Journal of *Marine Science and Engineering*

MDPI

Article

YOLOv7-Ship: A Lightweight Algorithm for Ship Object Detection in Complex Marine Environments

Zhikai Jiang [1], Li Su [1,2,*] and Yuxin Sun [1]

[1] College of Intelligent Science and Engineering, Harbin Engineering University, Harbin 150001, China; jzkjzk@hrbeu.edu.cn (Z.J.); heu_syx@hrbeu.edu.cn (Y.S.)
[2] Key Laboratory of Ministry of Education on Intelligent Technology and Application of Marine Equipment, Harbin Engineering University, Harbin 150001, China
* Correspondence: suli406@hrbeu.edu.cn

Abstract: Accurate ship object detection ensures navigation safety and effective maritime traffic management. Existing ship target detection models often have the problem of missed detection in complex marine environments, and it is hard to achieve high accuracy and real-time performance simultaneously. To address these issues, this paper proposes a lightweight ship object detection model called YOLOv7-Ship to perform end-to-end ship detection in complex marine environments. At first, we insert the improved "coordinate attention mechanism" (CA-M) in the backbone of the YOLOv7-Tiny model at the appropriate location. Then, the feature extraction capability of the convolution module is enhanced by embedding omnidimensional dynamic convolution (ODconv) into the efficient layer aggregation network (ELAN). Furthermore, content-aware feature reorganization (CARAFE) and SIoU are introduced into the model to improve its convergence speed and detection precision for small targets. Finally, to handle the scarcity of ship data in complex marine environments, we build the ship dataset, which contains 5100 real ship images. Experimental results show that, compared with the baseline YOLOv7-Tiny model, YOLOv7-Ship improves the mean average precision (mAP) by 2.2% on the self-built dataset. The model also has a lightweight feature with a detection speed of 75 frames per second, which can meet the need for real-time detection in complex marine environments to a certain extent, highlighting its advantages for the safety of maritime navigation.

Keywords: ship detection; coordinate attention; CARAFE; SIoU; Yolov7

Citation: Jiang, Z.; Su, L.; Sun, Y. YOLOv7-Ship: A Lightweight Algorithm for Ship Object Detection in Complex Marine Environments. *J. Mar. Sci. Eng.* **2024**, *12*, 190. https://doi.org/10.3390/jmse12010190

Academic Editor: Fausto Pedro García Márquez

Received: 20 December 2023
Revised: 13 January 2024
Accepted: 17 January 2024
Published: 20 January 2024

1. Introduction

With the rapid development of marine equipment, the requirements for the accurate and reliable identification of ship object detection are increasing [1]. Maritime enforcement officers can access visual information intuitively through maritime surveillance videos. However, supervisors may be affected by the complex environment and visual fatigue, which can cause them to overlook important information and pose a risk of safety hazards to ships traveling at sea. With the help of image vision and neural network algorithms in deep learning, automatic ship detection has been a critical technology in ship applications, significant in marine monitoring, port management, and safe navigation. It guarantees the orderly anchoring of ships and the smoothness and safety of maritime traffic [2,3].

Ship target detection technology has flourished under the rapid development of artificial intelligence technology. Ship target detection technology based on deep learning has become popular in the application field because of its better performance and lower workforce cost than traditional ship detection technology. The dataset images used in this technique mainly include remote sensing, SAR, and visible light images. The detection of remote sensing images is easily affected by factors such as cloud cover and light, and due to the vast amount of data, the time for data preprocessing and image transmission is too long, which leads to specific difficulties in real-time detection. SAR images cannot provide

sufficiently rich texture features due to the absence of rich spectra, and it is not easy to accurately provide the classification information of multicategory ships [4]. On the other hand, visible light images have the highest resolution and contain rich feature information, such as details and colors, which can intuitively present real human vision. In addition, visible light images can be easily acquired by standard acquisition devices such as cameras. Therefore, how to detect targets faster and more accurately on visible light ship images has become one of the leading research directions.

Deep-learning-based object detection models are gradually becoming the main research method in visible ship detection. Object detection algorithms can be classified into two types: single-stage and two-stage algorithms. Representative two-stage algorithms are R-CNN [5], faster R-CNN [6], and mask R-CNN [7]. However, they need to generate many candidate regions, which increases the computation and time complexity. The algorithms represented by single-stage algorithms are the single-shot multibox detector (SSD) [8], you only look once (YOLO) [9], and RetinaNet algorithm [10]; these algorithms do not require additional generation of candidate regions, simplifying the detection process with faster detection speed.

The YOLOv7-Tiny model [11] has high speed and efficiency in real-time target detection. Still, it may be limited by the resolution and prone to localization bias when dealing with small targets, and there may be omission or misdetection of partially occluded ship targets. A lightweight ship target detection model that can effectively identify ship positions in complex and changing environments, help ships avoid collisions, and reduce the risk of accidents is significant for maritime navigation safety. Therefore, we propose the YOLOv7-Ship detection model based on YOLOv7-Tiny in this study. A concise overview of the primary contributions of this study is summarized below:

- We introduce the improved CA-M attention module to the YOLOv7 backbone module to weaken the background feature weight, introduce ODconv in the neck, and propose an improved aggregation network module, OD-ELAN, which efficiently enhances the network's feature extraction capacity for ships in complex scenes with less computational increase.
- We use the lightweight CARAFE method in the feature fusion layer, which can utilize learnable interpolation weights to interpolate the low-resolution feature maps, thus reducing the loss while processing small-target ship feature information.
- We adopt SIoU as the loss function, which more accurately captures the orientation-matching information between target bounding boxes and improves the convergence speed of algorithm training.
- We construct a ship target detection dataset containing thousands of accurately labeled visible ship images in complex marine environments.

2. Related Work

Compared with general-purpose target detection, ship target detection is more likely to be affected by unfavorable factors such as complex marine areas and bad weather. In addition, ship targets have significant differences in scale, and their visualization features are more likely to be disturbed. Therefore, the practicality and robustness of ship target detection algorithms are more demanding. Liang et al. [12] proposed a ship target detection method based on SRM segmentation and hierarchical line segment feature extraction to solve the problem of difficulty in analyzing high-resolution ship images. The method uses hierarchical line segment search updating and merges line segments near the subthreshold to achieve the detection of ship targets. Zhu et al. [13] proposed a method based on neighborhood feature analysis for detecting ships on the sea surface. The method analyzes the mean variance product characteristic of the neighborhood window and initially performs segmentation to eliminate most of the sea surface background, and then verifies the detection of the target by ship-related features. Yang et al. [14] proposed a detection algorithm based on saliency segmentation and local binary pattern descriptors combined with ship

structure and used the morphological contrast method to improve the detection accuracy of ship targets on optical satellite images.

