electronics

MDPI

Article

Analysis of Scale Sensitivity of Ship Detection in an Anchor-Free Deep Learning Framework

Yongxin Jiang [1], Li Huang [2], Zhiyou Zhang [1], Bu Nie [3] and Fan Zhang [4,*]

[1] Department of Navigation, Dalian Naval Academy, Dalian 116018, China
[2] School of Computer Science, Wuhan University, Wuhan 430072, China
[3] The 722th Institute, China State Shipbuilding Corporation, Wuhan 430010, China
[4] School of Navigation, Wuhan University of Technology, Wuhan 430070, China
* Correspondence: michael_zf@whut.edu.cn

Abstract: Ship detection is an important task in sea surveillance. In the past decade, deep learning-based methods have been proposed for ship detection from images and videos. Convolutional features are observed to be very effective in representing ship objects. However, the scales of convolution often lead to different capacities of feature representation. It is unclear how the scale influences the performance of deep learning methods in ship detection. To this end, this paper studies the scale sensitivity of ship detection in an anchor-free deep learning framework. Specifically, we employ the classical CenterNet as the base and analyze the influence of the size, the depth, and the fusion strategy of convolution features on multi-scale ship target detection. Experiments show that, for small targets, the features obtained from the top-down path fusion can improve the detection performance more significantly than that from the bottom-up path fusion; on the contrary, the bottom-up path fusion achieves better detection performance on larger targets.

Keywords: ship detection; multi-scale features; convolutional neural network; object detection; scale sensitivity

Citation: Jiang, Y.; Huang, L.; Zhang, Z.; Nie, B.; Zhang, F. Analysis of Scale Sensitivity of Ship Detection in an Anchor-Free Deep Learning Framework. *Electronics* **2023**, *12*, 38. https://doi.org/10.3390/electronics12010038

Academic Editor: Fernando De la Prieta

Received: 27 October 2022
Revised: 13 December 2022
Accepted: 15 December 2022
Published: 22 December 2022

1. Introduction

Ships are a main tool for the transportation of large commodities such as oil, steel, coal, and so on. Ship collisions over the sea are disastrous, causing both life loss and economic loss. The accurate detection of ships from a distance via surveillance is of great importance for secure navigation.

With the success of deep learning in visual perception and recognition, a number of deep learning-based object detection methods have been proposed in the past decade [1–3] and have been applied to many tasks, such as vehicle detection [4,5], pedestrian detection [6–8], and so on. Object detection based on supervised deep learning requires a large number of labeled samples for training [9,10]. Then, we can define these methods as data-driven object detectors. Current deep learning-based detectors are based on the deep convolutional neural networks (DCNN). Generally, the DCNN-based detectors are composed of a backbone network and a box head network. The backbone network can be treated as a feature extractor, which can be pre-trained from a large dataset of image classification, e.g., the ImageNet, the COCO dataset, etc. The box head network processes the input features and generates the detection boxes to locate the objects in the image.

For the box head networks, some have a pre-defined anchor box, e.g., RCNN [1], Fast RCNN [11], Faster RCNN [2], and SSD [12], while others do not have this, e.g., Yolov1 [3], Yolov2 [13], CenterNet [14], CornorNet [15], and FCOS [16]. As a result, the deep learning-based object detectors can be divided into two categories, i.e., anchor-based and anchor-free methods. The anchor-based methods place a large number of preset boxes on the convolution feature. The network predicts the coordinate offset between the object frame and the anchor frame and combines the anchor frame coordinates and offset coordinates to

obtain the final object position. The anchor-free methods do not use dense preset boxes but use one or more key points to indicate the position of the object box on the image. in this way, the network predicts the key points.

Although the various detectors have shown their high performance in vehicle detection [17–19], pedestrian detection [6–8], and line detection [20–22], there are still some problems when handling the task of ship detection [23,24]. The main reason is the changing of scale. First, the scale of ships over the sea, lakes, and rivers can be very different. Some ships are very big, while some others are very small. Second, the focal length of the cameras can be very close or very far. As a result, some ships can be captured with a large scale or a small scale. For the above reasons, the detectors may suffer when detecting objects with widely varying scales. However, the DCNN-based detectors have only one backbone to extract the features. The question of how to fully use the convolutional features to cope with the ship features in different scales thus arises.

In order to solve the above problem, we investigate the scale sensitivity of ship detection. Without loss of generality, we choose the an anchor-free deep learning method CenterNet [14] as a framework. The CenterNet detection framework is equipped with a Resnet50 as a backbone network. In this framework, we will explore the characteristics of the size, depth and fusion mechanism of the convolutional features of the backbone network in detecting ships of different scales.

In the remainder of this paper, we will first introduce the related works in Section 2 and then describe the analysis of the scale sensitivity of ship detection in Section 3, present the experiments and results in Section 4, and finally conclude the work in Section 5.

2. Related Work

In this section, we briefly overview the related work on the convolutional neural network-based object-detection frameworks such as the two-stage and anchor-based method, the one-stage and anchor-based method, and the anchor-free method.

2.1. Two-Stage and Anchor-Based Method

Inspired by the success of conpvolutional neural networks in image classification tasks, Girshick et al. used the features of convolutional neural networks for image classification tasks for the first time in the target detection process and made great achievements in object detection and pedestrian detection. This is the birth of RCNN [1] using AlexNet [25]. However, t has three serious problems: (1) for a traditional CNN, the input map needs to be of a fixed size, and the deformation of the image during the normalization process will cause the image size to change, which is extremely unfavorable for the feature extraction of the CNN; (2) it is necessary to extract images corresponding to multiple candidate regions in advance, which will take up a lot of disk space; (3) each region proposal needs to enter the CNN network for calculation, and the same feature extraction is repeated multiple times.

Girshick et al. proposed FastRCNN [11] later. It uses the idea of feature sharing to map the region to be inspected and the last layer of convolutional features on the input image and extract the region on the convolution feature map corresponding to the region to be inspected on the image. In this way, the image is only subjected to one forward calculation in the convolutional network, and then all the features of the area to be inspected are extracted without having to perform multiple forward calculations. After the final convolutional layer, FastRCNN adds a region of interest pooling layer. The RoI pooling layer is actually a single-layer structure of SPP, which converts RoI regions with different shapes and sizes into the same dimension eigenvectors. The detection accuracy and running speed are thus improved. However, it still relies on algorithms outside the network to extract the area to be inspected, and the algorithm for extracting the area to be inspected has become the performance bottleneck of FastRCNN.

Some studies have shown that convolutional networks have the ability to locate in the image, and fully connected networks will reduce this effect. Therefore, Ren Shaoqing et al. [2] proposed the FasterRCNN framework and added their invented Region

Proposal Network (RPN) to the FastRCNN framework. In the FasterRCNN framework, RPN and FastRCNN share the convolutional features extracted by the backbone network. The final output of the convolutional feature of the backbone network is input into two branches: one branch is used to extract the frame to be anchored by the RPN, and the other branch is used to modify the coordinates of the area to be inspected to make it more accurate and determine the target category. RPN sets up n anchor boxes with different shapes and sizes for each coordinate position of the input feature map. When it is judged as a target, the coordinates of the anchor frame and the corresponding convolution feature area are extracted and then sent to the RoI pooling layer to obtain a fixed-dimensional feature vector. Then, they go through the following fully connected layer, and finally, classification and coordinate offset regression are performed. It completely uses convolutional neural networks to extract features without using manual features. However, each region mapped by the extracted anchor frame must undergo such classification and regression, which is very time-consuming, so the detection speed cannot meet the real-time requirements.

2.2. One-Stage and Anchor-Based Method

Redmon et al. proposed the YOLO [3] algorithm. Its core idea is to transform target detection into a regression problem using the entire image as the input of the network and many very small grids as the candidate regions for position regression. It only goes through one neural network to obtain the location and category of the bounding box. YOLO does not need to generate a proposal box, so it can run in real time at a speed of 45FPS. However, its accuracy is low, and especially for small targets and neighboring targets, the detection effect is poor; the overall prediction accuracy is slightly lower than Fast-RCNN. The main reason is that the grid settings are relatively sparse, and each grid only predicts two borders. In addition, the pooling layer loses some detailed information, which affects the positioning. Subsequently, YOLOV2 [13] and YOLOV3 [26] were also proposed. They all showed extremely high real-time performance, but their accuracy was not as good as the two-stage method.

