same topology, and aggregates them through element-wise addition. Although the ResNeXt Block outperforms the Res Block with fewer parameters and a higher detection accuracy in the natural scene dataset, since the semantic relevance between background and detected objects in ORSIs is stronger than that in NSIs [38], the operation of the ResNeXt Block easily breaks this relevance, and is thus not conducive to the detection performance in ORSIs. The Res2 Block can generate multi-scale features through a hierarchical short-cut connection and increase in receptive fields, thus improving the detection accuracy and reducing the computational consumption. However, its structure with parallel convolution and interactive operations significantly increases the inference time. The Dense Block contains several dense layers. The output of the dense layer is concatenated with its input, and the concatenated feature map serves as the input of the next dense layer. This structure takes full advantage of the short-cut that can better retain the feature and reduce the computation burden. However, the Dense Block will deteriorate in the situation where the background submerges features of detected objects in ORSIs, since the background information is more redundant and complex. Meanwhile, the structure of the Dense Block will reduce its inference speed due to the asymmetry of the input channels number and output channels number for a convolution operation. The CSP Block is the feature extraction module of the backbone CSP DarkNet in YOLOv4. It is mainly composed of several Res units based on a short-cut and a cross-stage part containing a 1×1 convolution layer. Although this structure can double the number of gradient paths and improve the detection accuracy through a splitting and merging strategy, there is parallel convolution and the problem of the trunk of the CSP Block being stacked alternately by an excessive convolution layer, which significantly increase the degree of network fragmentation and thus decrease the inference speed [39].

In order to alleviate the shortcomings of the above Blocks, a novel feature extraction module DenseRes Block is proposed in this study to improve the backbone in YOLOv4. Firstly, the input feature map of the DenseRes Block was compressed in order to increase the proportion of object feature information. Then, the series-connected residual structure with the same topology was utilized not only to obtain the high-level semantics of the object feature but also to reduce the computational consumption and inference time. Finally, the feature map output from the residual structure was combined with the input of the DenseRes Block to enhance the semantic relevance between background and detected objects.

2.3. Structure of the Neck

In the neck, feature maps output from the backbone will be processed and transmitted to the prediction part of the detector. The neck of the early DL-based detectors only directly transmits the last feature map of the backbone to the prediction part. The shallow feature map contains rich location information but low-level semantic information, whereas the deep feature map is the opposite; thus, this structure is not conducive to object detection,

especially for small objects. In order to improve the detection performance of detectors for small objects, Liu et al. [3] proposed a neck structure that directly transfers the feature maps of different levels from the backbone to the prediction part of the detector for multi-scale detection, and proves that the utilization of a shallow feature map is beneficial for small object detection. However, shallow feature maps still lack high-level semantic information, while deep feature maps are still short of location information. FPN [16] is designed to transfer the high-level semantic information to the shallow feature map through the bottom-up structure to further improve the detection performance of the detector for small objects. In order to make the deep feature map possess rich location information and high-level semantic information, BFPN [40] has been developed to fuse the penultimate feature map and the last feature map based on FPN, while PANet [41] adds a top-down structure based on FPN to transmit location information to the deep feature map. Both BFPN and PANet can improve the detection performance for middle and large objects while maintaining a high detection accuracy for small objects.

YOLOv4 adopts the PANet as the framework in the neck. However, YOLOv4 suffers from the problem of feature loss in ORSIs due to many convolution operations in the neck. Therefore, a short-cut connection based on a residual is introduced to each CBL$\times$5 in the neck for strengthening the transmission of object features without an increase in the computational burden and inference time.

2.4. Attention Mechanism

The attention mechanism assigns different weights to the pixel according to the spatial or channel relationship between the pixels in the feature map to enhance the representation of the feature, and it mainly includes three categories: a channel attention mechanism (e.g., an SE Block [42] and ECA Block [43]), spatial attention mechanism (e.g., a CA Block [44]) and hybrid attention mechanism (e.g., a CBAM Block [45]). The attention mechanism can improve the detection accuracy in NSIs with a few parameters and computation burden increase for detectors. The SE Block squeezes and then extends channel information through two full connection layers in order to learn the relationship of global channel information and effectively improve the detection performance, but the relationship between local channel information is not considered. The ECA Block learns the relationship between local channels through 1-D convolution with an adaptive convolution kernel, which promotes the detection performance but ignores the relationship of global channel information. In the CA Block, the information is extracted by average pooling in horizontal and vertical directions, respectively, and then concatenated and fused by 2-D convolution. The fused information is split into two parts and each part is further extracted by the convolution layer, respectively. The hybrid attention mechanism CBAM Block combines the channel and spatial attention mechanism. Both the CA Block and CBAM Block bring an obvious improvement in the detection accuracy in the natural scene dataset, but their complex structure increase the inference time. Meanwhile, it is difficult for them to use a few parameters to extract the spatial information of ORSIs due to their more complex background and less spatial feature information for detected objects in ORSIs.

In order to more efficiently highlight the feature related to the detection task in ORSIs with a better robustness, a novel channel attention mechanism named a DCA Block was proposed to enhance the representation of the object feature in ORSIs through combing global and local channel information with a slight inference time increase.

3. Proposed Methods

3.1. Method Overview

The structure of YOLOv4 is given in Figure 2. YOLOv4 consists of a backbone, neck and prediction. YOLOv4 is established for NSIs and not practical enough to be adopted in ORSIs directly. Specifically, the backbone CSP DarkNet in YOLOv4 utilizes the CSP Block [37] as the feature extraction module and performs well in detection accuracy, but its model complexity and computational burden can be further reduced to improve its

deployability and inference speed for ORSIs. The neck PANet [41] employed in YOLOv4 can strengthen the integration of a shallow and deep feature map, but its CBL×5 modules will easily cause the problem of feature loss, which is not conducive to information transmission for objects in ORSIs. Moreover, attention mechanisms that can enhance the feature representation are not utilized in YOLOv4.

The proposed detector YOLO-DSD based on YOLOv4 is shown in Figure 3. Three new modules are presented to improve the performance of YOLOv4. In the backbone, we developed a DenseRes Block as the main module for a better feature extraction and reduction in computational cost. In the neck, S-CBL×5 was proposed to handle the information loss problem, and the proposed attention mechanism, the DCA Block, was added after each S-CBL×5 module to enhance the representation of features.

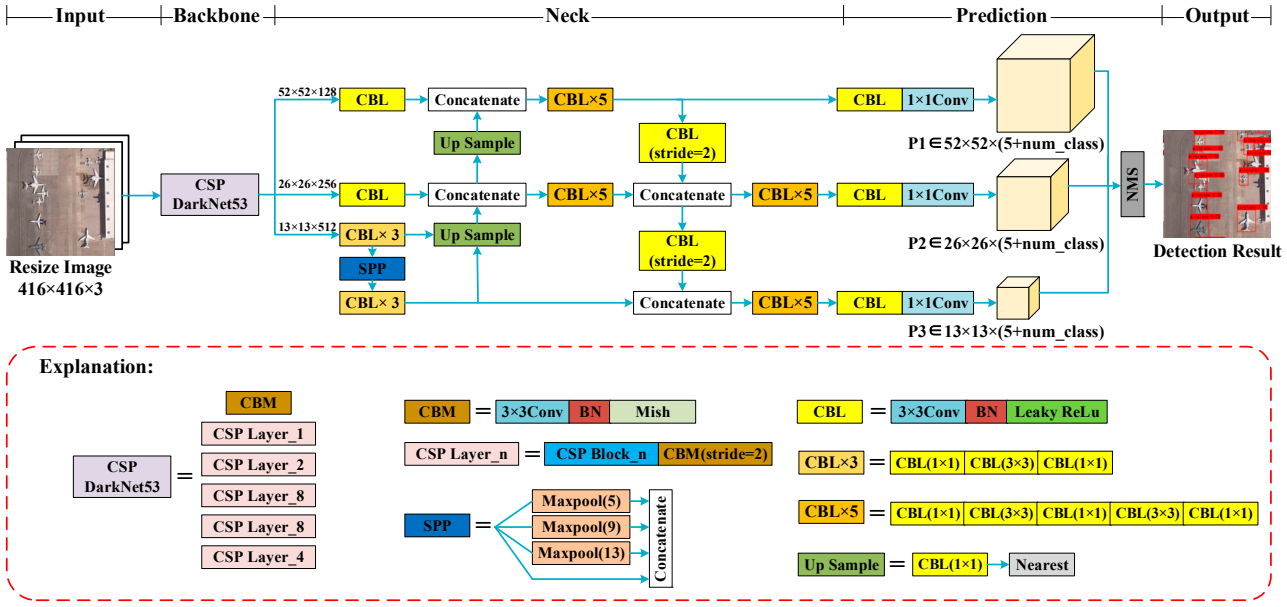

Figure 2. The architecture of YOLOv4.

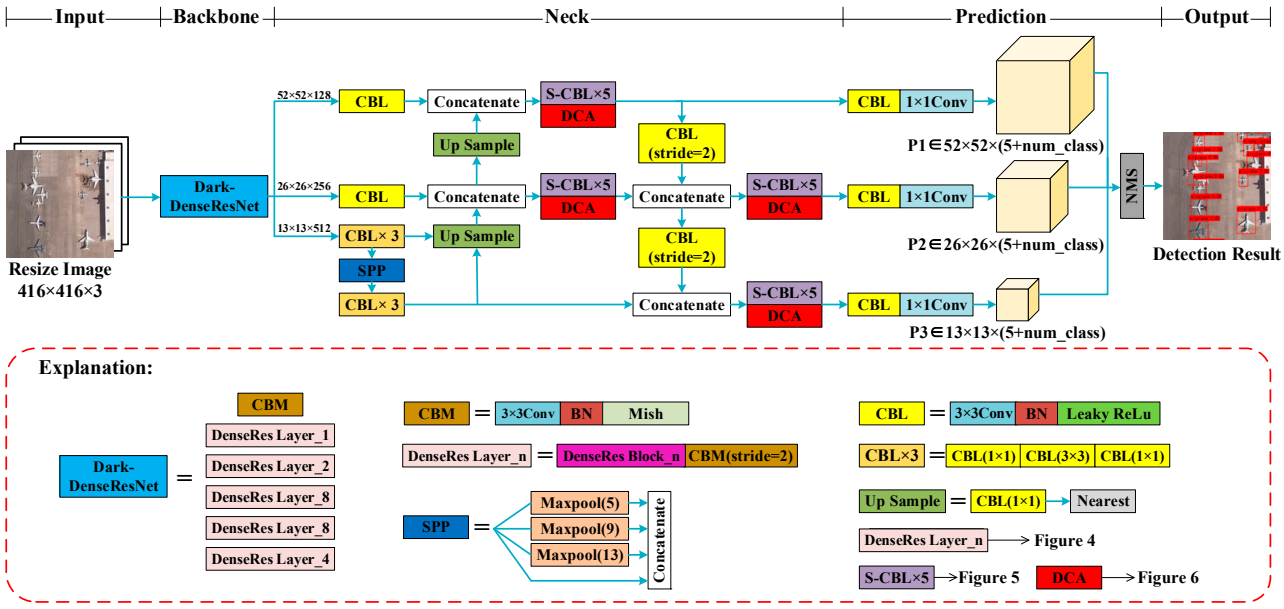

Figure 3. The architecture of YOLO-DSD.