With the booming development of computer technology, much research has been conducted on deep-learning-based visible ship object detection technology. Among these, single-stage algorithms have been the mainstream of visible ship object detection. For example, Yang et al. [15] combined the repulsion loss function and soft nonmaximum suppression algorithms with the SSD model, which can effectively reduce the leakage rate of tiny ships. Li et al. [16] combined the adaptively spatial feature fusion (ASFF) module with the YOLOv3 algorithm and used the ConvNeXt module to ameliorate the problem of insufficient feature extraction capability when ships occlude from each other. Huang et al. [17] incorporated a multiscale weighted feature fusion structure into the YOLOv4 model, improving small ships' detection efficiency. Zhou et al. [18] used mixed depthwise convolutional kernels to improve the traditional convolutional operation and coordinated attention mechanism (CA) based on YOLOv5, which enables the model to extract a more comprehensive ship feature while reducing the computation effectively. Gao et al. [19] proposed a lightweight model for small infrared ship detection by replacing the backbone of YOLOv5 with that of Mobilev3, resulting in an 83% reduction in parameters. Wu et al. [20] introduced the multiscale feature fusion module into the YOLOv7 model and established suitable anchor boxes to replace the fixed anchor boxes, effectively improving the ship feature's capture ability. Chen et al. [21] combined the convolutional attention mechanism and residual connectivity into the YOLOv7 model, enabling the model to accurately locate ships in dark environments and achieve effective ship classification detection. Lang et al. [22] proposed LSDNet, a mobile ship detection model that introduces partial convolution in YOLOv7-Tiny to reduce redundant computations and memory accesses, thereby extracting spatial features more efficiently. Xing et al. [2] integrated the FasterNet module into the backbone of YOLOv8n and employed the lightweight GSConv convolution method instead of the traditional convolution module, which retains detailed information about the ship target.

Although there have been many studies on ship detection, their results in complex real-time environments are often unsatisfactory. The above methods are often difficult to balance the high accuracy and speed of ship detection. On the one hand, the ship's target scale varies greatly. It is prone to problems such as multiple targets overlapping, small targets carrying little information, and background interference information such as land buildings, reefs, and buoys. On the other hand, the marine environment is complex and changeable, with frequent fog, rain, snow, sun glare, and other inclement weather [23]. Especially when the image clarity is not enough, the recognition ability of small targets decreases, resulting in severe problems of ship object false detection and missed detection [24]. The detection models currently studied are mainly large-volume models with high requirements for equipment, and there is an urgent need for a lightweight model that can be deployed on low-configuration computing devices to accomplish the detection task of ships in complex scenarios efficiently [3].

3. Methods

3.1. YOLOv7 Network Structure

Wang et al. optimized the network structure, data augmentation, and activation function to propose the YOLOv7 algorithm model in 2022. Its comprehensive performance improves the detection efficiency and accuracy compared with those of the algorithms of YOLOv4 [25], YOLOv5, YOLOX [26], and YOLOv6 [27]. The YOLOv7-Tiny algorithm is a lighter version of the YOLOv7 algorithm, which simplifies the E-ELAN module to the ELAN module and maintains the path aggregation idea. Compared with the original version, it has fewer computational and parameter counts, which improves detection speed at the expense of some accuracy. In particular, the YOLOv7-Tiny model has good compatibility on ship mobile devices and has shown superior performance in detecting small objects, making it well suited for detecting ships, so we chose it as an improved

baseline model. Figure 1 illustrates the architecture of the YOLOv7-Tiny model. Its four main components are the input, backbone, neck, and output.

The backbone mainly consists of CBL layers, ELAN modules, and maximum pooling layers. The ELAN modules are layer aggregation architectures with efficient gradient propagation paths, which can mitigate the gradient vanishing problem. To show the network's simplified effect, we also offer the E-ELAN module used in YOLOv7 in Figure 1. The neck module uses the path aggregation feature pyramid network (PAFPN), which achieves the effect of multiscale learning of different levels of features by fusing the semantic information conveyed by feature pyramid networks (FPNs) [28] from the more profound level and the localization information conveyed by the path aggregation network (PANet) [29] from the shallower level. The output part uses the IDetect detection head, which classifies the detection scale into three scales, including large, medium, and small targets.

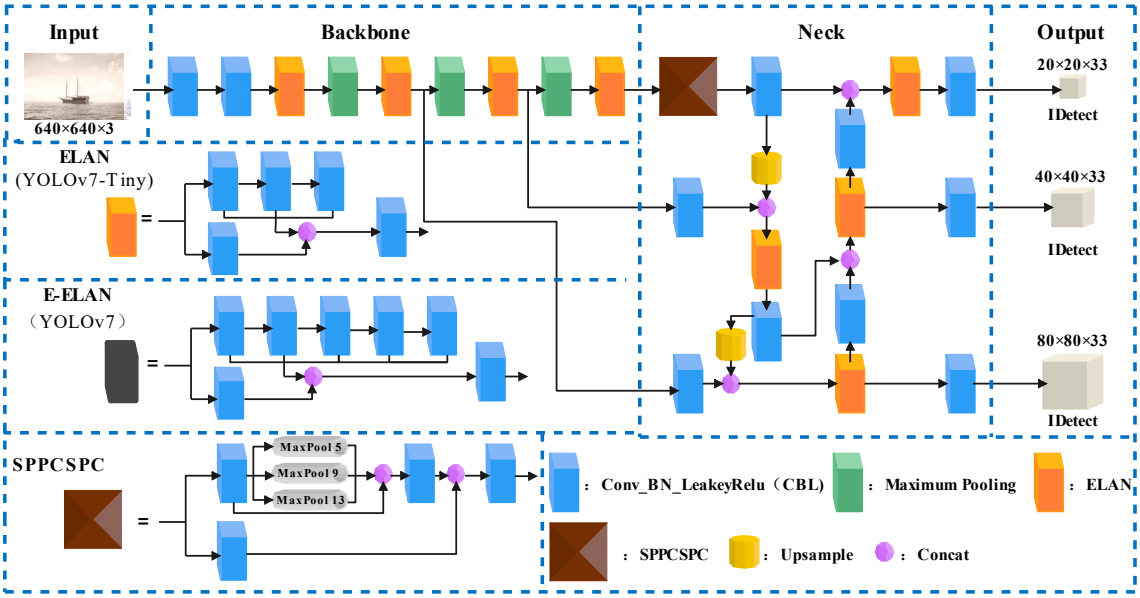


Figure 1. Structure of the YOLOv7-Tiny network.