Liu Wei et al. proposed SSD [12], which is a single-stage method that is faster than YOLO and maintains an advanced level of detection accuracy. It absorbs the innovative ideas of FasterRCNN and YOLO and effectively combines the anchor frame mechanism and multi-scale convolution features. SSD uses a fully convolutional structure, and its backbone network uses the convolutional layer part of VGG and adds several convolutional layers after the VGG convolutional network to increase its depth. The resolution of the feature map of the final convolutional layer is very small, the information contained is very rough, and the precise spatial position cannot be estimated. Therefore, SSD also uses high-resolution shallow convolution features to detect small targets. For all areas to be inspected, SSD uses convolutional feature maps of various sizes. On each layer of feature maps, the class probability of the target and the offset coordinate of the target coordinate relative to the predefined box are estimated. Finally, SSD uses the non-maximum suppression (NMS) algorithm to remove the repeated boxes to obtain the best detection results.

2.3. Anchor-Free Method

Law et al. recognized that the most obvious shortcoming of the anchor-based method is the process of setting up a predefined box. First, the number of anchor frames is very large. SSD needs to set close to 40,000 predefined frames, while Faster-RCNN uses more than 100,000 anchor frames to cover the image plane; secondly, dense anchor frames make positive and negative samples balanced, which is not conducive to training. Finally, setting the anchor frame requires a lot of hyperparameters, such as the number, size, shape, etc. of the anchor frame, which takes up a lot of video memory resources and affects the training and inference speed of the model.

Therefore, Law et al. [15] abandoned the idea of anchor boxes and proposed CornerNet. CornerNet converts the label data into heat maps corresponding to the upper left and lower right corners, associative embedding, and quantitative compensation and optimizes the

network weight by calculating the loss between these actual values and the corresponding network predicted values. The width and height of the heat map indicate the position of the corner point, and the number of its channels indicates the category, so the corner point position of CornerNet and the estimation of the category are coupled in the same loss function. When extracting the corner point, the position of the corner point in the heat map channel is also determined, so the category of the object it represents is also determined at the same time. Later, Law et al. [27] proposed CornerNet-Lite on the basis of improving CornerNet. It is a combination of CornerNet-Saccade and CornerNet-Squeeze, which are two variants of CornerNet. CornerNet-Saccade introduces an attention mechanism, which reduces the number of processed picture pixels and is suitable for offline processing. CornerNet-Squeeze uses a novel and compact basic network to reduce the amount of calculations for processing each pixel of the picture, which is suitable for real-time processing.

Tian et al. [16] proposed FCOS. It uses the characteristics of a full convolutional network, and the ratio of the feature map size to the image size is only determined by the step size of the pooling layer or the convolutional layer. It predicts the position of the target pixel by pixel. FCOS expresses the target box as any point in the box and the distance from this point to the 4 sides of the box, thus discarding the setting of the anchor box. The feature extraction network of FCOS uses the feature pyramid idea to predict convolution feature maps of multiple scales, and the predicted results are suppressed by non-maximum values to remove duplicate low-precision results. The FCOS prediction network uses all convolutional layers and outputs 3 tensors with the same resolution as the input image. Each pixel in the first tensor is a C-dimensional vector, which represents the category of the target that the pixel belongs to. Each pixel in the second tensor represents the centrality of the point in the box to which it belongs. Each pixel in the third tensor is a 4-dimensional vector, which represents the distance from the point to the four sides of the target box.

3. Scale Sensitivity of Ship Detection under the Anchor-Free Framework

This section uses CenterNet as the basic model to explore the influence of the size, depth, and fusion mechanism of the backbone network's convolution features on multi-scale target ships and find the best convolution feature settings in each scale range.

3.1. The Structure and Principle of the CenterNet Model

We chose CenterNet [14] as the basic pdetection framework and Resnet50 [28] as the basic backbone network for feature extraction. The backbone network structure is shown in Figure 1, consisting of a total of 5 network segments: *conv*1, *conv*2_*x*, *conv*3_*x*, *conv*4_*x*, and *conv*5_*x*.

The original first network segment *conv*1 is a single-layer convolutional network with a core of 7×7 and a step size of 2. We used two convolutional layers with a core of 3×3 instead, where the first convolutional layer has a step size of 2. The second network segment is composed of three identical residual blocks in series, and there is a maximum pooling layer with a core of 3×3 and a step length of 2 before the residual block. The third network segment is composed of 4 identical residual blocks in series. In the first residual block of the network segment, the convolutional layer with a core of 3×3 has a step size of 2. The fourth network segment is composed of 6 identical residual blocks. In the first residual block of the network segment, the convolutional layer with a core of 3×3 has a step size of 2. The fifth network segment is composed of 3 identical residual blocks, and the step size of the convolutional layer with a core of 3×3; the first residual block of the network segment is set to 2. It can be observed from the image input network that each time it passes through a network segment, the resolution of the output convolution feature map drops to $\frac{1}{2}$ of the previous network segment. We use the second, third, fourth, and fifth network segment output convolution feature maps to explore the network feature fusion strategy, denoted as C2, C3, C4, and C5, respectively. Inspired by the idea of CornerNet [15,27]

discarding anchor boxes and using corner points to represent box positions, Zhou et al. designed CenterNet. Unlike CornerNet, CenterNet uses the center point of the target box to indicate the position of the box, and the shape of the box is directly represented by its width and height. Similar to CornerNet, CenterNet does not directly predict the coordinates of the center point of the box, but it predicts the distribution of the center point of the box. The center point distribution is represented by a heat map.

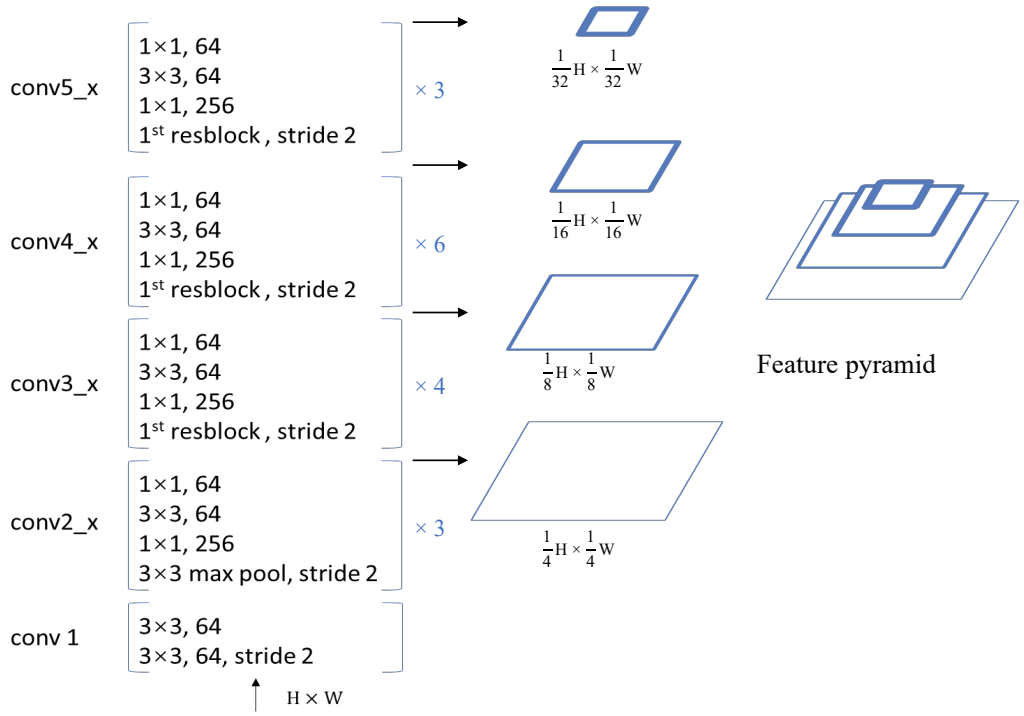


Figure 1. The structure of Resnet50 and its output feature pyramid. Note that H and W denote the height and width of the input image, respectively.

In Figure 2, the solid red rectangular box represents the actual box marked by the ship. When the center of the yellow rectangular box is near the center of the red rectangular box, the yellow dashed box can also correctly represent the ship. This is because the red solid line frame and the yellow dashed line frame have a larger intersection ratio at this time. The closer the center of the yellow box is to the center of the red box, the greater the intersection ratio between the two; on the contrary, the smaller the intersection ratio between the two. The green circle indicates the allowable range of the center point of the yellow dashed box. When it exceeds this range, it can no longer indicate the position of the ship. We call the green circle the distribution circle, and the radius of the green circle is the distribution radius. When the width and height of the red actual label box are determined, the distribution radius r is determined by the intersection ratio of the red and yellow boxes.