3.2. Improvement in the Backbone

YOLOv4 adopts a CSP Block, shown in Figure 4a, to extract features of images in the backbone. Although the CSP Block performs well in detection accuracy, the structure of the CSP Block containing a parallel convolution operation for reusing the feature of the 'Input' and excessive convolution layers caused by 'Res Unit' takes up a large amount of computing resources and inference time [39]. Aiming at this problem of the CSP Block, we proposed a DenseRes Block, shown in Figure 4b, and employed it in the backbone for feature extraction.

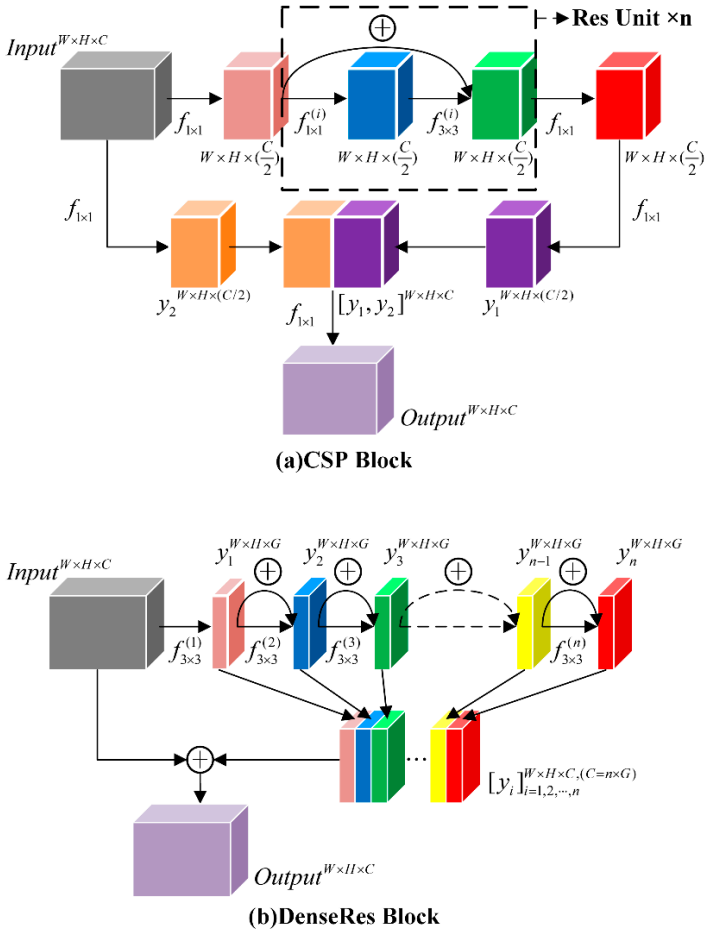

Figure 4. The structure comparison between CSP Block (**a**) and DenseRes Block (**b**).

The DenseRes Block is only composed of several series-connected 3 × 3 convolution operations $f^{(i)}_{3\times3}$ $(i = 1, 2, \cdots, n)$ and short-cut connections based on residual learning. y_i is the output feature map of $f^{(i)}_{3\times3}$. For the feature map $Input \in \mathbb{R}^{W\times H\times C}$, W, H and C indicate the height, width and channel number of the map, respectively. Since the feature of detected objects in ORSIs is easily overwhelmed by that of the background when transmitted, we utilized a feature map with fewer channels as the output of the first convolution operation to compress the '*Input*' to focus on object features and reduce the proportion of background information. Therefore, the feature map $y_1 \in \mathbb{R}^{W\times H\times G}$ was computed by

$$y_1^{W\times H\times G} = f^{(1)}_{3\times3}\left(Input^{W\times H\times C}\right) \quad (1)$$

where $C = n \times G$, $f^{(1)}_{3\times3}$ contains the 3 × 3 convolution layer that compacts the number of channels from C to G, the BN layer and the leaky ReLu activation function. If $n = 1$, the DenseRes Block is the same as the Res Block. When $n > 1$, the DenseRes Block will compress the '*Input*' and make a feature extraction. It was proven in Ref. [39] that the following

operations can effectively reduce the memory access cost and the inference time of the model: (1) the input channel and output channel of the convolution layer should be equal as much as possible; (2) the number of fragmented operators (i.e., the number of individual convolution or parallel operations in one building block) should be reduced. Therefore, $y_j(1 < j \leq n) \in \mathbb{R}^{W \times H \times G}$ could be designed as

$$y_{j(1<j\leq n)}^{W\times H\times G} = y_{j-1}^{W\times H\times G} \oplus f_{3\times 3}^{(j)}\left(y_{j-1}^{W\times H\times G}\right) \tag{2}$$

where $f_{3\times 3}^{(j)}(1 < j \leq n)$ contains the 3×3 convolution layer with the same number G of input and output channels, the BN layer and the leaky ReLu activation function. $\oplus$ indicates the element-wise addition. From the comparison between the CSP Block and the DenseRes Block shown in Figure 4, the output of each 'Res Unit' in the CSP Block will go through two convolution layers with different kernel sizes, whereas that of each 'y_i' in the DenseRes Block only goes through one 3×3 convolution layer. Therefore, the fragment degree can be decreased. Moreover, we used a short-cut based on residual learning to connect $y_j(1 < j \leq n)$ and $y_{j-1}(1 < j \leq n)$ for the problem of feature loss in the process of feature extraction.

In ORSIs, there will be potential semantic relevance between the object and the background [21,38]. For example, cars and airplanes tend to park on land whereas ships tend to sail on the sea, and bridges are built over water whereas overpasses are built over land. In order to make the network better learn high-level semantic relevance, the $Output \in \mathbb{R}^{W \times H \times C}$ was designed as

$$Output^{W\times H\times C} = Input^{W\times H\times C} \oplus [y_i]_{i=1,2,\cdots,n}^{W\times H\times C(C=n\times G)} \tag{3}$$

where $[y_i]_{i=1,2,\cdots,n}^{W\times H\times C(C=n\times G)}$ concatenates $y_1, y_2, \ldots, y_n$ in the channel dimension to a feature map with the same size as $Input \in \mathbb{R}^{W \times H \times C}$. $[y_i]_{i=1,2,\cdots,n}^{W\times H\times C(C=n\times G)}$ possessing more object information was combined with the $Input \in \mathbb{R}^{W \times H \times C}$ holding more background information by element-wise addition directly to improve the detection accuracy. Compared with the CSP Block, such a designed structure in the DenseRes Block not only reuses the feature of '*Input*' but also omits a parallel convolution operation, which can further reduce the degree of the fragment in the backbone.

The DenseRes Block was utilized in order to replace the original module, the CSP Block, in the backbone. The architecture and complexity of the restructured backbone, named DarkNet-DenseRes, is shown in Table A1, Appendix A.

3.3. Improvement in the Neck

YOLOv4 uses the feature pyramid structure of PANet in the neck to fuse feature maps of different levels and extract a feature, which performs well in object detection in natural scenes. However, the feature information of objects in ORSIs is usually far less obvious than that of objects in natural scenes, and information loss caused by excessive convolutional operations in PANet limits the detection performance of the network for the objects in ORSIs. In order to solve this problem, S-CBL×5 was utilized to replace each CBL×5 in the original neck as shown in Figure 3. The structure comparison between CBL×5 and S-CBL×5 is given in Figure 5. S-CBL×5 adds two short-cuts based on CBL×5 and does not add additional parameters and inference time.

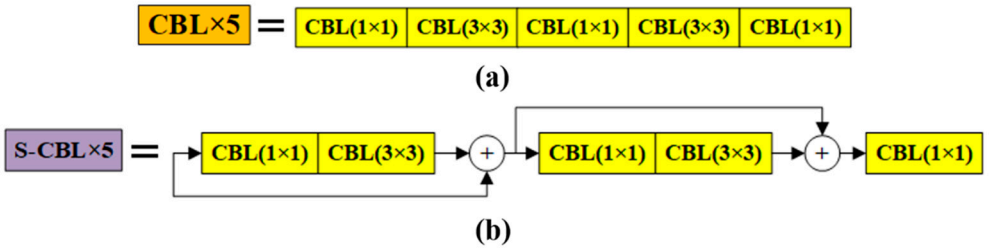

Figure 5. The structure comparison between CBL×5 (**a**) and the proposed S-CBL×5 (**b**).

To highlight significant features related to the detection task, the DCA Block was proposed to optimize the weight distribution of each feature map in the channel dimension by combining the local and global relationship between channels with a slight increase in computations cost and inference time. The structure of the DCA Block is shown in Figure 6.

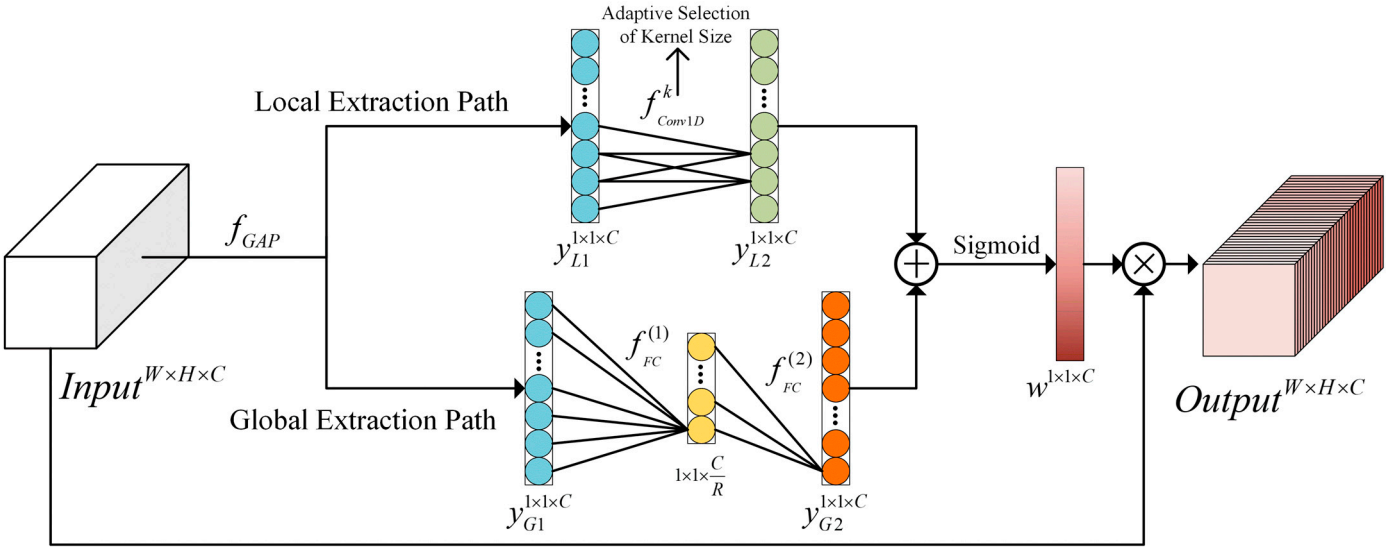

Figure 6. The structure of the proposed DCA Block.