3.2. OD-ELAN Module

Static convolution convolves the input feature mapping with a constant kernel size. However, due to its fixed weights, it cannot adapt to input data changes and cannot capture global context information. The dynamic convolution method uses a linear combination of kernel weights to perform an attentional weighting operation on the input data. Unlike traditional convolution, the dynamic convolution kernels can automatically resize their receptive field according to the input image information. In addition, the dynamic convolution kernel dynamically generates different weights at each position, significantly reducing the computational complexity and memory utilization. Current dynamic convolution techniques like CondConv [30] and DyConv [31] solely concentrate on the dynamic nature of kernel numbers and dynamically weight the convolution kernel to adapt to different inputs only in the two-dimensional plane. Equation (1) provides the definition of the dynamic convolution operation.

Chao Li et al. [32] proposed a new dynamic convolution, ODconv. ODconv utilizes a multidimensional attention mechanism to make the convolution kernel adaptively weight adjustment in four different dimensions of the kernel space, fully utilizing the number of convolutional kernels, spatial size, input channel, and output channel information, with improved multiscale perception and global context information, which is calculated as Equation (2):

$$y = (\alpha_{w1} W_1 + \cdots \alpha_{wn} W_n) * x \quad (1)$$

$$y = \left(\alpha_{w1} \odot \alpha_{f1} \odot \alpha_{c1} \odot \alpha_{s1} \odot W_1 + \cdots + \alpha_{wn} \odot \alpha_{fn} \odot \alpha_{cn} \odot \alpha_{sn} \odot W_n\right) * x \quad (2)$$

where the input and output features are denoted by the symbols x and y, respectively; the symbol W_i represents the ith convolutional kernel, while α_{wi} serves as the attention scalar for W_i; for the convolutional kernel W_i, the attentions α_{ci}, α_{fi}, and α_{si} are assigned along the input channel, output channel, and spatial dimension of the kernel space, respectively; $*$ represents the convolution operation; and $\odot$ represents the multiplications performed along the various dimensions.

ODconv first squeezes the feature x into a feature vector of the same length as the input channel using channel-wise global average pooling (GAP) operation. Next, the squeezed feature vectors are mapped to the low-dimensional space through a fully connected (FC) layer and a rectified linear unit (ReLU). Each of the four head branches has an FC layer and a sigmoid or softmax function that generates the attentions α_{wi}, α_{ci}, α_{fi}, and α_{si}, respectively. Figure 2 displays the ODconv structure.

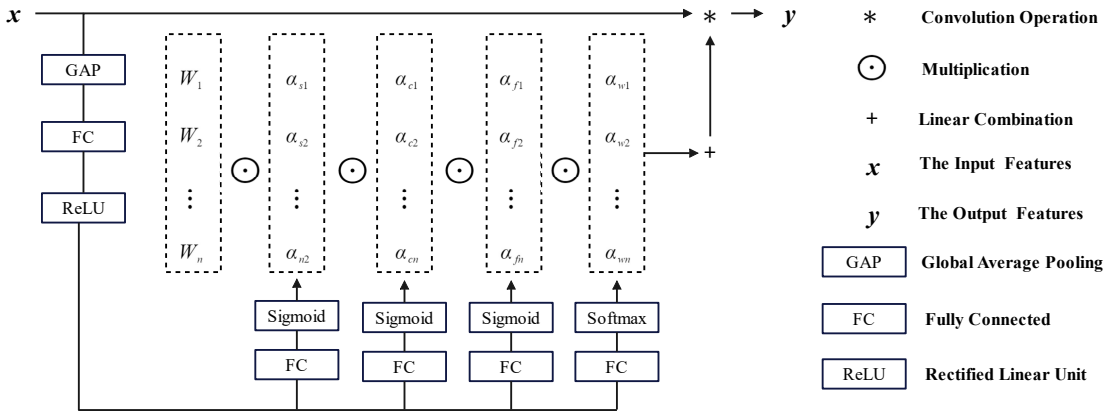


Figure 2. Schematic diagram of the structure of ODconv.

The ELAN network module is used in the YOLOv7-Tiny model to extract target features. The structure consists of convolutional layers, but the smaller number makes it difficult to extract deep target features, and there may be ineffective feature fusion or redundancy. Hence, the module cannot sufficiently extract features from small or low-definition ship targets in a real complex environment. For this reason, we introduce ODconv into the ELAN module and construct an improved ELAN-OD module in the neck part. This module can effectively enhance the mining ability of the network for the deep feature information of ship targets while reducing the computational complexity.

3.3. CA-M Attention Mechanism

To optimize the model's emphasis on the ship's priority edge feature regions, an attention mechanism needs to be introduced into the network to suppress confusing information interference, such as wake and partial occlusion. The traditional channel attention mechanism, such as squeeze-and-excitation networks (SENets) [33], only considers the importance level between feature map channels and ignores the target's location information. The convolutional block attention module (CBAM) [34], which adds a spatial attention mechanism, uses sequential channel and spatial attention operations. However, it ignores the interrelationships between channel and space and loses information across dimensions. The CA mechanism can comprehensively analyze the inter-relationship between feature map channels and spatial information [35]. To better enhance the performance of the attention mechanism, this paper proposes the improved CA-M mechanism.

Coordinate attention encodes the decomposition of the channel relationship into one-dimensional features containing precise positional information, and the fusion of features along both spatial directions enables the model to concentrate on an extensive

range of positional features. Since ships have more significant detail features, such as flat hulls, slender masts, straight chimneys, and hull markings, the global average pooling in the original coordinate attention cannot retain the relative differences between the original features. It may blur certain detailed feature information, while the global adaptive maximum pooling takes the maximum value in the input image region as the output, further increasing the network's sensitivity to critical information.