In the exploration experiment, we set $t = 0.7$, that is, the intersection ratio of the yellow box and the red box must be at least 0.7 to be marked as a positive sample. This processing can expand the number of positive sample points and reduce the impact of the imbalance of positive and negative samples when training the network. In addition, this central point representation mechanism can also improve the robustness of the model during reasoning and avoid false negative detection errors.

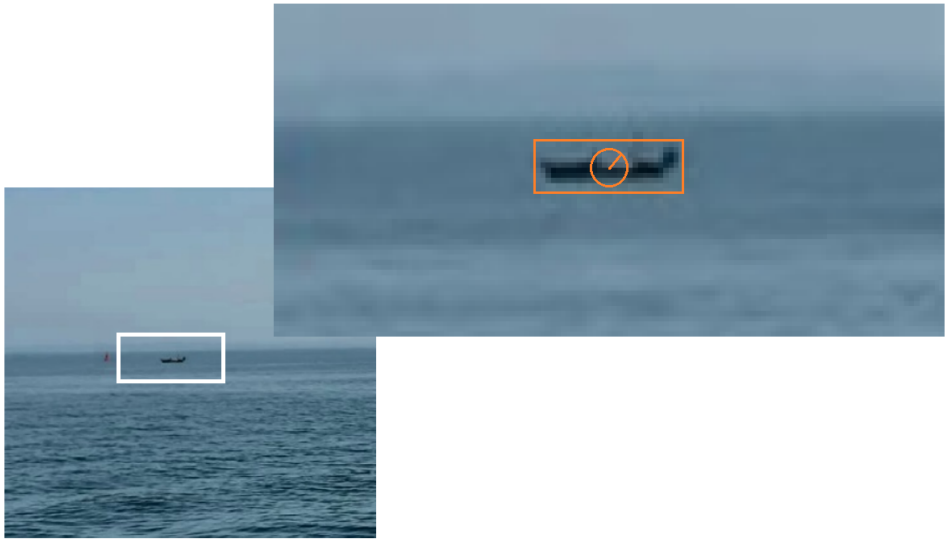


Figure 2. Using the distribution of center points to indicate the position of the ship.

We replaced the hourglass backbone network of CenterNet with Resnet50 in the actual setting of the exploration, and used it to study the 4-layer convolution feature map of the second, third, fourth, and fifth network segments, as shown in Figure 3. When using single-layer convolution feature map detection, the selected feature map is input into multiple consecutive transposed convolution layers, and the convolution feature map is enlarged to the required resolution; when using multi-layer convolution feature map detection, the selected feature map is input into the fusion network, and the convolution feature with the required resolution is obtained for detection. The feature map is obtained and then input to the three branches detection part. Each branch is a two-layer convolutional network. The convolution kernel of each layer is 3×3, the step size is 1, and the number of convolution kernels in the first layer is 256; the number of second-layer convolution kernels is different. The first branch is used to predict the heat map of the center point, the second branch is used to predict the width and height of the box, and the third branch is used to predict the quantized compensation value of the center point. The output of the three branches is three tensors. The resolution of the three tensors is the same, which is determined by the feature map of the input detection network, but their numbers of channels are different. The number of channels for predicting the branch output tensor of the center point heat map is the number of sample categories. In this experiment, there is only one type of ship, so the number of channels is 1. The number of channels for predicting the branch output tensor of the box width and height is sample 2 times the number of categories; each category predicts the width and height of the sample box in its respective category. The number of channels for predicting the branch output tensor of the quantized compensation value of the center point is 2, that is, the offset value in the x and y directions.

During training, the sample data used by CenterNet were converted into the form shown in Figure 4. Figure 4a is a training image. Taking the box of this ship as the center, a two-dimensional Gaussian distribution was generated to cover the location of the target, as shown in Figure 4b, and the center point heat map corresponding to this image was obtained. The function that produces this two-dimensional Gaussian distribution is formulated by Equation (1):

$$hm = exp(-((x - x_0)^2 + (y - y_0)^2)/(2\sigma^2)), \quad (1)$$

where (x_0, y_0) represents the coordinates of the center point of the target box, and σ is $\frac{1}{3}$ of the radius of the circle shown in Figure 2, where the radius is determined by the width and height of the target box. In the original CenterNet network, the resolution of the center point heat map is downsampled by $\frac{1}{2}$ from the original input, as shown in Figure 4c. The width and height of the box and the center quantization compensation are represented

by two vectors with length M and the number of channels 2. In addition, the index vector is required to establish a mapping relationship between the center point heat map and the box width and height vector and the center quantization compensation vector, as shown in Figure 4d.

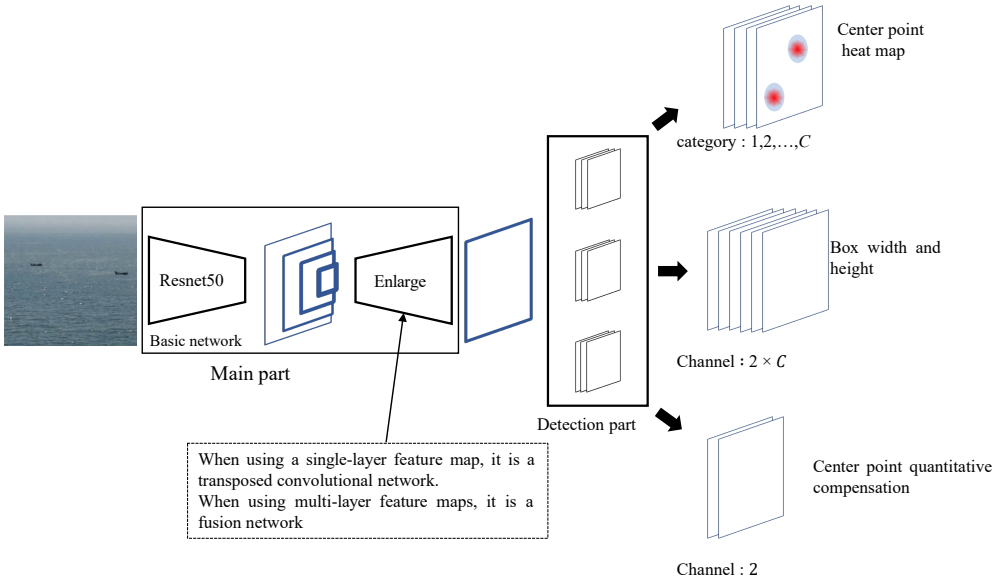
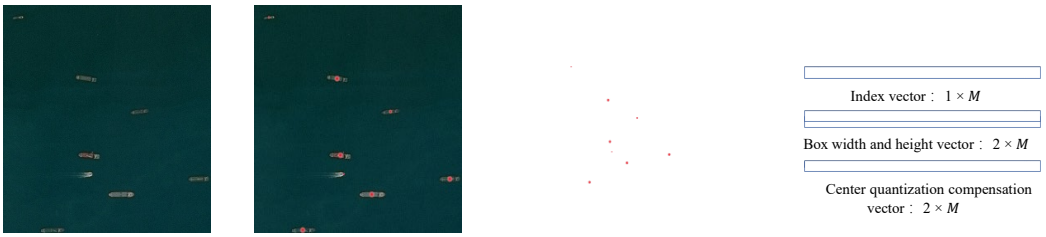


Figure 3. CenterNet algorithm diagram. Note that C denotes the number of channels of the feature map.



(a) (b) (c) (d)

Figure 4. Converting the labeled data to the form used during training. (**a**) The original image with a height H and a width W. (**b**) Gaussian distribution at the center of the box, with a resolution of $H \times W$. (**c**) The heat map of the center point; the resolution is $\frac{1}{4}H \times \frac{1}{4}W$. (**d**) M denotes the maximum number of predicted targets.

In inference, after the image was input to CenterNet, the center point heat map tensor, the center point quantization compensation value tensor, and the box width and height value tensor were obtained, as shown in Figure 5. From the center point heat map, the center point coordinates of the box and the box category can be directly extracted. Each specific value in the box width and height value tensor and the center point quantization compensation tensor has no practical meaning but can be converted into a box width and height vector and a center point offset vector through post-processing. Combined with the extracted coordinates of the center point of the box, the final coordinate value of the box can be obtained.