The DCA Block is composed of a 'Local Extraction Path' and 'Global Extraction Path' in parallel. The 'Global Extraction Path' is used to learn the global relationship between channels, whereas the 'Local Extraction Path' is employed to extract the local channel relationship.

Firstly, global average pooling was employed to obtain the integrated information in the space dimension of each channel $y_{L1}^{1\times1\times C}$ and $y_{G1}^{1\times1\times C}$, where $y_{L1}^{1\times1\times C}$ and $y_{G1}^{1\times1\times C}$ indicate the input of the 'Local Extraction Path' and 'Global Extraction Path', respectively, and $y_{L1}^{1\times1\times C} = y_{G1}^{1\times1\times C}$.

Secondly, $y_{L2}^{1\times1\times C}$ in the 'Local Extraction Path' could be computed by

$$y_{L2}^{1\times1\times C} = f_{Conv1D}^{k}\left(y_{L1}^{1\times1\times C}\right) \tag{4}$$

$$k = \frac{\log_2 C + 1}{2} \tag{5}$$

where f_{Conv1D}^{k} represents the 1-dimension convolution layer. Since each feature map has a different number of channels and the kernel size of the convolution layer is proportional to the number of the channels [43], the mapping between its kernel size (k) and the number of input channels (C) is given in Equation (5). f_{Conv1D}^{k} could adaptively select the kernel size according to non-linearly mapping Equation (5); thus, it can extract the local relationship between covered channels more effectively than the convolution layer with a hand-given convolution kernel size.

At the same time, two full connection layers were used as a bottleneck in the 'Global Extraction Path' to build the global relationship of each channel:

$$y_{G2}^{1\times1\times C} = f_{FC}^{(2)}\left(f_{FC}^{(1)}(y_{G1}^{1\times1\times C})\right) \tag{6}$$

where $f_{FC}^{(1)}$ is the first full connection layer that compresses the channel number from C to C/R, and $f_{FC}^{(2)}$ is the second full connection layer that extends the channel number from C/R to C. The value of the zoom factor R that could reduce the complexity of the structure was set to 32 according to the experimental results in Section 4.4.1. The structure of the 'Global Extraction Path' with two full connection layers has a stronger non-linearity and can fit better with the complex global relationship between each channel.

Thirdly, the output of the 'Global Extraction Path' and 'Local Extraction Path' were combined by element-wise addition, and the sigmoid function was applied to generate the weight $w \in \mathbb{R}^{1\times1\times C}$. Finally, the output of the DCA Block was calculated as:

$$w^{1\times1\times C} = Sigmoid\left(y_{G3}^{1\times1\times C} \oplus y_{L2}^{1\times1\times C}\right) \tag{7}$$

$$Output^{W\times H\times C} = w^{1\times1\times C} \otimes Input^{W\times H\times C} \tag{8}$$

where $\otimes$ represents the operation of the element-wise product. As shown in Figure 3, we added the proposed DCA Block after each S-CBL×5 to generate an improved PANet (shown in Figure 7) with a structure that is more suitable for optical remote sensing object detection and has a nearly equal computational cost compared to the original structure.

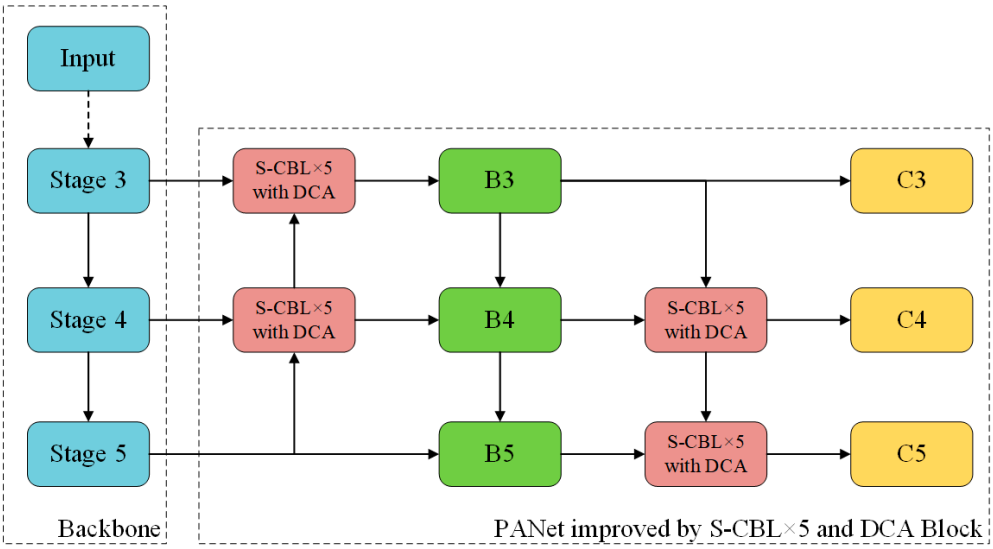

Figure 7. The structure of the improved PANet.

3.4. Prediction

Decoding and obtaining the detection result were processed in the prediction. As shown in Figure 3, each output of the neck went through with a CBL module and a 1 × 1 convolution layer, and three feature maps, $P_1 \in \mathbb{R}^{52\times52\times num_class}$, $P_2 \in \mathbb{R}^{26\times26\times num_class}$ and $P_3 \in \mathbb{R}^{13\times13\times num_class}$, were generated. Then, as shown in Figure 8, P_1, P_2 and P_3 were mapped back to the original image and the image was divided into 52 × 52, 26 × 26 and 13 × 13 sizes of grids. Each grid corresponding to a feature map contains the information of three anchors. In each anchor, (x, y) and (w, h) are the offset coefficient and size coefficient, respectively, Cf is the confidence of the grid containing the object and $C_1, C_2, C_3, \ldots, C_n$ are the confidence of each object class, respectively.

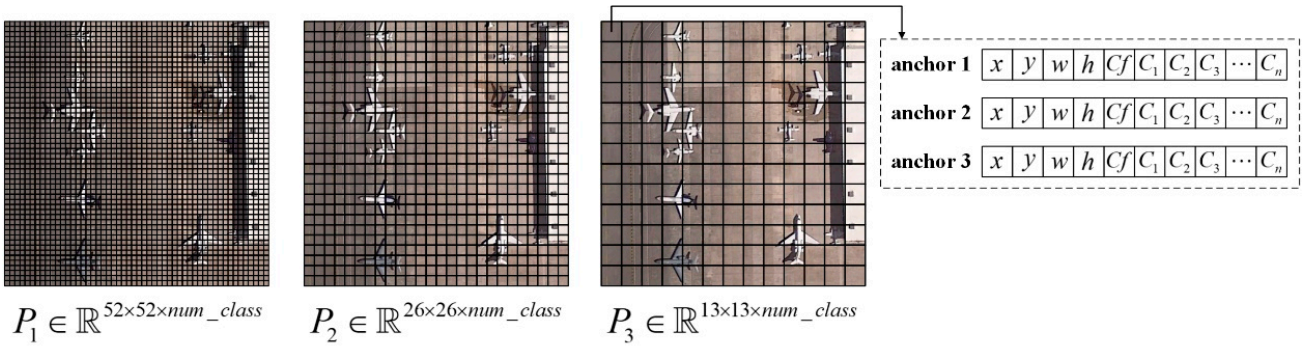

Figure 8. The image is divided into 52 × 52, 26 × 26 and 13 × 13 by P_1, P_2 and P_3, respectively.

Then, each grid generated three bounding boxes according to the information combined with anchors, and the process of converting the anchor to the bounding box is illustrated in Figure 9. (C_x, C_y) are the upper left corner position of the current grid and the center of each grid anchor. ($\sigma(x)$, $\sigma(y)$) is the offset of the bounding box relative to the anchor. The width b_w and height b_h of the bounding box were obtained through multiplying the width p_w and height p_h of the anchor by scaling factors e^w and e^h, respectively. Finally, the detection results were obtained after redundant bounding boxes were removed through NMS.

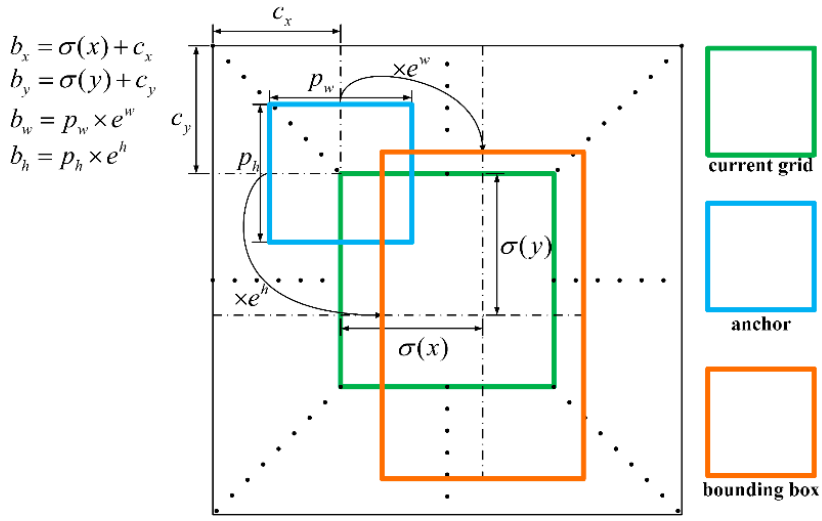

Figure 9. The process of converting anchor to bounding box.