Consequently, we use the adaptive global maximum pooling layer in the coordinate information embedding so that the model can better extract salient features of ships in complex scenes. Figure 3 illustrates the structure of CA-M.

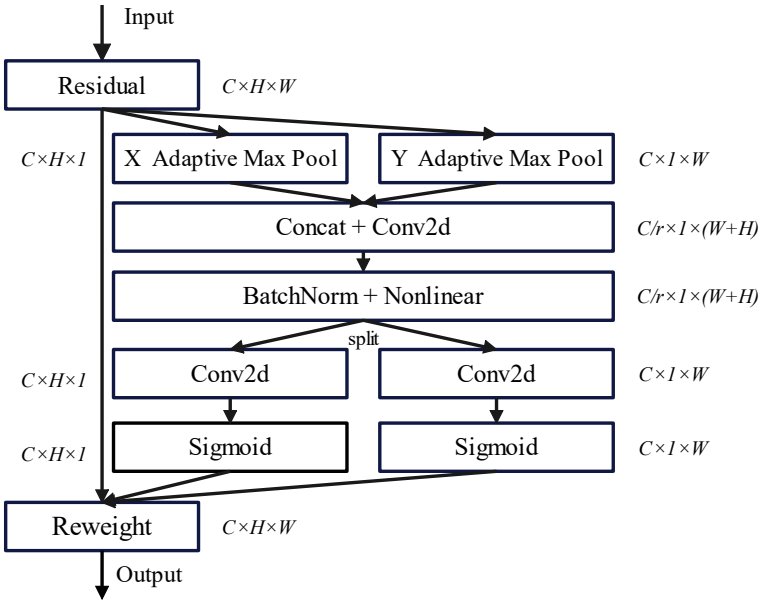


Figure 3. Structure of the improved coordinate attention (CA-M) mechanism.

The first decomposes the global adaptive maximum pooling into two separate 1D feature encoding operations. Subsequently, two spatially scoped ensemble kernels are used to encode each channel in horizontal and vertical coordinates, respectively. Consequently, the output of the cth channel with the vertical dimension h can be mathematically expressed as

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

The output of the c-th channel with the horizontal dimension w can be formulated as

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

The above encoding operation enables the CA-M mechanism to obtain long-range dependency information in one dimension and positional information in another. Then, the above two aggregated feature maps are subjected to the concatenate operation, followed by the transform operation using a shared 1×1 convolutional transform function.

Subsequently, the intermediate feature mapping is split into two distinct tensors along the spatial dimension. These two tensors are then converted into tensors equal to the input. The expanded results g^h and g^w are utilized as attention weights. Ultimately, the output of the coordinate attention block **Y** can be denoted as

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j) \tag{5}$$

Compared with the original CA mechanism, adding the CA-M mechanism to the backbone improves mAP@0.5 and mAP@0.5:0.95 by 0.2% and 0.3% in ship detection,

respectively. Compared with other attention mechanisms, the CA-M mechanism can focus more on the areas with high feature weights in the inference and only adds a small amount of computational overhead. Its specific results are shown later in the experimental section.

3.4. CARAFE Upsampler

The CARAFE upsampler is an efficient and lightweight image upsampling algorithm proposed by Wang et al. [36]. It can avoid the problem of nearest neighbor interpolation upsampling algorithms weakening small targets' feature information while bringing computational effort to integrate more feature information in large receptive fields. Small target ship detection is susceptible to complex environmental interference. Thus, we use CARAFE in the neck module, thus replacing the original nearest neighbor interpolation algorithm to extract smaller target features. Figure 4 illustrates the CARAFE upsampler's architecture. It is composed of two modules: the kernel prediction module and the content-aware reassembly module.

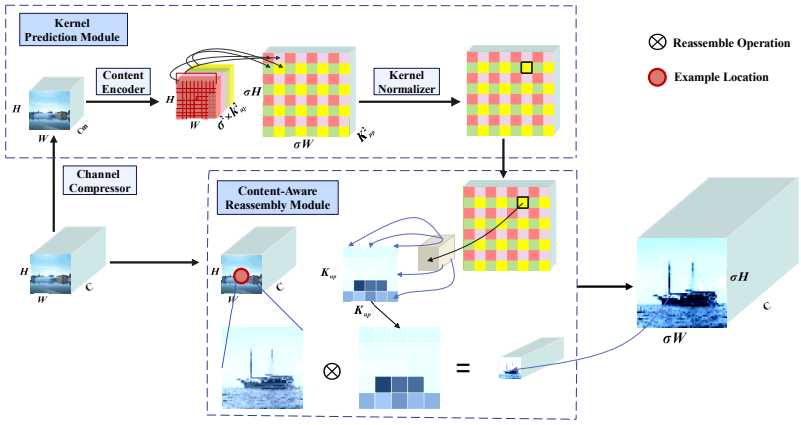


Figure 4. The overall structure of CARAFE.

First is the kernel prediction module, where the feature map with an input size (H, W, C) is compressed by a 1×1 convolution, and then a convolution layer of the kernel size $K_{encoder} = k_{up} - 2$ is used to predict the upsampling kernel to generate features $(\sigma H, \sigma W, k_{\text{up}}^2)$, where σ and k_{up} indicate the upsampling ratio and the reorganization kernel size, respectively. Subsequently, the channels undergo spatial dimension expansion, and the softmax function is employed to normalize the upsampling kernel.

Next, every position in the output feature map is mapped back into the input feature map during the content-aware reassembly processes, taking the $k_{\text{up}} \times k_{\text{up}}$ feature region at its center and the predicted upsampled convolution kernel at that position to make a dot product. Finally, the new features (shape = (σH, σW, C)) are obtained by repeating the above operations.