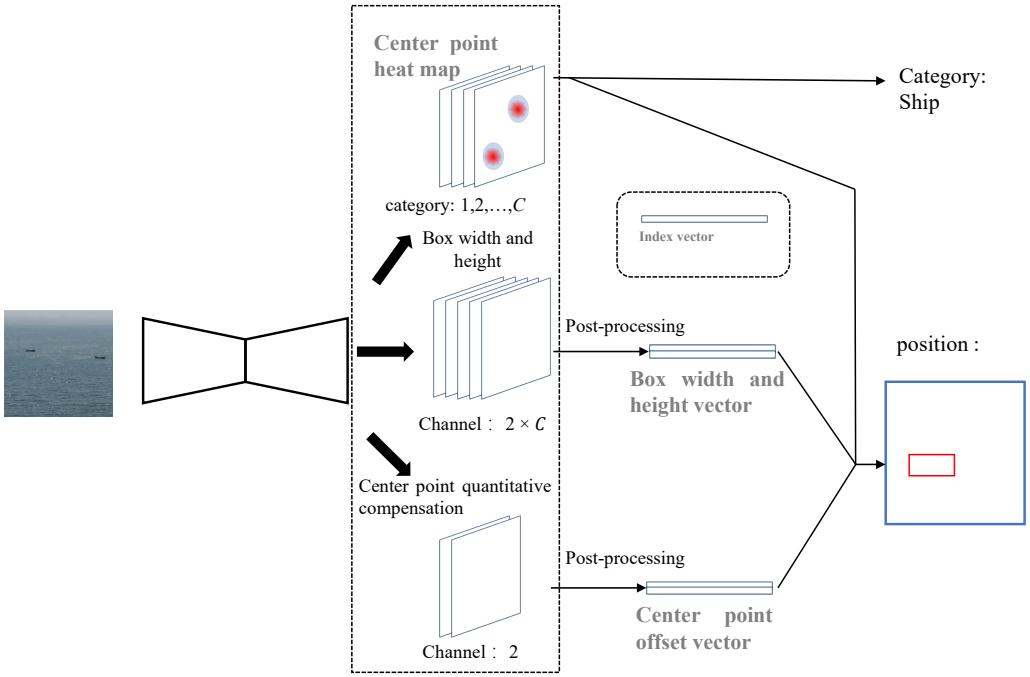


Figure 5. Schematic diagram of CenterNet detection part.

3.2. The Influence of the Size of Convolution Features on Multi-Scale Ship Target Detection

We used a downsampling rate of the convolution feature relative to the input image as the size of the convolution features. For example, the size of C4 is $\frac{1}{16}$, the size of C5 is $\frac{1}{32}$, the size of the input image is 1, and the size of the input image obtained by upsampling 2 times is 2. The original CenterNet connects 4 layers of deconvolution layers with a step length of 2 behind C5, and the resulting convolution featurep size is $\frac{1}{2}$.

In order to explore the influence of the convolution features of the backbonep network on the change of ship target scale, we especially studied the convolution features of C5. We connected a number of deconvolution layers with a step size of 2 and a 3×3 convolution kernel after C5 so that the size could be enlarged to $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$, 1 and 2.

The evaluation method is shown in Section 4.1.2. The experimental results are shown in Table 1. For small-scale ship targets, the larger the size of the convolution feature output by the backbone network, the better the detection performance. For medium-scale and large-scale ship targets, better detection results were obtained using smaller-size convolution features during detection. However, as the size of the convolution feature used for detection changes, the detection accuracy gain effects of ship targets of different scales also become different, as shown in Figure 6.

For small target ships, as the size of the convolution feature increases, the gain in detection performance obtained gradually decreases. When the size of the convolution feature increases from $\frac{1}{8}$ to $\frac{1}{4}$, the detection accuracy of small target ships can be improved by 6.3%. When the convolution feature size increases from $\frac{1}{4}$ to $\frac{1}{2}$ and from $\frac{1}{2}$ to 1, the detection accuracy of small target ships increases by 4.4% and 3.1%, respectively; however, when the convolution feature size is increased from 1 to 2, the detection accuracy gain of the small target ship is very slight, only 0.2%. The detection of small target ships requires a larger feature size, because this can make the receptive field smaller, and the detector does not mix too much background information when detecting small target ships; however, when the size of the convolution feature exceeds the size of the input image, it does not bring significant gain. We believe this is because the convolutional features that exceed the input size without being enhanced [29] will not obtain more refined details. Therefore, we believe that the convolutional features with the same size as the input image are most suitable for detecting small-scale ship targets. They can obtain almost the same detection accuracy as 2 times the input size, and at the same time, the computational cost is lower.

Table 1. The influence of convolution feature size change on ship detection accuracy.

Size	AP	AP^s	AP^m	AP^l
2	59.6	**51.5**	78.2	74.3
1	61.4	51.3	84.9	85.7
1/2	**63.3**	48.1	87.1	91.2
1/4	62.1	43.7	**88.3**	92.2
1/8	61.8	37.4	87.9	**93.1**

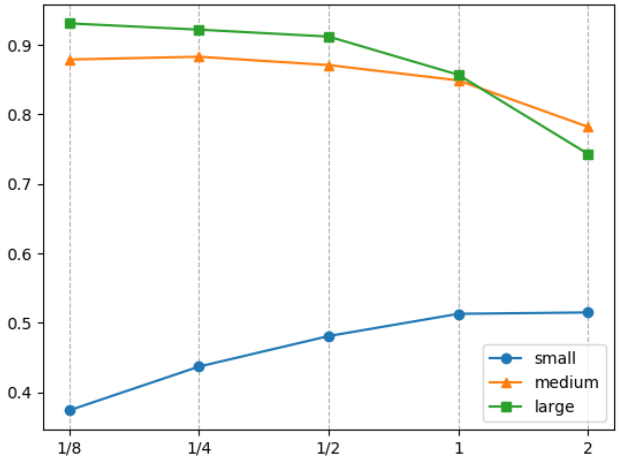


Figure 6. The detection accuracy of ship targets of different sizes is affected by changes in the convolution features' size. Thse ordinate represents the AP value; the abscissa represents the size of the convolution feature, $\frac{1}{2}$ represents $\frac{1}{2}$ of the input image size, 1 represents the same size as the input image, and 2 represents 2 times the input image size.

For ships of medium target size, the best detection result is obtained when the convolution feature size is $\frac{1}{4}$. When the convolution feature size is less than $\frac{1}{4}$, the larger the size, the lower the detection accuracy. When the convolution feature size is greater than $\frac{1}{4}$, the detection accuracy of the medium-sized target ship begins to decrease. When the convolution feature size decreases from $\frac{1}{4}$ to $\frac{1}{8}$, the detection accuracy drops by 0.4%. With the gradual decrease in the size of the convolution feature comes a greater decrease in the target detection accuracy of medium-sized ships, or, in other words, the greater the negative gain. When the size of the convolution feature is increased from $\frac{1}{4}$ to $\frac{1}{2}$, the detection accuracy of medium-sized target ships is reduced by 1.2%; when the size of the convolution feature is increased from $\frac{1}{2}$ to 1, the detection accuracy is reduced by 2.2%; when the convolution feature size is increased from 1 to 2, the detection accuracy is reduced by 6.7%. We believe that the most suitable convolution feature size for detecting medium-scale ship targets is $\frac{1}{4}$ of the input image. When the convolution feature size is greater than $\frac{1}{4}$, because the receptive field is too small, the detector cannot perceive all the information of the medium-sized ship target, which reduces the detection result; when the convolution feature size is greater than $\frac{1}{4}$, the receptive field is too large, so that the detector experiences interference from background information, which weakens the performance of the detector.

For large-scale ship targets, as the size of the convolution feature decreases, the detection performance gain obtained gradually decreases. When the size of the convolution feature is reduced from 2 to 1, the detection accuracy of large-scale ship targets can be increased by 11.4%; when the size of the convolution feature is reduced from 1 to $\frac{1}{2}$, the detection accuracy of large-target ships is increased by 5.5%. When the convolution feature size is reduced from $\frac{1}{2}$ to $\frac{1}{4}$ and from $\frac{1}{4}$ to $\frac{1}{8}$, the detection accuracy is increased by 1% and 0.9%, respectively, and the increase is very small. Detecting large ship targets requires a larger receptive field. Using a smaller size convolution feature can enable the detector to obtain a larger receptive field and obtain better detection results. When the convolution

feature size is large, the target detector receives interference from the background signal. However, compared with small and medium-sized ship targets, the detection of large-scale ship targets is less sensitive to background signals. With certain background information mixed in, the detector can also detect large-scale ship targets. Therefore, when detecting large ship targets, and when the size of the convolution feature is reduced from $\frac{1}{8}$ to $\frac{1}{16}$, we speculate that if the detection accuracy increases, the increase will not exceed 0.9%; if the detection accuracy decreases, the drop will not be too great. According to the experimental results and analysis, we believe that the size of the input image after downsampling by $\frac{1}{8}$ is most conducive to detecting large-scale ship targets.