3.5. Loss Function

The loss function of YOLOv4 includes three parts: confidence, classification and bounding box regression loss. YOLOv4 employs the complete intersection over union (*IoU*) loss (*CIoU*) [46], replacing the mean squared error loss adopted in YOLOv3 with the bounding box regression loss. *CIoU* takes the overlap area, center point distance and aspect ratio into consideration simultaneously, and the convergence speed and detection accuracy were improved. *CIoU* introduces a penalty item $\alpha\nu$ based on the distance *IoU* loss to impose the consistency of the aspect ratio for the ground truth (bb^{gt}) and bounding box (bb^b). The loss of *CIoU* can be defined as Equation (9).

$$\begin{aligned} Loss_{CIoU} &= 1 - \left(IoU - \frac{\rho^2\left(b^{gt}, b^b\right)}{c^2} - \alpha\nu \right) \\ \alpha &= \frac{\nu}{1 - IoU + \nu}, \ \nu = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w^b}{h^b}\right)^2 \end{aligned} \tag{9}$$

where b^{gt}, b^{b} are the center of bb^{gt}, bb^{b}, respectively, ρ denotes the Euclidean distance, c represents the diagonal length of the smallest enclosing rectangle covering bb^{gt}, bb^{b}, α is a positive trade-off value and ν means the consistency of the aspect ratio. w^{gt}, w^{b} are the width of the bb^{gt}, bb^{b}, respectively. h^{gt}, h^{b} are the height of the bb^{gt}, bb^{b}, respectively.

CIOU can directly minimize the distance between the bounding box and ground truth and accelerate the model convergence. Previous works [47–49] have proved that *CIOU* can perform better in detecting objects with diverse sizes, which can match well with the characteristics of remote sensing object detection tasks.

4. Experiments and Discussion

In this section, we conduct ablation and comparative experiments on a public optical remote sensing dataset DIOR [2] with 20 categories to validate the proposed YOLO-DSD, considering the accuracy, deployability and speed indictors. Another optical remote sensing dataset RSOD [50] with 4 categories was utilized to further verify the effectiveness of the proposed YOLO-DSD compared with YOLOv4.

4.1. Datasets

4.1.1. DIOR Dataset

DIOR [2] is a large ORSIs dataset that was established in 2020 to develop and validate data-driven methods. It contains 23,463 images and 192,472 objects in total, covering 20 categories in optical remote sensing field. Images in this benchmark dataset have been clipped into 800 × 800 pixels. There are vast scale variations across objects in DIOR because it contains images with spatial resolutions ranging from 0.5 m to 30 m. According to the definition of COCO [15], objects with area of ground truth less than 32 × 32 pixels, between 32 × 32 pixels and 96 × 96 pixels and larger than 96 × 96 pixels are taken as small, middle and large-sized objects, respectively. Each category and the size distribution of objects in DIOR is shown in Figure 10. It can be seen that objects in DIOR possess great size difference and are concentrated in small and middle-sized.

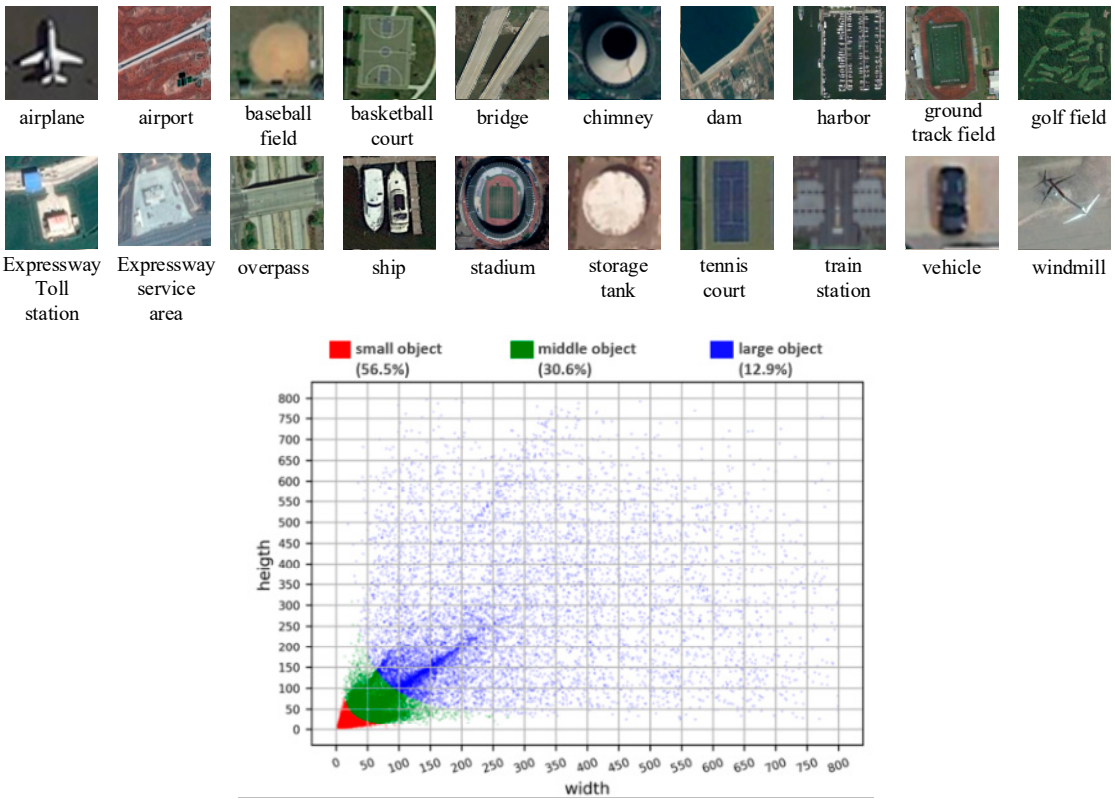

Figure 10. Each category and the size distributions of objects in DIOR.

Moreover, since images in DIOR are carefully collected under various environment conditions, such as different weathers and seasons, these images possess richer variations in viewpoint, background, occlusion, etc. Problems of intra-class diversity and intra-class similarity are more laborious due to the above characteristics. The main difficulties in real-world tasks can be well reflected by DIOR; thus, ablation experiments of YOLO-DSD and comparative experiments with SOTA detectors were conducted in DIOR dataset.

4.1.2. RSOD Dataset

RSOD [50] contains 976 images that have been clipped into approximately 1000 × 1000 pixels, and the spatial resolution of these images ranges from 0.3 m to 3 m. There are 6950 object instances in this dataset in total, covered by 4 common classes in ORSIs, including 4993 aircraft, 1586 oil tanks, 180 overpasses and 191 playgrounds. Each instance of classes is shown in Figure 11.

In addition, instances in RSOD dataset are under various scenes, including urban, grasslands, mountains, lakes, airport, etc. Although the scale of RSOD is not as large as that of DIOR, the characteristics of images in optical remote sensing object detection task can also be reflected by RSOD dataset. Therefore, we further analyzed the effectiveness of YOLO-DSD compared with YOLOv4 in RSOD dataset.



Figure 11. Each category of objects in RSOD.

4.2. Evaluation Indicator

Detectors in this study were analyzed from three perspectives, including detection accuracy, deployability and speed. The evaluation indicators of each performance are shown in Table 1. The higher the mAP and FPS, but the lower Params and Flops, the better the detector.

Table 1. The evaluation indicators.

Indicator Class	Indicator	Description
Accuracy	$mAP^{0.5}$ (%)	Average precision when IOU = 0.5. It is the most used indicator in remote sensing object detection.
	$mAP^{0.5:0.95}$ (%)	Mean values of mAPs under each IOU, which are taken at an interval of 0.05 between 0.5 and 0.95.
	mAP^{S}, mAP^{M}, mAP^{L} (%)	The $mAP^{0.5:0.95}$ of small, middle and large-sized object defined in MS COCO.
Deployability	Params	Number of detector parameters.
	Flops	Floating point operations.
Speed	FPS (img/s)	Frames transmitted per second.

4.3. Experiment Setting

In this study, the deep learning framework PyTorch1.7.1 was utilized to implement all of the detectors in this study. The experimental environment was ubuntu18.04, CUDA11.1, CUDNN8.0.5 and NVIDIA GeForce RTX 3080. In order to ensure enough training samples and to make the test set reflect the characteristics of each dataset well, training and test sets in DIOR were split by 1:1, whereas those in RSOD were split by 4:1 randomly. A total of 90% of the training set was utilized for training detectors, and 10% was used for monitoring to avoid overfitting. The input size and batch size of detectors was set to

416 × 416 and 7, respectively. Adam optimizer was employed to update the parameters, with a weight decay of 2×10^{-4}. The relationship between learning rate and epoch is shown in Figure 12. For anchor-based detectors, K-means was utilized to optimize the size of anchors before training.

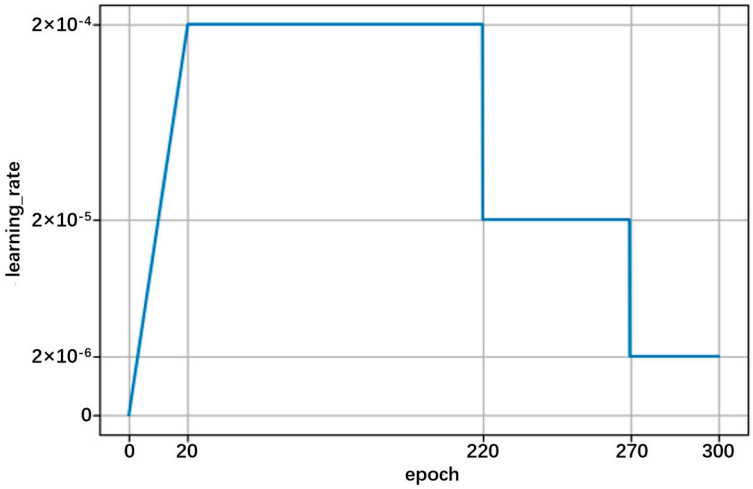

Figure 12. The relationship between learning rate and epoch.

4.4. Experiment Results and Discussion in DIOR Dataset

4.4.1. Ablation Experiment

Ablation experiments were conducted to verify the effectiveness of each improved module in YOLO-DSD, and the results are shown in Table 2. The detector improved with the DenseRes Block reduces Params by 23.9% ($\frac{48.81-64.17}{64.17} \times 100\%$) and Flops by 29.7% ($\frac{21.12-30.07}{30.07} \times 100\%$), and increases FPS by 63.4% ($\frac{65.7-40.2}{40.2} \times 100\%$), while achieving a 0.2% higher $mAP^{0.5}$ and almost the same $mAP^{0.5:0.95}$ compared with YOLOv4 as the baseline. The detector improved by S-CBL×5 in the neck based on "+DenseRes Block" is beneficial for $mAP^{0.5}$ and $mAP^{0.5:0.95}$, which are brought about by the increase in mAP^{M} and mAP^{L} without affecting the deployability and inference speed. However, the mAP^{S} slightly decreased by 0.3% because the short-cut utilized in S-CBL×5 strengthened the transmitting of the feature, and thus introduced background features additionally, which attenuated the representation of the feature for small-sized objects. The detector further improved by the DCA Block achieved a significant increase in mAP due to the enhancement of feature expression, and made up for the loss of mAP^{S} caused by the short-cut with the same Params and Flops, while the FPS was only slightly reduced by 5.3 img/s.