3.5. SIoU Loss Function

The YOLOv7-Tiny model contains three loss functions: classification loss, confidence loss, and localization loss. A practical bounding box loss function is essential for target localization. By default, YOLOv7-Tiny employs the complete intersection over union (CIoU) [37] localization loss function, which considers three distinct factors, and its calculation formula is displayed below:

$$L_{CIoU} = 1 - I_{IoU} + \frac{\rho^2\left(B, B^{gt}\right)}{c^2} + av \tag{6}$$

where B and B^{gt} represent the centroids of the prediction box and the ground truth box, respectively. The value of Euclidean distance between B and B^{gt} is denoted by $\rho\left(B, B^{gt}\right)$. Additionally, c indicates the minimum outer rectangle's diagonal length necessary to encompass both boxes. The consistency of the aspect ratio v is calculated as

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

where $\frac{w^{gt}}{h^{gt}}$ and $\frac{w}{h}$ denote the aspect ratios of the prediction box and the ground truth box, respectively. α is the weight coefficient, which is calculated by the formula

$$\alpha = \frac{v}{(1 - I_{IoU}) + v} \tag{8}$$

The CIoU loss function introduces $\frac{w^{gt}}{h^{gt}}$ and $\frac{w}{h}$ into the loss value calculation. It adds a penalty term, which effectively improves the degradation problem of the GIoU [38] loss function and addresses the challenge posed by the DIoU [39] loss function when the prediction box does not overlap with the ground truth box but still gives the bounding box a moving direction. However, the CIoU loss function does not consider the angular mismatch between the ground truth box and the prediction box. It can be seen from Equation (7) that when $\frac{w^{gt}}{h^{gt}}$ and $\frac{w}{h}$ are the same, v takes 0, at which time the penalty term fails, leading to large fluctuations in the convergence of the training and a lack of precision in the prediction box.

Gevorgyan [40] proposed the SIoU loss, which also investigated the orientation matching problem between the prediction box and the ground truth box based on the consideration of the distance between the frame centers, the aspect ratio, and the overlap area. Additionally, it added an angle cost term. In this paper, we adopt this efficient bounding box regression loss function SIoU, which effectively improves the total degrees of freedom of the loss and penalty terms, further increasing the training convergence performance of the model so that the target box has a better regression localization accuracy [41]. The formula of SIoU is shown below.

$$\Lambda = 1 - 2 \times \sin^2\left(\arcsin\left(\frac{C_h}{\sigma}\right) - \frac{\pi}{4}\right) \tag{9}$$

$$\Delta = \sum_{t=x,y}\left(1 - e^{-(2-\Lambda)\rho_1}\right) \tag{10}$$

$$\Omega = \sum_{t=w,h}\left(1 - e^{-w_t}\right)^{\theta} \tag{11}$$

In Equations (9)–(11), Λ is the angle cost, Δ is the distance cost considering Λ, Ω is the shape cost, and θ determines the level of concern for shape loss. As shown in Figure 5, C_h and σ are the height difference and geometric distance between B and B^{gt}, respectively.

The definition of the final bounding box loss SIoU is

$$L_{SIoU} = 1 - I_{IoU} + \frac{\Delta + \Omega}{2} \tag{12}$$

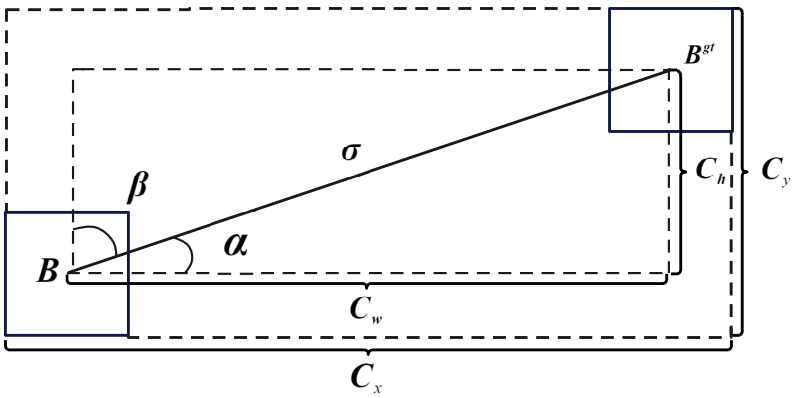


Figure 5. Schematic diagram for calculating the angular cost contribution in the loss function.

3.6. The YOLOv7-Ship Model

The original YOLOv7-Tiny model has a fast detection rate. However, in complex marine environments, such as in severe weather and illumination conditions, or when the ship is multiscale or partially obscured, the model may have the problem of false or missed detection. For this reason, we propose the YOLOv7-Ship model, which optimizes the network structure of the baseline model while keeping it lightweight. We first consider the performance advantages of ODconv and insert the proposed OD-ELAN module in the neck network. The structure of the OD-ELAN module acquires the feature in different dimensions, which is illustrated in Figure 6. It utilizes the dynamically changing convolution kernel structure to achieve learning of multidimensional feature information. Second, to further enhance the backbone network's feature extraction capacity for targets with minimal information or clarity, we add the CA-M module in the backbone network. The CA-M module allows the network to concentrate on the linkage of the ship's salient detail features between space and channel, suppressing the irrelevant interfering information and efficiently extracting the critical location information of the ship object detection.

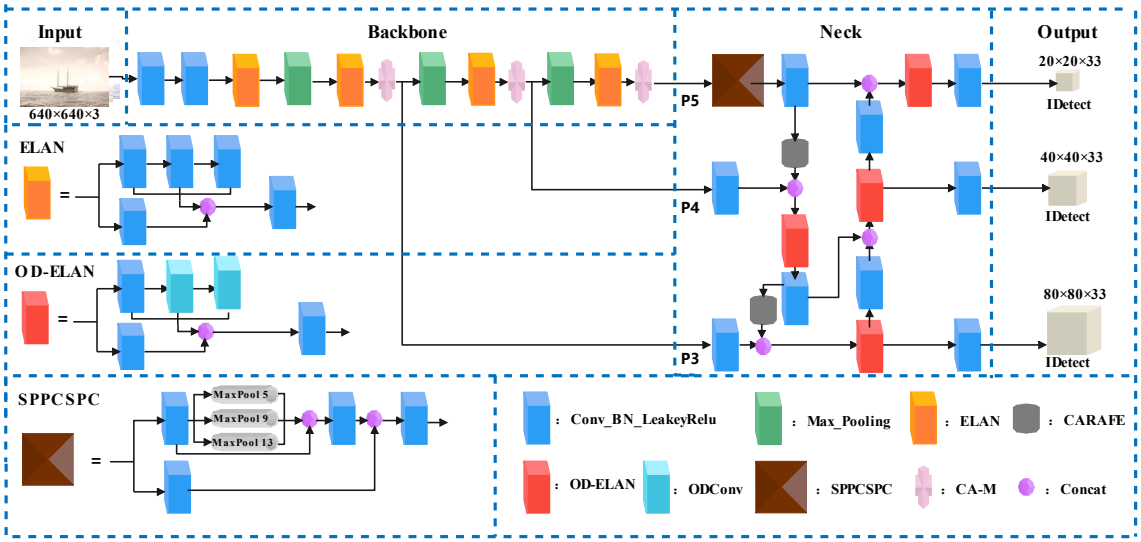


Figure 6. Structure of the YOLOv7-Ship model.