In addition, from Figure 6, it can be found that the detection performance characteristics of medium-scale and large-scale ship targets regarding convolution feature size are similar in trend. However, compared with medium-scale and large-scale ship targets, the detection performance characteristics of small target ships regarding convolution feature size is significantly different. First, the detection accuracy of the convolution feature size of large-scale and medium-scale ship targets is about 75–95%, while the range of small targets is about 35–55%; second, the detection accuracy of large-scale and medium-scale ship targets tends to decrease with the increase in the convolution feature's size, while the detection accuracy of small target ships increases with the increase of the convolution feature's size. These two differences show that the size of the convolution feature has inconsistencies in the detection of small target ships and non-small target ships, but this inconsistency is not the main factor that leads to the poor detection accuracy of small target ships.

3.3. The Influence of the Depth of Convolution Features on Multi-Scale Ship Target Detection

C2, C3, C4, and C5 come from the output of different depth layers of Resnet50, where C2 is the output of the 11th layer, C3 is the output of the 23rd layer, C4 is the output of the 41st layer, and C5 is the output of the 50th layer. Therefore, we directly use these four convolution features to explore the influence of the depth of the backbone network convolution feature on the changes in ship target scale.

In the experiment, we first used the complete Resnet50 as the basic feature extraction network and added a 4-layer transposed convolution to enlarge the feature size. We used the trained convergent model to characterize the performance of C5; we took the C4 part of the convergent network and added 3 transposed convolutional layers, froze the weight of the C4 part during training, and only updated the network weights of the amplified part and the detected part. The resulting model was used to characterize the performance of C4. We took the C3 part of the convergence network and added 2 transposed convolutional layers, froze the weight of the C3 part during training, and only updated the weight parameters of the amplification part and the detection part, and the resulting model is used for characterization of C3's performance. We took the C2 part of the convergence network and added a transposed convolutional layer, froze the weight of the C2 part during training, and only updated the weight parameters of the amplification part and the detection part; the resulting model was used to characterize the performance of C2. Adopting the above-mentioned experimental settings, on the one hand, prevents the interference of the convolution feature's size; on the other hand, the weight of the shared feature extraction network remains consistent, which means that the inconsistency of the initialization parameters and the inconsistency of the convergence path are eliminated.

The experimental results are shown in Table 2. Small-scale ship targets use C3 depth convolution features to obtain the best detection results; medium-scale ship targets use C4 depth convolution features to make the detection accuracy the highest; large-scale ship targets use C5 depth convolution features to make the detection accuracy the highest. Because the weight parameters of the feature extraction basic network are shared in the experiment, from the point of view of parameter fitting, for small target ships, the C3 convolution feature is just fitting, the C2 feature is under-fitting, and C4 and C4 are under-fitting. The convolutional features of the C5 layer are in an over-fitting state. For

medium-sized ship targets, the C4 feature is in a fitting state, the C2 and C3 features are in an under-fitting state, and the C5 feature has entered an over-fitting state. For large-scale ships C2 and C3 are obviously under-fitting, while C4 to C5 are only increased by 0.2%, and the two are almost the same. It can be considered that C4 and C5 have reached the fitting state.

Table 2. The influence of convolution feature depth changes on ship detection accuracy.

Size	AP	AP^s	AP^m	AP^l
C5	63.3	48.1	87.1	**91.2**
C4	**66.8**	52.4	**90.8**	91.0
C3	58.7	**54.3**	63.0	67.7
C2	27.9	34.8	28.2	23.8

The characteristics of ship targets of different scales with respect to the depth of convolution features are shown in Figure 7. It can be found that large- and medium-scale ship targets have strong similarities in trends, while small targets have great differences compared with them. The detection accuracy of large and medium target ships on the convolution features of C2 depth is less than 30%, while the detection accuracy of small target ships on the convolution features of C2 depth is 34.8%, which is higher than that of large and medium targets. The detection accuracy of large and medium targets increases rapidly when the convolution feature depth increases from C2 to C4, and it rises or decreases slightly from C4 to C5, while the detection accuracy of small targets only rises from C2 to C3 and reaches C3. After the high point, C4 and C5 slowly fall back. Through analysis, it can be seen that the detection of large- and medium-sized ship targets is more dependent on deeper convolutional layer features, while the detection of small target ship targets is more dependent on shallower convolutional layer features. In other words, the depth of the convolution feature has inconsistency in the detection of small target ships and non-small target ships. This inconsistency will aggravate the phenomenon in which the detection accuracy of small target ships is much lower than that of non-small target ships. It can be said that, among the factors that make it more difficult to detect small targets than non-small target ships, the inconsistency of the convolution feature depth on the target scale is more important than the inconsistency of the convolution feature size on the target scale.

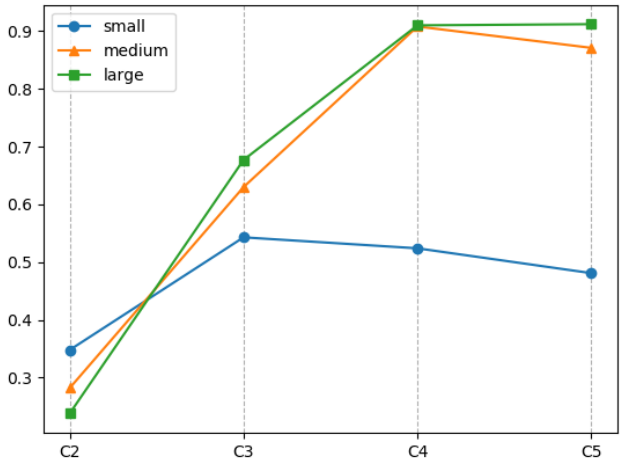


Figure 7. The detection accuracy of ship targets of different sizes is affected by changes in convolution feature depth. The ordinate represents the AP value; the abscissa represents the depth of the convolution feature. C2, C3, C4, and C5 represent the convolution features output by the 11th, 23rd, 41st, and 50th layers, respectively.

From Figures 6 and 7, it can be observed that whether it is about the size or depth of the convolution features, the characteristics of large and medium-sized ship targets have a strong similarity, while their characteristics and the characteristics of small target ships are very inconsistent. We believe that this similarity and inconsistency are determined by the way the convolutional network expresses the characteristics of ships of different scales.

Some studies have pointed out that the convolutional network mainly recognizes objects by extracting texture features instead of shape or color features [30]. Texture can be characterized by spectral characteristics, and the forward process of convolutional neural networks is a multi-layer nonlinear activated filter. Many studies [12,31] believe that in the target detection model based on convolutional neural networks, shallow features have more accurate detailed information, while deep features have richer semantic information. The shallow layer of the convolutional network extracts fine-grained texture features with a smaller receptive field, and the deep layer extracts coarse-grained texture features with a larger receptive field. The shallow convolution features are more detailed, and as the network deepens, the expression of the convolution features becomes more and more abstract. However, the texture information of the small target itself is sparse, and the convolutional neural network can only rely on the texture feature pattern extracted in the shallow layer for detection. Large- and medium-scale targets are rich in texture information, and convolutional neural networks can only obtain their proper feature expression at a deeper level. In the deep convolutional features, the feature expression of small targets will be too abstract. Therefore, the detection results of small target ships using shallower convolution features are better than using deep convolution features, and large and medium-scale ships have to use deep convolution feature roots to obtain the best detection results. We believe that large and medium-scale ship targets have similarities in the expression of convolution features, while small target ships and non-small target ships have inconsistencies in the expression of convolution features. These similarities and inconsistencies are caused by the inherent working mode of convolutional neural networks, which can be regarded as the inherent characteristics of ship targets of different scales.

3.4. The Effect of Convolutional Feature Fusion Mechanism on Multi-Scale Ship Target Detection

There are two types of convolution feature map fusion operations: (1) adding elements element by element; (2) splicing along the channel. They can be divided into early fusion and late fusion according to the order of fusion and prediction. Early fusion refers to the fusion of multi-layer features into single-layer features, and predictions are made on the single-layer features after the fusion, such as in [32,33]. Late fusion refers to the use of multiple different levels of convolutional feature maps for prediction and the fusion of the detection results of multiple features. Late fusion can be divided into two categories: the first category directly uses the multi-layer features extracted by the basic network to predict, and the multi-layer prediction results are merged to obtain the final result, such as SSD [12]; After the feature map is fused, the multi-layer fusion feature of the pyramid structure is obtained, and the prediction result of the multi-layer fusion feature is further fused to obtain the final result. FPN and PAN are the most typical representatives.