In summary, YOLO-DSD outperforms YOLOv4 both in the detection accuracy, deployability and speed evaluation indicator. YOLO-DSD based on YOLOv4 increases the commonly used indicator $mAP^{0.5}$ by 1.7% and the more rigorous indicator $mAP^{0.5:0.95}$ by 0.9%. Specifically, YOLO-DSD has a greater advantage in mAP^{M} and mAP^{L}, while it achieves a similar and competitive mAP^{S} compared with YOLOv4. In terms of deployability performance, the Params and Flops of YOLO-DSD decreased by 23.9% and 29.7% more than those of YOLOv4, respectively. YOLO-DSD also performs well in inference speed: it is 50.2% faster than YOLOv4 in FPS.

Table 2. The ablation results of YOLO-DSD.

Detectors	Params	Flops	FPS	$mAP^{0.5}$	$mAP^{0.5:0.95}$	mAP^{S}	mAP^{M}	mAP^{L}
YOLOv4(Baseline)	64.17 M	30.07 G	40.2	71.3	39.1	10.1	30.2	55.1
+DenseRes Block	48.81 M	21.12 G	65.7	71.5	38.8	9.4	30.4	54.9
+S-CBL×5	48.81 M	21.12 G	65.7	71.9	39.2	9.1	30.9	55.7
+DCA(YOLO-DSD)	48.81 M	21.12 G	60.4	73.0	40.0	9.6	31.6	56.4

We further analyzed the performance of the DenseRes Block. The ablation results of the DenseRes Block are shown in Table 3. The structure of the DenseRes Block in each detector is shown in Figure 13. Model 1 is the detector improved by the DenseRes Block without the structure of the 'Short-cut' and 'Combine'. 'Short-cut' and 'Combine' are introduced to the DenseRes Block in Model 2 and Model 3, respectively. Model 4 utilizes the complete DenseRes Block to improve the backbone of YOLOv4. From the comparison between Model 1 and Model 2, the 'Short-cut' introduced to DenseRes Block for the mitigation of feature loss can improve the mAP of objects in each size. After adding the 'Combine' to DenseRes Block, Model 3 performs better on the middle and large-sized object, while the mAP^S decreases slightly by 0.1%. The possible reason for this is that the feature of the middle and large-sized object is obvious enough to build high-level semantic relevance with the background feature, while the feature of the small object is not obvious enough and thus it is easy for it to be overwhelmed. Model 4 improved by the complete DenseRes Block achieves the highest mAP and a significant increase in mAP^S, mAP^M and mAP^L. It is probable that, on the basis of the 'Short-cut', the feature of each size object can be better retained when transmitting in the DenseRes Block, and can thus benefit the building of high-level semantic relevance with a background feature through 'Combine'.

Table 3. The ablation results of DenseRes Block.

Detectors	DenseRes Block		FPS	$mAP^{0.5}$	$mAP^{0.5:0.95}$	mAP^S	mAP^M	mAP^L
	+Short-Cut*	+Combine*						
Model 1			65.7	70.8	38.2	9.1	29.9	54.4
Model 2	√		65.7	71.3	38.7	9.5	30.3	54.7
Model 3		√	65.7	71.2	38.5	9.0	30.2	54.9
Model 4	√	√	65.7	71.5	38.8	9.4	30.4	54.9

Note: 'Short-cut*' indicates a short-cut to connect y_i $(1 < i \leq n)$ and $y_{(i-1)}$ $(1 < i \leq n)$ in DenseRes Block; 'Combine*' means $[y_i]$ $(1 \leq i \leq n) \oplus$ *Input* in DenseRes Block.

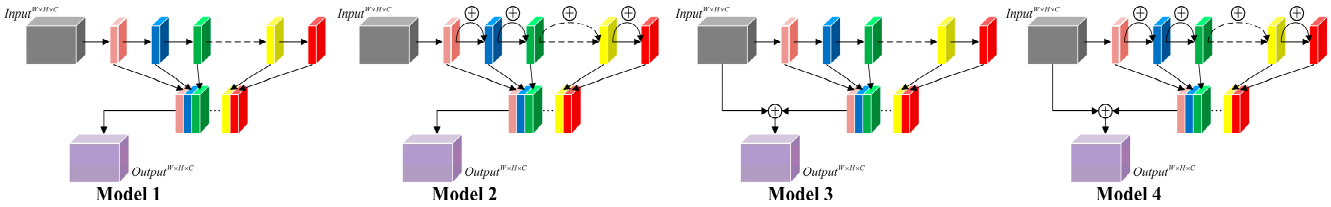

Figure 13. The structure of DenseRes Block in each detector in Table 3.

The experimental results of DCA module ablation are shown in Tables 4 and 5. Table 4 shows the influence of scaling factor R on the performance of the DCA Block. The results show that, when R = 32, DCA can achieve the best performance. Table 5 exhibits the influence of three different fusion methods shown in Figure 14 on the performance of the DCA Block. The results show that the DCA Block with a different fusion method can effectively improve the detection accuracy. Specifically, compared with DCA in series, DCA in parallel has a more obvious advantage in small and middle-sized objects, while the FPS is slightly reduced by 0.7 img/s. This may be due to the fact that, when employing the same number of operation layers in one building block, although the structure designed in parallel has a higher fragment, it can keep the integrity of the feature better compared with that in series. For the proposed DCA Block, which has a small structure complexity, utilizing the structure in parallel makes it perform better in the enhancement of feature expression without an obvious sacrifice of inference time.

Table 4. Results of different zoom factor 'R's in DCA Block.

Detectors	DCA Block	FPS	$mAP^{0.5}$	$mAP^{0.5:0.95}$	mAP^{S}	mAP^{M}	mAP^{L}
	Zoom Factor 'R'						
Model 1	R = 1	60.3	72.4	39.3	9.2	31.1	55.8
Model 2	R = 2	60.3	72.2	39.2	9.1	30.3	55.9
Model 3	R = 4	60.3	71.8	39.0	9.2	30.1	55.9
Model 4	R = 8	60.4	72.8	39.8	9.7	31.1	56.4
Model 5	R = 16	60.4	72.3	39.5	9.4	31.2	56.2
Model 6	R = 32	60.4	73.0	40.0	9.6	31.6	56.4
Model 7	R = 64	60.4	72.1	39.6	9.2	31.4	56.1

Table 5. Results of different fusion forms in DCA Block.

Detectors	DCA Block	FPS	$mAP^{0.5}$	$mAP^{0.5:0.95}$	mAP^{S}	mAP^{M}	mAP^{L}
	Fusion Form						
Model 1	'Global Path' + 'Local Path' (In series)	61.1	72.2	39.7	9.3	31.2	56.5
Model 2	'Local Path' + 'Global Path' (In series)	61.1	72.0	39.5	9.3	30.5	56.6
Model 3	'Global Path' + 'Local Path' (In parallel)	60.4	73.0	40.0	9.6	31.6	56.4

Note: 'Global Path' and 'Local Path' refer to 'Global Extraction Path' and 'Local Extraction Path', respectively.

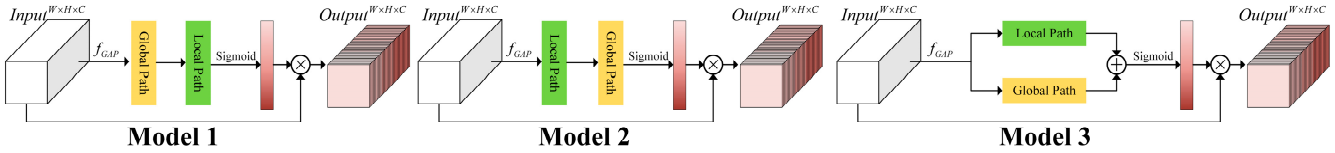

Figure 14. The structure of DCA Block in each detector in Table 5.

4.4.2. Comparative Experiment

Four experiments were conducted in this study to verify the superiority of the proposed method. (1) ResNet50 [34], VGG16 [33] and the backbones that were established based on the CSP DarkNet framework with different feature extraction modules, including the Res Block [34], ResNeXt Block [35], Res2 Block [36], Dense Block [27], CSP Block [37] and DenseRes Block, were compared. (2) A comparison of different neck structures, including FPN [16], BFPN [40], PANet [41], S-PANet (PANet improved with the proposed S-CBL×5) and none (without feature pyramid structure), was conducted. (3) The performance of different attention mechanisms, including the SE Block [42], ECA Block [43], CA Block [44], CBAM Block [45] and DCA Block(R = 32), was compared and analyzed. (4) YOLO-DSD was compared with eight SOTA detectors, including Faster-RCNN, SSD, RetinaNet, YOLOv3, YOLOv4, YOLO-Lite (MobileNetV2 [51]—YOLOv4), CenterNet [7] and EfficientDet [9], which have been widely applied in various natural scene visual detection tasks due to their acceptable tradeoff between accuracy, deployability and inference time.

Comparative experiment for different backbones: The performances of the CSP DarkNet, which is improved by the proposed DenseRes Block (DarkNet-DenseRes) and other backbones, are demonstrated in Table 6. Based on the CSP DarkNet framework, the proposed DenseRes Block outperforms the ResNeXt Block and Dense Block in all indicators. Although the $mAP^{0.5}$ and $mAP^{0.5:0.95}$ of DarkNet-DenseRes are slightly lower than those of DarkNet-Res by 0.1% and 1.3%, the Params and Flops of DarkNet-DenseRes are only approximately 1/3 and 1/4 of DarkNet-Res, while the FPS of DarkNet-DenseRes is approximately 1/4 higher than that of DarkNet-Res. Similarly, the $mAP^{0.5}$ and $mAP^{0.5:0.95}$ of DarkNet-DenseRes are 0.9% and 1.1% lower than those of DarkNet-Res2; however, the Params and Flops of DarkNet-DenseRes are only approximately 1/3 and 1/2 of those of DarkNet-Res2, while the interference speed is 2.3 times that of DarkNet-Res2 according to FPS. The superiority of DarkNet-DenseRes compared with CSP DarkNet was analyzed and

proved in ablation experiments. DarkNet-DenseRes also has obvious advantage in all indicators compared with ResNet50. Although DarkNet-DenseRes has a similar accuracy and speed to VGG16, VGG16 has seven times as much Flops than that of DarkNet-DenseRes. Therefore, DarkNet-DenseRes achieves the optimal balance of accuracy, deployability and speed.