Next, to address the issue of the model's missing detection for multiscale targets and low-resolution images, we use CARAFE as the upsampling operator in the feature fusion network. CARAFE utilizes cross-scale feature information fusion, which predicts the upsampling kernel and reorganizes the features based on it to better retain the semantic information of the original picture, effectively reducing the loss of feature information processing for small targets. CARAFE can also adjust the parameters through training to obtain better upsampling results. Finally, we employ the SIoU loss function, which can better handle the box regression problem between targets at different scales to capture the directional matching information between the target bounding boxes. The YOLOv7-Ship model's structure is displayed in Figure 6.

The training and validation process of the YOLOv7-Ship model is relatively simple, but the hyperparameters must be carefully tuned to ensure the generalization performance on the dataset. The YOLOv7-Ship model is designed with cross-platform support in mind and has good portability. It can run on different operating systems and supports a variety of hardware accelerators, such as GPU and CPU. However, the YOLOv7-Ship model relies on more powerful hardware and may suffer from some performance limitations on low-end hardware.

4. Experiments

In this section, we present the dataset construction and the experimental design part. Specifically, we first introduce a new self-constructed ship dataset designed for studying the ship target detection problem in complex scenarios. Then, we report our experimental platform setup and training details and provide comprehensive evaluation metrics for the target detection task.

4.1. Data Collection and Processing

The complex environment around ships in the natural environment, such as land buildings, reefs, buoys, and other background interference information, is prone to cause interference to the target detection of ships. Existing public ship datasets, such as the SeaShips dataset proposed by Shao et al. [42], contain 7000 images of ship detection in six categories. However, the majority of the photos in this dataset are a collection of shots taken of the same ship at nearby moments, and the data scene is single and little affected by interference information. Therefore, we constructed a ship dataset containing more complex marine scenes to enhance the algorithm's detection accuracy and generalization ability under interference conditions such as bad weather, multiple occlusions, and small targets.

The images in this dataset were self-collected by the team at sea using a visible light camera, supplemented by adding some images from the publicly available ship dataset. In this paper, data cleaning was performed on the collected images. After deleting the damaged or blurred images, the images that meet the requirements were used as the labeled dataset, which contains diversity-rich environments, such as a harbor with heavy traffic, a fishery area with dense ships, and a mixed traffic scene between ships and shore. Most images also have different climatic interferences, such as solid illumination, rain, and snow, for 5100 original images. Selected images of some sample datasets are shown in Figure 7.

(**a**) (**b**) (**c**)

Figure 7. Display of ship detection dataset: (**a**) sailboat, (**b**) container ship, and (**c**) small target ship.

We categorized the target labels into six groups based on the Pascal VOC dataset format: sailboat, island reef, container ship, linear, and other ships. The sample images were labeled sequentially using the LabelImg software to generate XML files, which were then converted to YOLO format. For experimental purposes, we arbitrarily partitioned the dataset into three sets: training, validation, and test, in the following proportion: 8:1:1. Table 1 shows the number of images in the dataset under different weather and light conditions.

Table 1. Distribution of ship images in the dataset.

Condition	Training	Validation	Test	Total	Percentage
Sunny	2522	316	319	3157	61.9%
Rainy, foggy, and snowy	1021	128	140	1289	25.3%
Dusk or darkness	537	66	51	654	12.8%

This dataset uses the Microsoft COCO dataset's method of defining scales. Table 2 shows the definition of different object scales. Table 3 shows the number of objects labeled as small, medium, and large for each category in the dataset. The total number of small

objects is 7737 (38.2%), the total number of medium objects is 6975 (34.5%), and the total number of large objects is 5528 (27.3%).

Table 2. The definitions of small, medium, and large objects in the COCO dataset.

Objects	Metric (Square Pixels)
Small	Area < 32^2
Medium	32^2 < Area < 96^2
Large	Area > 96^2

Table 3. Statistics of the number of small, medium, and large objects of six label types.

Category	Small	Medium	Large	Total
Liner	735	150	42	927
Container ship	362	20	8	390
Bulk carrier	1338	308	100	1746
Island reef	1926	1869	468	4263
Sailboat	1339	1011	755	3105
Other ship	2037	3617	4155	9809

Figure 8 illustrates the distribution of the dataset's bounding boxes, including their center points and sizes. Figure 8a depicts the normalized bounding boxes' center coordinate distribution. Figure 8b illustrates the proportions of the labeled box's width and height to the original figure. It is evident that the overwhelming majority of our dataset comprises small objects, and most targets are mainly concentrated in the central region, which indicates that these features make the dataset well suited for detecting small and multiscale ship targets.

To avoid overfitting, we used the basic methods of random brightness, horizontal flipping, cropping images, and so on for the dataset images. We also used the mosaic data enhancement method, which not only enriches and expands the original detection dataset but also reduces the occupancy of GPU video memory. The input to the model after a series of operations is shown in Figure 9. The numbers 1 to 6 represent liners, container ships, bulk carriers, islanders, sailing ships, and other ships, respectively.