We noticed that the feature pyramid fusion adopts a layer-by-layer transfer and gradual fusion method. There are the same number of feature layers before and after the fusion, and the feature maps of the corresponding levels have the same resolution. In [31,34], the feature pyramid has two fusion paths: (1) top-down and (2) bottom-up. Figure 8a shows the top-down pyramid fusion method. The left side is the feature pyramid composed of 4 layers of tensors extracted by the basic network, which are $C_n(n \in 2, 3, 4, 5)$, and on the right is the merged feature pyramid, which are $P_n(n \in 2, 3, 4, 5)$. Before the fusion operation, a 1×1 convolution will be used to convert the feature tensor of the 4-layer basic network into the same number of channels, N, generally set as $N = 256$. The top-down fusion method starts from C5; C5 obtains I5 after 1×1 convolution, and I5 directly enters a 3×3 convolution layer to obtain P5. The up-sampling of I5 and the result of 1×1 convolution with C4 are added element-by-element to obtain I4, and I4 is obtained

after a 3×3 convolution layer to obtain P4. Up-sampling I4 to double its size, adding the result to the tensor of C3 after 1×1 convolution element-by-element to obtain I3. I3 goes through a 3×3 convolution layer to obtain P3. We then upsample I3 to enlarge the size, add the result to the tensor after 1×1 convolution of C2 element by element to obtain I2, and let I2 pass through a 3×3 convolution layer to obtain P2. By observing the pyramid fusion operation process of the top-down path, it can be found that the fusion features of each layer are unequal. P5 does not incorporate the convolutional features of other layers; it is the result of C5 passing through a 2-layer convolutional network. P4 only incorporates two convolutional features of C4 and C5. C4 and C5 are finally passed to P4 through I4, but C5 is passed to I4 after I5 and upsampling. I5 will lose a certain amount of information during upsampling. C4 is directly passed to I4 through a layer of convolution with a step size of 1, so the information of C4 obtained in the fusion feature P4 is stronger than that of C5. P3 combines the three-layer convolution features of C3, C4 and C5. C4 and C5 are passed to I3 after I4 and up-sampling. At the same time, I3 integrates the characteristic information from C3. Since I4 is up-sampled and C4 is not scaled, the information of C3 obtained in the fusion feature P3 is stronger than C4, and C4 is stronger than C5. Only P2 incorporates the convolutional features of all four layers of C2, C3, C4 and C5. I3, which combines the features of C3, C4, and C5, is up-sampled and transferred to I2. At the same time, I2 combines the information passed by C2. Similar to the previous one, the information of C2 obtained in the fusion feature P2 is stronger than that of C3, and C3 is stronger than C4, while C5 has the weakest information.

Figure 8b shows the bottom-up pyramid fusion method. The difference is that the bottom-up fusion method starts from C2 and gradually transfers to each layer above. Similar to the top-down fusion path, the fusion characteristics of each layer of the bottom-up fusion path are not equal. The difference is that in the bottom-up feature pyramid, P2 does not integrate the feature information of any other layers. It is only obtained by C2 through a two-layer convolutional network. P3 combines the two-layer convolution features of C3 and C2. C2 is fused with C3 through I2 downsampling to obtain I3, which means that C2 and C3 pass the fusion information to P3 through I3. Since I2 loses certain information after downsampling, the information of C3 obtained by fusion in P3 is stronger than C2. Similarly, P4 combines the three-layer convolution features of C2, C3, and C4. The information obtained by fusion in P4 is stronger than that of C3, and the information of C3 is stronger than that of C2. Only P5 integrates the convolution features of all four layers. The information obtained by the fusion of C5 is stronger than C4, C4 is stronger than C3, and the information obtained by C2 is the weakest.

From the above content, we can know that whether it is top-down or bottom-up, only the fusion feature tensor at the end of the fusion path accepts the information of all convolutional features of the basic network, and the fusion network feature behind the fusion path does not contain the front of the fusion path information about basic network characteristics. In addition, the basic feature information acquired in the fusion feature is not equal. The basic feature corresponding to the fusion feature transmits the strongest information; the earlier the basic feature on the fusion path transmits, the weaker the information. If the basic feature pyramid has N layers, from shallow to deep, respectively, denoted as $C_0, C_1, C_2, \ldots, C_{N-1}$, then there are $N(N-1)$ ways to select the basic features of two or more adjacent layers to be merged into a single-layer feature.

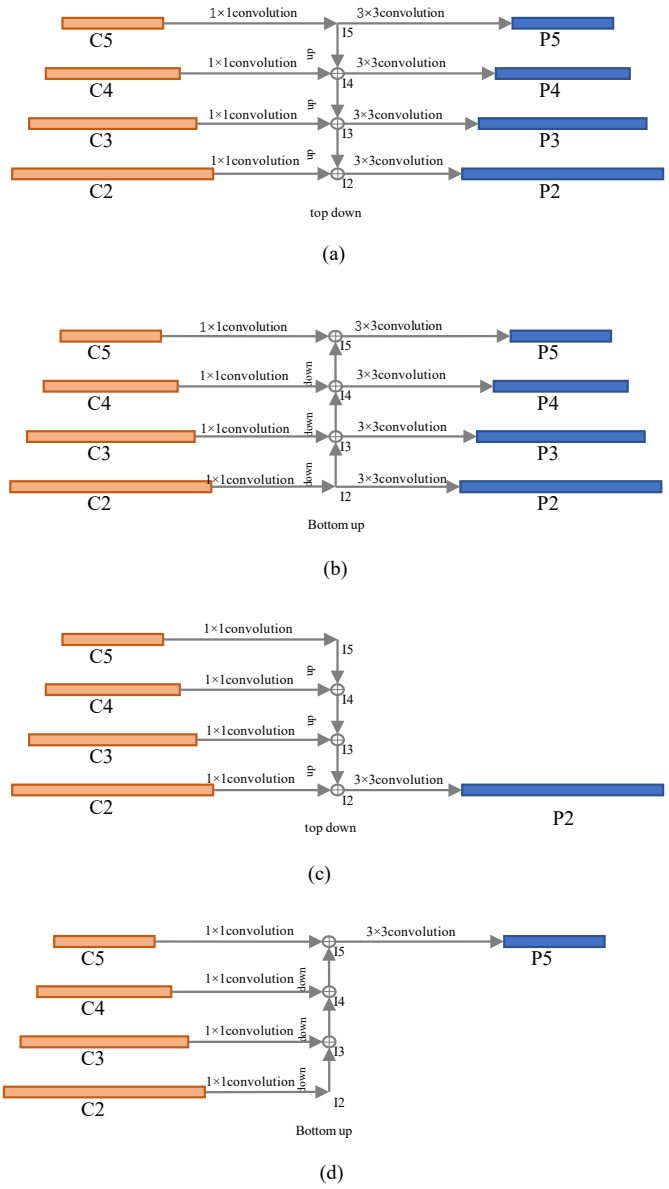


Figure 8. Two strategies of feature pyramid fusion. (**a**) Top-down pyramid fusion. (**b**) Bottom-up pyramid fusion. (**c**) Top-down single-layer fusion. (**d**) Bottom-up single-layer fusion.

4. Experiments and Results

4.1. Dataset and Evaluation Method for Ship Target Detection

4.1.1. Dataset

This paper uses the Airbus ship detection dataset for experimental study. In order to promote the development of ship detection technology from satellite optical images, the Airbus Group has produced the Airbus ship detection dataset and released related competitions on the Kaggle competition platform. The Airbus ship detection dataset on the Kaggle platform only discloses part of the training set. We use part of the public dataset for exploratory experiments. The public dataset of Airbus ship detection contains a total of 192,556 images, each with a resolution of 768 × 768. In the pictures, some do not contain ships, and some contain one or more ships. We divide the public part of the Airbus ship detection dataset into two parts: a training set and a test set. The training set has a total of 134,789 images, in which 29,789 images contain ships, and 105,000 images do not contain any ship. A total of 12,767 images in the test set contain ships, and 45,000 images do not contain ships. The Airbus ship detection dataset is labeled in the form of binary segmentation. In addition, following the protocol of the COCO dataset, we define ships

with a pixel area less than 32^2 as small targets, ships with a pixel area in the range of 32^2 to 96^2 as medium-scale targets, and ships with a pixel area greater than 96^2 as large-scale targets. The details of ships of each size in the Airbus ship detection dataset are shown in Table 3.

Table 3. Airbus ship detection dataset.