Table 6. Results of comparative experiment for different backbones.

Backbone	Params	Flops	FPS	$mAP^{0.5}$	$mAP^{0.5:0.95}$	mAP^{S}	mAP^{M}	mAP^{L}
CSP DarkNet (Baseline)	26.61 M	17.34 G	40.2	71.3	39.1	10.1	30.2	55.1
DarkNet-Res [1]	40.58 M	24.61 G	52.8	71.6	40.1	9.9	31.7	56.1
DarkNet-ResNeXt [2]	20.55 M	12.71 G	39.1	68.4	36.4	8.2	28.3	52.6
DarkNet-Res2 [3]	31.65 M	19.33 G	28.4	72.4	39.9	10.2	31.6	55.4
DarkNet-Dense [4]	14.06 M	8.16 G	50.9	69.6	37.5	7.8	29.7	54.5
DarkNet-DenseRes [5](Ours)	11.26 M	8.42 G	65.7	71.5	38.8	9.4	30.4	54.9
ResNet50	23.51 M	13.41 G	47.4	68.5	36.5	7.8	28.7	53.2
VGG16	17.07 M	54.64 G	70.1	71.1	38.9	9.9	29.8	55.2

Note: [1, 2, 3, 4, 5] means CSP DarkNet utilizes Res Block, ResNeXt Block, Res2 Block, Dense Block and DenseRes Block as the main feature extraction module, respectively.

Comparative experiment for different necks: Table 7 shows the performance of each neck structure that was tested by applying a no-feature pyramid structure (None), FPN, BFPN, PANet (Baseline) and S-PANet to the modified YOLOv4, with the DenseRes Block in the backbone. 'None' has the lowest Params (18.83 M) and Flops (4.89 G) and the highest FPS (85.5 img/s), but it does not perform well in detection accuracy, and, in particular, its mAP^{S} is only 8.1%, whereas that of the other four necks ranges from 9.1% to 9.5%. Therefore, the feature pyramid structure is vital for detection accuracy and, in particular, for small size objects, which occupy more than 50% in DIOR. Although FPN and BFPN are slightly better than PANet in deployability and inference speed, they have more than a 2.6% inferiority in mAP of middle and large-sized objects, which, in total, account for approximately 50% of objects in DIOR. It was proven that the structure of PANet is important to the detection accuracy in YOLOv4 for ORSIs. PANet and S-PANet have almost the same Params, Flops and FPS, but our S-PANet performs better than PANet in $mAP^{0.5}$ and $mAP^{0.5:0.95}$. In conclusion, S-PANet is more suitable for optical remote sensing object detection than other necks.

Table 7. Results of comparative experiment for different necks.

Neck	Params	Flops	FPS	$mAP^{0.5}$	$mAP^{0.5:0.95}$	mAP^{S}	mAP^{M}	mAP^{L}
None	18.83 M	4.89 G	85.5	68.3	35.5	8.1	27.4	51.6
FPN	27.22 M	8.50 G	71.8	69.1	36.0	9.2	27.2	51.4
BFPN	35.68 M	10.84 G	68.1	69.9	36.8	9.5	27.8	52.1
PANet (Baseline)	37.55 M	12.73 G	65.7	71.5	38.8	9.4	30.4	54.9
S-PANet(ours)	37.55 M	12.73 G	65.7	71.9	39.2	9.1	30.9	55.7

Comparative experiments for different attention mechanisms: Taking modified YOLOv4 with the DenseRes Block in the backbone and S-PANet in the neck as the baseline (None), the indicator values of different attention mechanisms are exhibited and compared in Table 8. The CA Block and CBAM Block containing the spatial attention mechanism fail to improve the detection accuracy, and the FPS decreases significantly due to those complex structures. Most channel attention mechanisms, including the SE Block, ECA Block and DCA Block, can improve the detection accuracy. The DCA Block improves the detection accuracy for small, medium and large sizes of objects, and achieves the highest $mAP^{0.5}$ = 73.0% and $mAP^{0.5:0.95}$ = 40.0%, with an increase of 1.1% and 0.8% compared with 'None', respectively, when R = 32, and the FPS only decreases by 5.3 img/s. In the case of the SE Block, $mAP^{0.5}$ and $mAP^{0.5:0.95}$ increases by 0.2% and 0.1%, and the FPS decreases by 3.4 img/s. The ECA Block improves both $mAP^{0.5}$ and $mAP^{0.5:0.95}$ by 0.1%, and decreases

the FPS by 2.8 img/s. Therefore, the proposed DCA Block can achieve the best balance between accuracy and speed.

Table 8. Results of comparative experiment for different attention mechanisms.

Attention Mechanism	Params	Flops	FPS	$mAP^{0.5}$	$mAP^{0.5:0.95}$	mAP^{S}	mAP^{M}	mAP^{L}
None (Baseline)	0	0	65.7	71.9	39.2	9.1	30.9	55.7
CA	42.36 K	1126.41 K	57.2	71.6	39.0	9.1	30.4	55.9
CBAM	102.79 K	516.59 K	52.8	70.6	38.1	8.4	29.3	55.7
SE	51.20 K	51.272 K	62.3	72.1	39.3	9.0	30.8	56.5
ECA	0.02 K	0.02 K	62.9	72.0	39.3	9.2	30.7	56.1
DCA (R = 32)	51.22 K	830.24 K	60.4	73.0	40.0	9.6	31.6	56.4

Comparative experiments for different detectors: The performances of the proposed YOLO-DSD and eight SOTA detectors are demonstrated in Table 9. RetinaNet and EfficientDet have a better deployability than YOLO-DSD, but their detection accuracy, especially for small-sized objects and speed, are far behind that of YOLO-DSD, so this hinders the application of these detectors in optical remote sensing object detection. The large Flops of SSD and Faster-RCNN require a huge amount of computing resources, which greatly increases the difficulty in deploying them on edge devices. Although the Params and Flops of CenterNet are 67% and 69% that of YOLO-DSD, and the FPS is 46% faster, the detection accuracy of CenterNet is significantly lower than that of YOLO-DSD ($mAP^{0.5:0.95}$:35.8% vs. 40.0%), and the mAP^{S} is only 62.5% that of YOLO-DSD. YOLO-Lite has an obvious disadvantage in detection accuracy for small and large-sized objects, even though it has a better deployability compared with YOLO-DSD. The inference speed of YOLOv3 is nearly the same as that of YOLO-DSD, but the deployability and detection accuracy of YOLOv3 are obviously inferior to that of YOLO-DSD. The superiority of YOLO-DSD compared with YOLOv4 was analyzed and proved in ablation experiments. Therefore, YOLO-DSD outperforms other SOTA detectors in the balance of accuracy, deployability and speed.

Table 9. Results of comparative experiment for different detectors.

Detector	Params	Flops	FPS	$mAP^{0.5}$	$mAP^{0.5:0.95}$	mAP^{S}	mAP^{M}	mAP^{L}
RetinaNet	36.72 M	17.24 G	44.6	62.7	37.6	4.8	30.9	57.5
EfficientDet	3.60 M	1.30 G	18.1	50.4	29.4	2.4	24.9	46.0
SSD	26.15 M	59.59 G	87.3	61.9	37.8	4.6	31.0	58.2
CenterNet	32.67 M	14.62 G	88.5	61.4	35.8	6.0	27.3	55.3
Faster-RCNN	28.47 M	364.14 G	21.9	56.1	31.8	2.8	23.7	53.2
YOLO-Lite	10.48 M	3.89 G	54.1	64.5	33.1	6.5	26.1	48.7
YOLOv3	61.63 M	32.83 G	61.4	69.2	34.7	7.8	27.4	49.8
YOLOv4	64.17 M	30.07 G	40.2	71.3	39.1	10.1	30.2	55.1
YOLO-DSD (ours)	48.81 M	21.12 G	60.4	73.0	40.0	9.6	31.6	56.4

Figures 15–17 exhibit the detection performance of Faster-RCNN, CenterNet, YOLOv4 and YOLO-DSD on DIOR. The detection result of the small-sized instance in Figure 15 indicates that both Faster-RCNN and CenterNet obviously miss detection. Although YOLOv4 could completely detect airplanes, it incorrectly detected a storage tank. Our YOLO-DSD can correctly detect all airplanes without any false detection. Figure 16 presents the detection results of an instance in the complex urban background. We can see that Faster-RCNN only detects one ground track field, and that CenterNet misses two bridges and two ground track fields and misdetects an overpass. YOLOv4 misses one bridge and one ground track field, whereas YOLO-DSD detects all objects correctly. The detection results of instances in a complex suburban background are given in Figure 17. It can be seen that Faster-RCNN detects only one Expressway-Service-Area, CenterNet has two false detections of an overpass and windmill, YOLOv4 detects two Expressway-Service-

Areas as one, and YOLO-DSD correctly detects all objects. The above instances verify that YOLO-DSD can handle object detection under different complex backgrounds well.

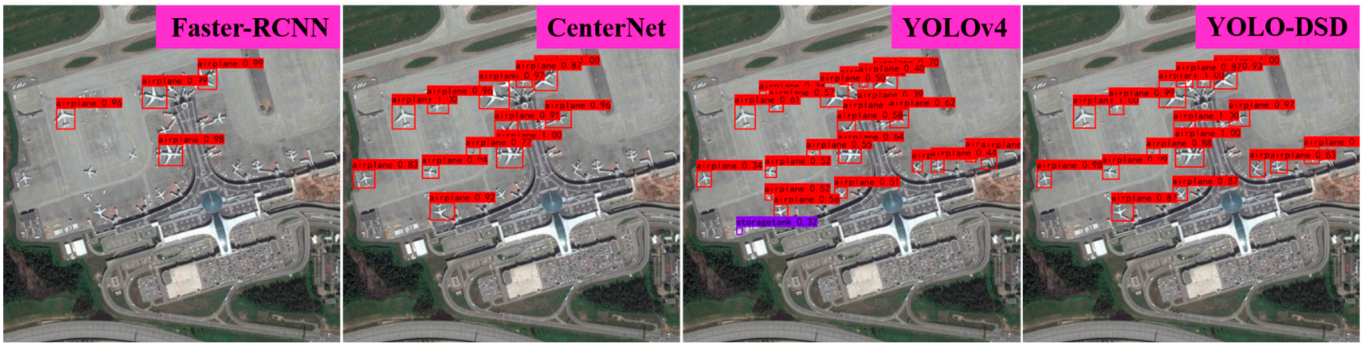

Figure 15. The detection result of small-sized instance.