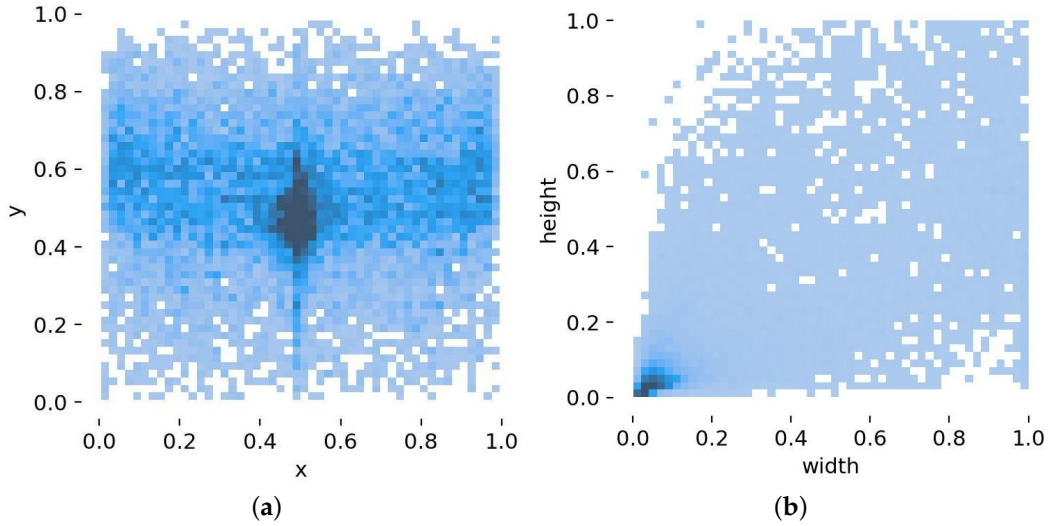


Figure 8. Dataset visualization and analysis results: (**a**) distribution of dataset object centroid locations and (**b**) distribution of dataset object sizes.

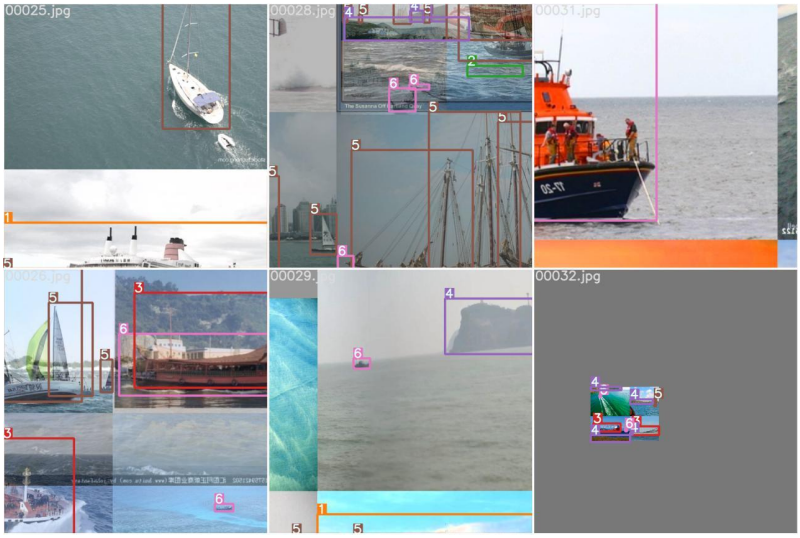


Figure 9. Renderings of data enhancements.

In addition to the data enhancement mentioned above, this paper employs standard methods such as early stopping, dropout, and batch normalization during the training phase to prevent model overfitting.

4.2. Experimental Environment

All experiments in this article were performed on the Windows 10 system. The system utilized an Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz and NVIDIA GeForce RTX 2060. The model was built based on the programming language Python 3.9 and the deep learning framework PyTorch 2.0.0. The specific configuration information of the experimental platform is outlined in Table 4.

The experiment employed an input image size of 640 × 640 pixels and conducted 200 training epochs. The momentum was set to 0.8, and the initial learning rate was set to 0.01. Detailed experimental parameters for model training are shown in Table 5.

Table 4. The configuration information of the experimental platform.

Configuration	Versions
Operation system	Windows 10
CPU	Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz
GPU	NVIDIA GeForce RTX 2060
RAM	16.0 GB
Toolkit	CUDA 11.7
Compiler	Python 3.9
Framework	PyTorch 2.0.0

Table 5. Experimental parameters of model training.

Component	Name/Value
Epochs	200
Image size	640 × 640
Batch size	8
Initial learning rate	0.01
Final learning rate	0.1
Momentum	0.8
Optimizer	SGD
Mosaic	0.9
Mixup	0.05
Copy_paste	0.05

4.3. Evaluation Metrics

To evaluate the quality of the ship target recognition and detection results more comprehensively, precision (P), recall (R), mean average precision (mAP), and frames per second (FPS) evaluation metrics are used in this paper, as shown in Equations (13)–(16).

$$P = \frac{TP}{TP + FP} \tag{13}$$

$$R = \frac{TP}{TP + FN} \tag{14}$$

$$mAP = \frac{1}{m}\sum AP(i) \tag{15}$$

$$FPS = \frac{n}{T} \tag{16}$$

In Equations (13)–(16), the variables denoted as TP, FP, and FN correspond to the quantity of true-positive, false-positive, and false-negative samples, respectively. AP means the average recognition accuracy for a single category whose definition is the area under the P–R curve, m represents the number of detected categories, and AP@0.5 indicates the average precision across different objects when the intersection over union (IoU) threshold is set to 0.5. AP_S, AP_M, and AP_L are used to evaluate the detection performance of the model for small, medium, and large targets, respectively. n represents the quantity of images the model processes, while T denotes the time required for consumption. In addition, we use the number of parameters (Params) and floating point operations (FLOPs) to measure the computational space and time complexity of the model.

5. Results and Analysis

5.1. Effectiveness of CA-M Module

Distinct parts of the Yolov7 network have different extraction roles for the input features. In order to optimize the attention mechanism's optimal placement, we inserted the CA-M module before the backbone's three feature layers for combination. Figure 10 illustrates the display diagram of different positions inserted by the CA-M module. The results of the experiments we subsequently performed on the models utilizing these six methods are presented in Table 6.

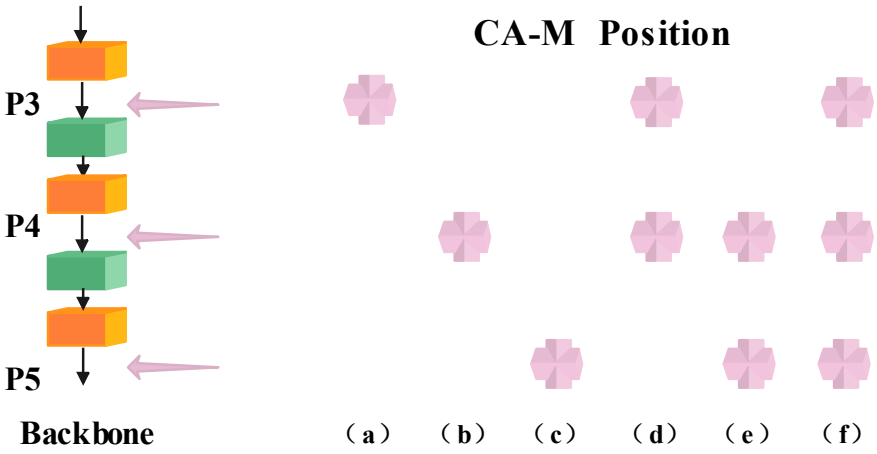


Figure 10. Different positions for inserting the CA-M module in the backbone of YOLOv7-Tiny.