Dataset	Number of Images with Ship/without Ship/Total	Image Size	$<32^2$	$32^2 \sim 96^2$	$>96^2$
Training set	29,789/105,000/134,789	768×768	37,863	50,500	6834
Test set	12,767 /45,000/ 57,767	768×768	16,047	21,432	2957
Total	42,556/150,000/192,556	768×768	53,910	71,932	9791

4.1.2. Evaluation Method

The experiment in this article used the evaluation method of COCO [35]. Therefore, the COCO APIs provided by the official COCO dataset could be used. In COCO APIs, the mean average precision is used as the main evaluation index. Average precision refers to the average of precisions of all categories under the determined prediction box and the IOU threshold of the labeled box. The calculation of average precision AP is defined by Equation (2):

$$AP = \frac{1}{c} \sum_c P_c(iou > \theta), \tag{2}$$

where θ is the threshold of the IOU, $P_c(iou > \theta)$ represents the accuracy of category c in $(iou > \theta)$, and C is the number of categories. The mean average precision refers to taking multiple IOU thresholds and calculating the average of the average precision under the multiple IOU thresholds. The mean average precision is defined by Equation (3):

$$mAP = \frac{1}{KC} \sum_k \sum_c P_c(iou > \theta), \tag{3}$$

where K is the number selected by the IOU threshold. Following the setting of COCO, the threshold of the IOU is 0.50:0.05:0.95, from 0.5 to 0.95, at an interval of 0.05, and with $K = 10$. In this experiment, as there is only one category of ships, the accuracy of one category of ships can be calculated.

In addition, we calculated the mean average precision of small-scale ships, medium-scale ships, and large-scale ships with Equations (4), (5), and (6), respectively.

$$mAP^{small} = \frac{1}{KC} \sum_k \sum_c P_c(iou > \theta_k, area < 32^2), \tag{4}$$

$$mAP^{medium} = \frac{1}{KC} \sum_k \sum_c P_c(iou > \theta_k, 32^2 < area < 96^2), \tag{5}$$

$$mAP^{large} = \frac{1}{KC} \sum_k \sum_c P_c(iou > \theta_k, area < 96^2). \tag{6}$$

For simplicity, in the following texts, mAP is recorded as AP, mAP^{small} is recorded as AP^s, mAP^{medium} is recorded as AP^m, and mAP^{large} is recorded as AP^l.

4.2. Experimental Setup

4.2.1. Model Training Details

We implemented the models on the PyTorch deep learning platform, with Python as the programming language. Custom algorithms were not used. For the Airbus ship detection data set, the input size was 768×768 in the training. The input size on the WHUCS ship detection data set was 1200×720, and the output sizes of the three branches—large, medium and small—were 150×90, 300×180, and 1200×720, respectively. In order to

reduce model overfitting, we used common data augmentation techniques, including random horizontal flipping, random scaling and cropping, and random color dithering, where color dithering involved adjusting the brightness, saturation, and contrast of the image. Finally, the image was normalized by Z-score and then input into the deep neural network.

4.2.2. Model Test Details

In the testing phase, we used a simple post-processing process to convert the heat map tensor, center offset tensor, and box width and height tensor output by each of the three detection branches into box coordinates. First, we used a 3×3 maximum pooling layer for the central heat map to suppress non-maximum values. Next, we selected the first 100 peaks in the center heat map of each branch, found the corresponding values in the center offset tensor and the box width and height tensor through the coordinates of the peaks, and removed the results that were not in the detection range of the respective branches. If tilted frame was detected, the dual direction vector was also predicted. Assuming that the output of the network was $[\vec{a_1}, \vec{a_2}]$, we took $\left[\frac{\vec{a_1}-\vec{a_2}}{||\vec{a_1}-\vec{a_2}||}, \frac{\vec{a_2}-\vec{a_1}}{||\vec{a_1}-\vec{a_2}||}\right]$ as the prediction result. Then, we converted the center point of the corresponding box, the center point offset, and the width and height of the box into the coordinate values of the box in the original image. The peak at the center point represented the confidence score of the detection as a ship. Finally, the respective results of the three detection branches were combined, and the non-maximum suppression algorithm was applied to remove redundant duplicate results; the first 100 detection frames were selected as the final detection results of the model according to the confidence level, from high to low.

4.2.3. Results

We used different fusion network to explore the impact of fusion features on the accuracy of ship target detection at different scales. In order to eliminate the influence of the size of the convolutional features, we enlarged the features output by the fusion network to a uniform size through several transposed convolutions. When studying the influence of the fusion feature on the detection accuracy of small target ships, the fusion feature was enlarged to 1, and the experimental results are shown in Table 4.

Table 4. Detection results of small-scale ships with different fusion methods.

Fusion Method	Top Down	Bottom Up
C2 + C3	55.2	56.8
C3 + C4	56.1	53.7
C4 + C5	52.7	48.4
C2 + C3 + C4	56.3	53.1
C3 + C4 + C5	**61.2**	51.3
C2 + C3 + C4 + C5	59.8	50.7

It can be seen from Table 4 that the best detection effect for small target ships is achieved by selecting C3, C4 and C5 and adopting the top-down path fusion feature. Except for the combination of C2 and C3, the fusion features of other combinations are the results of the top-down path better than the bottom-up path. In the top-down path fusion method, the lower-layer information obtained in the fusion feature is stronger than the higher layer, so its main information component is the closest to the lowest layer feature participating in the fusion. In the bottom-up fusion method, the lower-level information obtained in the fusion feature is weaker than the higher-level, so its main information component is the closest to the feature of the highest level participating in the fusion. Moreover, the use of shallower convolutional features to detect small-scale ship targets has better accuracy. Thus, the top-down path is better than the bottom-up path when detecting small target ships. However, when only the two layer features of C2 and C3 participate in the fusion, the main component of the feature P2 obtained from the top-down path is

C2, and the main component of the feature P3 obtained from the bottom-up path is C3. The detection of small target ships depends most on the information of the C3 layer, so the bottom-up path is better than the top-down path. Among all the fusion methods, selecting C3, C4 and C5 to obtain the best result from top-down path fusion is even better than the detection results of the fusion of all features. This is because under the premise of a top-down path, the main components of the fusion features of C3, C4, and C5 are closest to C3, and the main components of the fusion features of C2, C3, C4, and C5 are closest to C2. In the case of single-layer features, the results of C3 are better than C2. Using the same top-down path, the results of all fusions of C2, C3, C4, and C5 are better than that of the three-layer fusion of C2, C3, and C4. We speculate that C5 participates in the fusion and enhances the semantic information of the feature.

Table 5 shows the detection results of medium-scale ship targets by different fusion methods. Through observation, it can be found that in all combinations, the bottom-up fusion path is better than the top-down path. This is because the main component of the fusion feature of the top-down path is close to the single-layer feature of the lowest layer, and the main component of the fusion feature of the bottom-up path is similar to the feature of the highest layer; the detection of medium-scale ship targets mainly depends on the high-level feature. In the bottom-up fusion method, the detection results of the C2, C3, and C4 fusion features are the best, and it is better than the detection results of all four layers of C2, C3, C4, and C5. We believe this is due to two reasons. On the one hand, the main components of the C2, C3, and C4 fusion features are close to C4, while the main components of all four layer fusion features are close to C5. C5 is already in an over-fitting state, and the components of C4 in the four-layer fusion feature are weaker than C2 and C3 for the in the three-layer fusion feature. On the other hand, in the fusion features of C2, C3, and C4, C2 was attenuated twice, and C3 was attenuated once, while in the fusion features of C2, C3, C4, and C5, C2 was attenuated three times, and C3 was attenuated twice. Therefore, the shallow information in the C2, C3, and C4 fusion features is stronger than the shallow information in the C2, C3, C4, and C5 fusion features. At this time, the shallow information of C2 and C3 is more helpful than C5 to improve the detection ability of fusion features on medium-sized ships. This can also reasonably explain that the detection effect of the fusion features of C2, C3, C4, and C5 is better than that of the fusion features of C3, C4, and C5.

Table 5. Detection results of medium-scale ships with different fusion methods.