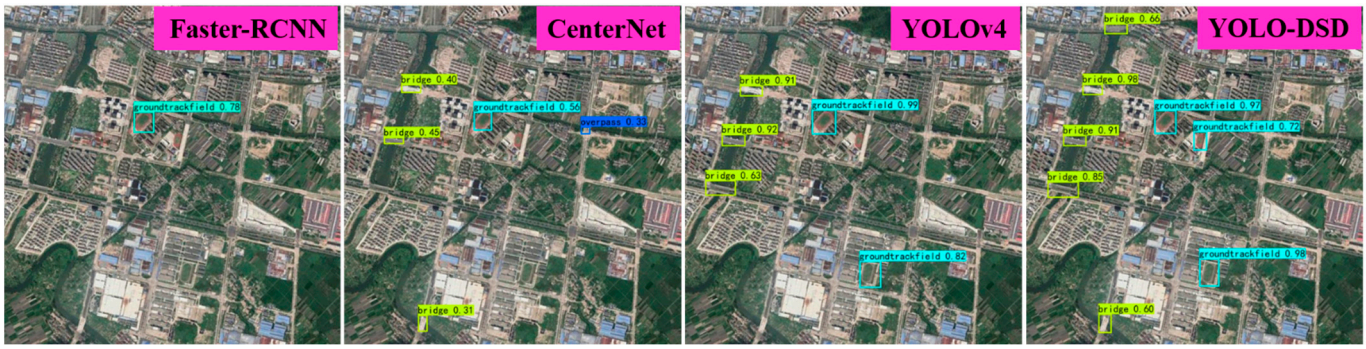

Figure 16. The detection result of instance in complex suburban background.

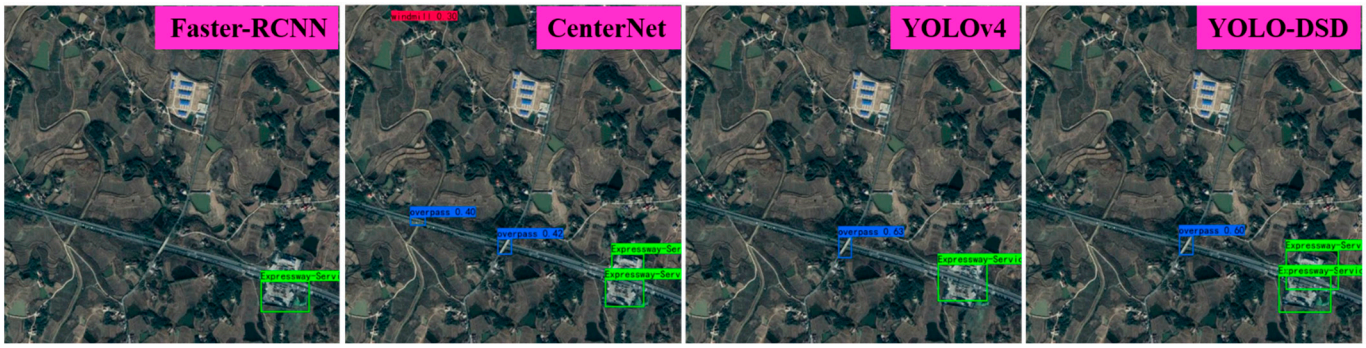

Figure 17. The detection result of instance in complex urban background.

The precision–recall curves and AP (IOU = 0.5) of YOLOv4 and YOLO-DSD in each category are given in Figure 18 for a better illustration of the difference in detection accuracy. It can be seen that YOLO-DSD detects better than YOLOv4 in 11 categories, including airplane, airport, baseball field, chimney, dam, Expressway-Service-Area, golffield, groundtrackfield, stadium, storagetank and transtation. In particular, the AP of YOLO-DSD in airport, baseballfield, Expressway-Service-Area and groundtrackfield is over 2% higher than that of YOLOv4. The AP of YOLO-DSD in airplane, transtation and stadium significantly increase by 6.63%, 5.21% and 17.02%, respectively. For the other nine categories, YOLO-DSD only slightly decreases by 0.35~1.78% compared with YOLOv4 in AP, but still has a competitive accuracy. Therefore, YOLO-DSD has a better accuracy performance than YOLOv4 in the large-scale ORSIs dataset DIOR in total.

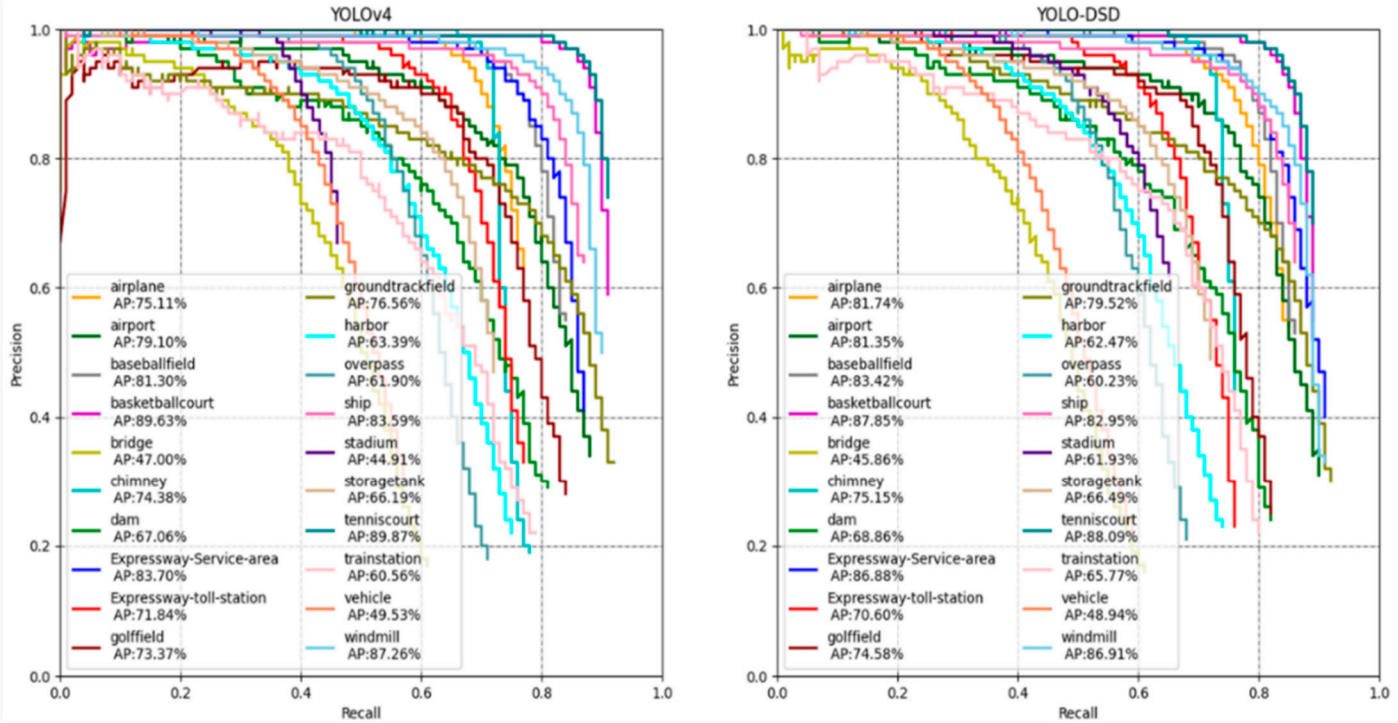

Figure 18. The precision–recall curves and AP (IOU = 0.5) of YOLOv4 and YOLO-DSD in each category.

4.5. Experiment Results and Discussion in RSOD Dataset

In order to further exhibit the superiority of the proposed YOLO-DSD based on YOLOv4 in optical remote sensing object detection application, another comparison experiment between YOLO-DSD and YOLOv4 was conducted in a four-category dataset, RSOD [50], which contained aircraft, oil tank, playground and overpass. The experiment result is shown in Table 10. It can be seen that YOLO-DSD outperforms in accuracy and inference time compared with YOLOv4 under different input sizes, including 416 × 416, 512 × 512 and 608 × 608. Specifically, YOLO-DSD increases $mAP^{0.5}$, $mAP^{0.5:0.95}$ and FPS by 2.6%, 0.8% and 50.2%, respectively, under the input size 416 × 416, while, under the input size 512 × 512, $mAP^{0.5}$, $mAP^{0.5:0.95}$ and FPS are improved by 2.1%, 1.9% and 54.9%, respectively. In terms of the input size 608 × 608, the $mAP^{0.5}$, $mAP^{0.5:0.95}$ and FPS of YOLO-DSD are 1.5%, 1.2% and 59.3% higher than those of YOLOv4.

However, it is noteworthy that the overpass AP of YOLO-DSD is higher than that of YOLOv4 in RSOD, whereas it is the opposite in DIOR. One possible reason for this is that 'bridge' and 'overpass' possess a significant inter-class similarity and thus interfere with the detection performance of YOLO-DSD in these two categories in DIOR. Therefore, how to overcome the inter-class similarity between 'bridge' and 'overpass' for a better detection accuracy while keeping its deployability and inference speed is one of our future works.

Table 10. Results of comparative experiment for YOLOv4 and YOLO-DSD in RSOD.

Detector	Input Size	FPS	$AP^{0.5}$				$mAP^{0.5}$	$mAP^{0.5:0.95}$
			Aircraft	Oil Tank	Playground	Overpass		
YOLOv4	416 × 416	40.2	97.8	95.7	99.4	67.2	90.0	52.1
YOLO-DSD		60.7	98.0	98.2	99.6	74.4	92.6	52.9
YOLOv4	512 × 512	37.1	98.1	97.5	99.5	73.5	92.2	55.1
YOLO-DSD		57.5	98.5	98.6	99.8	80.1	94.3	57.0
YOLOv4	608 × 608	34.9	98.2	98.5	99.9	79.2	94.0	58.5
YOLO-DSD		55.6	99.1	98.9	99.9	84.2	95.5	59.7

5. Conclusions

In this study, a new detector, YOLO-DSD, based on YOLOv4, was proposed to balance the accuracy, deployability and inference time for remote sensing object detection. Three main improvements were utilized in YOLO-DSD, including the DenseRes Block, S-CBL×5 and DCA Block. Firstly, the DenseRes Block improves the backbone, which can better compress and extract the object feature with a high accuracy but less computational consumption. Secondly, S-CBL×5 introduced in the neck can mitigate feature loss without increasing the consumption and inference time. Finally, a new channel attention mechanism, the DCA Block, added to S-CBL×5 better highlights the important features in the channel dimension.

Experiments on a large dataset, DIOR, were conducted to analyze the detection performance from the accuracy (mAP), deployability (Params and Flops) and speed (FPS). The results of the experiments indicate that the proposed DenseRes Block is superior to other feature extraction modules, such as the Res Block, ResNeXt Block, Res2 Block, Dense Block and CSP Block. Moreover, S-CBL×5 performs better than currently widely used FPN, BFPN and PANet. In addition, the proposed DCA Block outperforms other attention mechanisms, including the SE Block, ECA Block, CA Block and CBAM Block. Compared with YOLOv4, YOLO-DSD reduces Params by 23.9% and Flops by 29.7%, but increases FPS by 50.2%, while $mAP^{0.5}$ and $mAP^{0.5:0.95}$ increased from 71.3% to 73.0% and 39.1% to 40.0%, respectively. Compared with other SOTA detectors, including Faster-RCNN, SSD, RetinaNet, YOLOv3, YOLOv4, CenterNet, YOLO-Lite and EfficientDet, YOLO-DSD achieves the optimal balance of accuracy, deployability and inference time. In terms of the RSOD dataset, compared with YOLOv4, YOLO-DSD achieves 1.5~2.6%, 0.8~1.2% and 50.2~59.3% increases in $mAP^{0.5}$, $mAP^{0.5:0.95}$ and FPS under different input sizes, including 416 × 416, 512 × 512 and 608 × 608.

However, YOLO-DSD has a limitation in processing a serious inter-class similarity, such as 'bridge' and 'overpass', compared with YOLOv4. In order to further improve the performance of the proposed detector, we will try to combine depthwise separable convolution [52] with the proposed DenseRes Block for a better feature extraction and deployability reduction. Moreover, other non-consumption methods, such as image preprocessing and anchor optimization, will be considered to improve the detector.

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

Funding: This research was funded by Natural Science Foundation of Guangdong, China with grant number 2021A1515012395, and was supported by earmarked fund for China Agriculture Research System, grant number CARS-17.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data used during the study have been uploaded at: https://gcheng-nwpu.github.io/#Datasets (last accessed on 27 July 2022) and https://github.com/RSIA-LIESMARS-WHU/RSOD-Dataset- (last accessed on 27 July 2022).

Acknowledgments: We gratefully appreciate the editor and anonymous reviewers for their efforts and constructive comments, which have greatly improved the technical quality and presentation of this study.

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

Appendix A

Table A1. The architecture and complexity of DarkNet-DenseRes.

Stage	Output Size	Operation	Number	Params	Flops
CBM	416 × 416 × 32	Conv-BN-Mish (k = 3 × 3, c = 32, s = 1) *	1	928	166,133,760
DenseRes Layer_1	208 × 208 × 64	Conv-BN-Mish (k = 3 × 3, c = 64, s = 2)	1	18,560	805,748,736
	208 × 208 × 64	Conv-BN- Leaky ReLu (k = 3 × 3, c = 64, s = 1)	1	36,992	1,603,190,784
	208 × 208 × 64	Concatenation	1	/	/
DenseRes Layer_2	104 × 104 × 128	Conv-BN-Mish (k = 3 × 3, c = 128, s = 2)	1	73,984	801,595,392
	104 × 104 × 64	Conv-BN- Leaky ReLu (k = 3×3, c = 64, s = 1)	2	110,848	1,200,316,416
	104 × 104 × 128	Concatenation	1	/	/
DenseRes Layer_3	52 × 52 × 256	Conv-BN-Mish (k = 3 × 3, c = 256, s = 2)	1	295,424	799,518,720
	52 × 52 × 32	Conv-BN- Leaky ReLu (k = 3 × 3, c = 32, s = 1)	8	138,752	375,861,632
	52 × 52 × 256	Concatenation	1	/	/
DenseRes Layer_4	26 × 26 × 512	Conv-BN-Mish (k = 3 × 3, c = 512, s = 2)	1	1,180,672	798,480,384
	26 × 26 × 64	Conv-BN- Leaky ReLu (k = 3 × 3, c = 64, s = 1)	8	553,984	374,839,296
	26 × 26 × 512	Concatenation	1	/	/
DenseRes Layer_5	13 × 13 × 1024	Conv-BN-Mish (k = 3 × 3, c = 1024, s = 2)	1	4,720,640	797,961,216
	13 × 13 × 256	Conv-BN- Leaky ReLu (k = 3 × 3, c = 256, s = 1)	4	4,130,816	698,280,960
	13 × 13 × 1024	Concatenation	1	/	/
Total Params		11,261,600			
Total Flops		8,421,927,296			

Note: * k, c, and s mean kernel size, output channels and stride of the convolution layer, respectively.

References

1. Cheng, G.; Han, J.W. A survey on object detection in optical remote sensing images. *ISPRS J. Photogramm* **2016**, *117*, 11–28. [CrossRef]
2. Li, K.; Wan, G.; Cheng, G.; Meng, L.Q.; Han, J.W. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm* **2020**, *159*, 296–307. [CrossRef]
3. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
4. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
5. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
6. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
7. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.
8. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
9. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
10. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
11. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Boston, MA, USA, 7–12 June 2015; pp. 1440–1448.
12. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Processing Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
13. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 2961–2969.
14. 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]
15. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
16. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 2980–2988.

17. Al Ridhawi, I.; Bouachir, O.; Aloqaily, M.; Boukerche, A. Design Guidelines for Cooperative UAV-supported Services and Applications. *ACM Comput. Surv.* **2022**, *54*, 1–35. [CrossRef]
18. Xu, D.Q.; Wu, Y.Q. MRFF-YOLO: A Multi-Receptive Fields Fusion Network for Remote Sensing Target Detection. *Remote Sens.* **2020**, *12*, 3118. [CrossRef]
19. Cheng, G.; Si, Y.J.; Hong, H.L.; Yao, X.W.; Guo, L. Cross-Scale Feature Fusion for Object Detection in Optical Remote Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 431–435. [CrossRef]
20. Yin, W.; Diao, W.; Wang, P.; Gao, X.; Li, Y.; Sun, X. PCAN—Part-based context attention network for thermal power plant detection in remote sensing imagery. *Remote Sens.* **2021**, *13*, 1243. [CrossRef]
21. Yuan, Z.C.; Liu, Z.M.; Zhu, C.B.; Qi, J.; Zhao, D.P. Object Detection in Remote Sensing Images via Multi-Feature Pyramid Network with Receptive Field Block. *Remote Sens.* **2021**, *13*, 862. [CrossRef]
22. Li, Z.L.; Zhao, L.N.; Han, X.; Pan, M.Y. Lightweight Ship Detection Methods Based on YOLOv3 and DenseNet. *Math. Probl. Eng.* **2020**, *2020*, 4813183. [CrossRef]
23. Huyan, L.; Bai, Y.P.; Li, Y.; Jiang, D.M.; Zhang, Y.N.; Zhou, Q.; Wei, J.Y.; Liu, J.N.; Zhang, Y.; Cui, T. A Lightweight Object Detection Framework for Remote Sensing Images. *Remote Sens.* **2021**, *13*, 683. [CrossRef]
24. Lang, L.; Xu, K.; Zhang, Q.; Wang, D. Fast and Accurate Object Detection in Remote Sensing Images Based on Lightweight Deep Neural Network. *Sensors* **2021**, *21*, 5460. [CrossRef]
25. Li, Y.Y.; Mao, H.T.; Liu, R.J.; Pei, X.; Jiao, L.C.; Shang, R.H. A Lightweight Keypoint-Based Oriented Object Detection of Remote Sensing Images. *Remote Sens.* **2021**, *13*, 2459. [CrossRef]
26. Huang, W.; Li, G.Y.; Chen, Q.Q.; Ju, M.; Qu, J.T. CF2PN: A Cross-Scale Feature Fusion Pyramid Network Based Remote Sensing Target Detection. *Remote Sens.* **2021**, *13*, 847. [CrossRef]
27. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
28. Qin, Z.; Li, Z.; Zhang, Z.; Bao, Y.; Yu, G.; Peng, Y.; Sun, J. ThunderNet: Towards real-time generic object detection on mobile devices. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 6718–6727.
29. He, H.; Huang, X.; Song, Y.; Zhang, Z.; Wang, M.; Chen, B.; Yan, G. An insulator self-blast detection method based on YOLOv4 with aerial images. *Energy Rep.* **2022**, *8*, 448–454. [CrossRef]
30. Roy, A.M.; Bose, R.; Bhaduri, J. A fast accurate fine-grain object detection model based on YOLOv4 deep neural network. *Neural Comput. Appl.* **2022**, *34*, 3895–3921. [CrossRef]
31. Song, W.; Fu, C.; Zheng, Y.; Cao, L.; Tie, M.; Sham, C.W. Protection of image ROI using chaos-based encryption and DCNN-based object detection. *Neural Comput. Appl.* **2022**, *34*, 5743–5756. [CrossRef]
32. Gu, Y.; Si, B.J.E. A novel lightweight real-time traffic sign detection integration framework based on YOLOv4. *Entropy* **2022**, *24*, 487. [CrossRef]
33. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
35. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
36. Gao, S.-H.; Cheng, M.-M.; Zhao, K.; Zhang, X.-Y.; Yang, M.-H.; Torr, P. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 652–662. [CrossRef]
37. Wang, C.-Y.; Liao, H.-Y.M.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 13–19 June 2020; pp. 390–391.
38. Xu, C.; Li, C.; Cui, Z.; Zhang, T.; Yang, J. Hierarchical Semantic Propagation for Object Detection in Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4353–4364. [CrossRef]
39. Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Salt Lake City, UT, USA, 18–23 June 2018; pp. 116–131.
40. Zhang, X.; Wan, T.; Wu, Z.; Du, B. Real-time detector design for small targets based on bi-channel feature fusion mechanism. *Appl. Intell.* **2022**, *52*, 2775–2784. [CrossRef]
41. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
42. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
43. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
44. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.

45. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Salt Lake City, UT, USA, 18–23 June 2018; pp. 3–19.
46. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, Seattle, WA, USA, 13–19 June 2020; pp. 12993–13000.
47. Dai, W.; Li, D.; Tang, D.; Jiang, Q.; Wang, D.; Wang, H.; Peng, Y. Deep learning assisted vision inspection of resistance spot welds. *J. Manuf. Processes* **2021**, *62*, 262–274. [CrossRef]
48. Tian, R.; Jia, M. DCC-CenterNet: A rapid detection method for steel surface defects. *Measurement* **2022**, *187*, 110211. [CrossRef]
49. 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.* **2021**, *52*, 8574–8586. [CrossRef]
50. 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]
51. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
52. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.