Fusion Method	Top Down	Bottom Up
C2 + C3	52.3	63.7
C3 + C4	78.4	89.5
C4 + C5	88.3	87.6
C2 + C3 + C4	58.7	**90.1**
C3 + C4 + C5	76.2	89.3
C2 + C3 + C4 + C5	62.3	89.9

Table 6 shows the detection results of large-scale ship targets by different fusion methods. Observing the two columns, we can find that the bottom-up fusion path has better detection results than the top-down fusion path. This is the same rule for detecting medium-sized ship targets. The reason is similar. The model mainly uses high-level C4 and C5 features when detecting large ship targets. It can be noticed that choosing the three-layer features of C3, C4, and C5 to use the bottom-up path fusion feature to detect large-scale ships has the best effect, but it is only slightly better than the results of the fusion detection of C2, C3, C4, and C5. At the same time, the detection effect of the C2, C3, C4, and C5 fusion features on large-scale ship targets has been significantly improved compared with the C2, C3, and C4 fusion features. We believe this is because the fusion feature detection of large-scale ships mainly relies on two high-level features, C4 and C5. For shallow features, only C3 can improve the fusion feature detection of large target

ships, and C2 will also cause performance degradation. The detection results of the C3, C4, and C5 fusion features are significantly better than the C4 and C5 fusion features, which can enhance the previous conclusions.

Table 6. Detection results of lager-scale ships with different fusion methods.

Fusion Method	Top Down	Bottom Up
C2 + C3	0.591	0.638
C3 + C4	0.732	0.893
C4 + C5	0.819	0.906
C2 + C3 + C4	0.657	0.902
C3 + C4 + C5	0.864	**0.917**
C2 + C3 + C4 + C5	0.786	0.914

Through experiments, this subsection shows that the backbone features extracted by the neural network in the detection task have different characteristics for the representation of targets of different scales. The feature expressions of large-scale and medium-scale targets have certain similarities, while the feature expressions of small targets are quite different from them. We believe this is the main reason why the detection accuracy of small targets in deep learning-based detection algorithms is far lower than that of mass-scale targets. In addition, this chapter also explains the use of different basic feature selection fusion mechanisms, which can adjust the expression characteristics of the backbone network convolution features to ship targets of different scales, so that the neural network can achieve the best state of detection accuracy of ship targets at various scales.

When training our detection model, the training data included not only the stand-alone ships, but also some cargo ships. Therefore, the trained model can detect both stand-alone ships and moving cargo ships. If more types of ships are included in the training set, the models will gain the ability to perform more generalization.

This work focuses on the scale sensitivity of ship detection in an anchor-free framework. However, we do not analyze the effect of scale on feature fusion and ship detection in any anchor-based detectors. This is a limitation of our work. We plan to study this point in our next work.

5. Conclusions

In this paper, with CenterNet as the basic detection framework and ResNet50 as the backbone network, the influence of the scale, depth, and fusion strategy of the convolution features of the backbone network was studied in the context of multi-scale ship detection. Experiments showed that, in general, using larger-sized shallow features is beneficial to detect small target ships, while using smaller-sized deep features is more suitable for detecting large ship targets. This paper took pyramid feature fusion as the basic feature-fusion method and studied the effect of multi-layer fusion convolution features on ship target detection of different scales. Experiments showed that for small targets, the fusion features obtained by the top-down path can improve the detection performance better than the bottom-up path; on the contrary, the bottom-up path has a better detection effect on non-small target ships. Through a comprehensive analysis of the three sets of experiments, it was found that, regardless of the scale, depth and fusion strategy, the detection performances of the three factors on the ship targets of medium scale and large scale were relatively similar. The rules on small-scale ships are found to be completely opposite to those for large-scale ships. A possible reason is that the convolutional neural network has very different feature expressions on small and large ships. This is also the essential reason for the low accuracy of small object detection and multi-scale object detection.

Author Contributions: Conceptualization, Y.J.; Methodology, Y.J.; Software, L.H.; Validation, Z.Z.; Formal analysis, B.N.; Investigation, Z.Z.; Data curation, L.H.; Writing—original draft, Y.J.; Writing—review & editing, F.Z.; Supervision, B.N.; Project administration, F.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported partially by the Provincial Natural Science Foundation under grant 2021099127.

Data Availability Statement: Not applicable.

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

References

1. 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.
2. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
3. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
4. Hu, H.N.; Cai, Q.Z.; Wang, D.; Lin, J.; Sun, M.; Krahenbuhl, P.; Darrell, T.; Yu, F. Joint monocular 3D vehicle detection and tracking. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 5390–5399.
5. Chen, L.; Zou, Q.; Pan, Z.; Lai, D.; Zhu, L. Surrounding Vehicle Detection Using an FPGA Panoramic Camera and Deep CNNs. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 5110–5122. [CrossRef]
6. Pang, Y.; Xie, J.; Khan, M.H.; Anwer, R.M.; Khan, F.S.; Shao, L. Mask-guided attention network for occluded pedestrian detection. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4967–4975.
7. Liu, S.; Huang, D.; Wang, Y. Adaptive nms: Refining pedestrian detection in a crowd. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6459–6468.
8. Yu, H.; Yu, H.; Guo, H.; Simmons, J.; Zou, Q.; Feng, W.; Wang, S. Multiple human tracking in wearable camera videos with informationless intervals. *Pattern Recognit. Lett.* **2018**, *112*, 104–110. [CrossRef]
9. Kanjir, U.; Greidanus, H.; Ostir, K. Vessel detection and classification from spaceborne optical images: A literature survey. *Remote Sens. Environ.* **2018**, *207*, 1–26. [CrossRef] [PubMed]
10. Xiong, W.; Jia, X.; Yang, D.; Ai, M.; Li, L.; Wang, S. DP-LinkNet: A convolutional network for historical document image binarization. *KSII Trans. Internet Inf. Syst.* **2021**, *15*, 1778–1797. [CrossRef]
11. 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.
12. 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.
13. Redmon, J.; Farhadi, A. Yolo9000: Better and Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
14. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6569–6578.
15. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 765–781.
16. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9626–9635.
17. Zhao, X.; Sun, P.; Xu, Z.; Min, H.; Yu, H. Fusion of 3D LIDAR and camera data for object detection in autonomous vehicle applications. *IEEE Sens. J.* **2020**, *20*, 4901–4913. [CrossRef]
18. Chen, L.; Ding, Q.; Zou, Q.; Chen, Z.; Li, L. DenseLightNet: A light-weight vehicle detection network for autonomous driving. *IEEE Trans. Ind. Electron.* **2020**, *67*, 10600–10609. [CrossRef]
19. Chen, L.; Zhang, Y.; Tian, B.; Ai, Y.; Cao, D.; Wang, F.Y. Parallel Driving OS: A Ubiquitous Operating System for Autonomous Driving in CPSS. *IEEE Trans. Intell. Veh.* **2022**, 1–10. [CrossRef]
20. Chen, C.; Yang, B.; Song, S.; Peng, X.; Huang, R. Automatic clearance anomaly detection for transmission line corridors utilizing UAV-Borne LIDAR data. *Remote Sens.* **2018**, *10*, 613. [CrossRef]
21. Cao, Y.; Ju, L.; Zou, Q.; Qu, C.; Wang, S. A Multichannel Edge-Weighted Centroidal Voronoi Tessellation algorithm for 3D super-alloy image segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011; pp. 17–24.

22. Chen, C.; Jin, A.; Yang, B.; Ma, R.; Sun, S.; Wang, Z.; Zong, Z.; Zhang, F. DCPLD-Net: A diffusion coupled convolution neural network for real-time power transmission lines detection from UAV-Borne LiDAR data. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102960. [CrossRef]
23. Shao, Z.; Wang, L.; Wang, Z.; Du, W.; Wu, W. Saliency-aware convolution neural network for ship detection in surveillance video. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 781–794. [CrossRef]
24. He, Z.; Huang, L.; Zeng, W.; Zhang, X.; Jiang, Y.; Zou, Q. Elongated small object detection from remote sensing images using hierarchical scale-sensitive networks. *Remote Sens.* **2021**, *13*, 3182. [CrossRef]
25. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*; Pereira, F., Burges, C.J.C., Bottou, L., Eds.; Association for Computing Machinery (ACM): New York, NY, USA, 2012; pp. 1097–1105.
26. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
27. Law, H.; Teng, Y.; Russakovsky, O.; Deng, J. CornerNet-Lite: Efficient Keypoint Based Object Detection. *arXiv* **2020**, arXiv:1904.08900.
28. 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.
29. Li, J.; Liang, X.; Wei, Y.; Xu, T.; Feng, J.; Yan, S. Perceptual Generative Adversarial Networks for Small Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1951–1959.
30. Geirhos, R.; Rubisch, P.; Michaelis, C. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
31. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
32. Bell, S.; Zitnick, C.L.; Bala, K.; Girshick, R. Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2874–2883.
33. Kong, T.; Yao, A.; Chen, Y.; Sun, F. HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 845–853.
34. 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, 18–22 June 2018; pp. 8759–8768.
35. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J. Microsoft COCO: Common Objects in Context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.