Table 6. Detection effects of inserting the CA-M module at different positions.

Model	P/%	R/%	AP@.5/%	AP@.5:.95/%
(**a**)	80.7	74.2	78.6	53.7
(**b**)	81.1	75.4	78.7	53.7
(**c**)	78.8	76.7	78.1	52.5
(**d**)	80.4	75.1	78.6	53.4
(**e**)	79.9	76.2	78.4	53.7
(**f**)	80.2	75.3	78.9	53.9

The findings in Table 6 indicate that inserting CA-M at any position does not always enhance the network's detection performance. Model (**c**) introduces the CA-M module alone after the last ELAN in the backbone part, and AP@.5 decreased to 78.1, with the worst effect; Model (**f**) introduces CA-M at the position before all three feature layers, and AP@.5 and AP@.5:.95 improved to 78.9 and 53.9, respectively, with the highest accuracy improvement. From this, we can analyze that the use of the CA-M module after the efficient aggregation network module during the initial stage of feature extraction can capture the information of the region of interest, weaken the interference of the pseudo-target feature information, and improve the network's ability to detect the detailed features of the ship's target.

To assess the efficacy of introducing the CA-M module in the baseline model, we compare it with six different types of attention mechanisms introduced into the same location, namely, SE, CBAM, ECA [43], GAM [44], SimAM [45], and CA, and their outcomes are displayed in Table 7.

Table 7. Comparison experiments of different attention modules.

Model	P/%	R/%	AP@.5/%	AP@.5:.95/%
YOLOv7-Tiny	80.1	74.1	78.3	53.5
+SE	80.4	73.4	78.4	51.6
+CBAM	80.8	72.9	78.2	51.9
+ECA	80.3	74.2	78.6	52.2
+GAM	79.3	72.0	77.1	49.9
+SimAM	80.9	73.8	78.7	52.4
+CA	79.9	74.8	78.7	53.6
+CA-M	80.2	75.3	78.9	53.9

Table 7 shows that the effects of introducing various attention mechanisms into the backbone part differ. SE, CBAM, and ECA are all channel attention mechanisms, and their AP@.5:.95 values are reduced by 1.9%, 1.6%, and 1.3%, respectively, compared with the baseline model. In contrast, the GAM global attention mechanism increases the network's sensitivity to local noisy information and tends to cause the overfitting problem, and its AP@.5 is reduced by 1.2% at most. SimAM is a 3D attentional mechanism close to the boosting effect of our proposed CA-M mechanism, and its AP@.5 and AP@.5:.95 are improved by 0.4% and 0.1%, respectively. After introducing the proposed CA-M in the trunk section, AP@.5 and AP@.5:.95 were boosted by 0.6% and 0.4%, respectively. Compared with the introduction of the original CA, the AP@.5 and AP@.5:.95 of CA-M are improved by 0.2% and 0.3%, respectively. Therefore, we introduce the CA-M module to improve the object detection performance.

5.2. Comparative Analysis of Loss Functions

To determine whether the improved loss function can strengthen the model's performance and accelerate convergence, we conducted comparative experiments on the EIoU [46], CIoU, DIoU, GIoU, and SIoU loss functions using YOLOv7-Ship as the baseline model. Figure 11 shows their comparative effects.

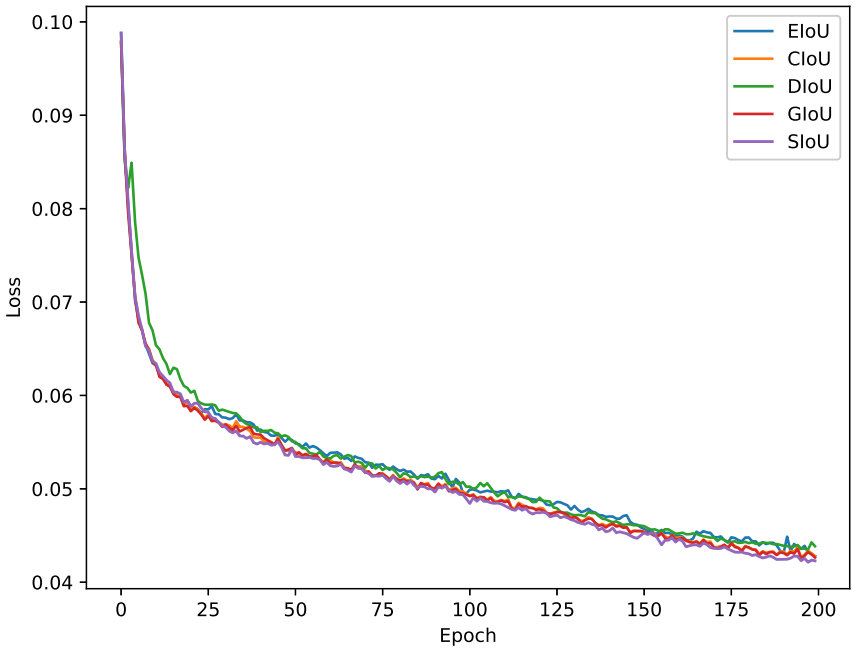


Figure 11. Comparison plot of loss function curves of the model validation set.

By analyzing Figure 11, we found that the model with the SIoU loss function has the fastest reduction in loss values during training. To ensure the integrity of the comparison experiment, we present the analysis outcomes in Table 8, which comprises the loss value and mAP at the 200th epoch.

Table 8. Loss values and mAP for different loss functions.

Model	Loss function	Loss	AP@.5/%
YOLOv7-Ship	CIoU	0.04281	80.3
	EIoU	0.04278	80.2
	DIoU	0.04385	80.1
	GIoU	0.04264	80.4
	SIoU	0.04229	80.5

Using the SIoU loss function compared with the CIoU loss function, the bounding box loss decreases by 0.00052, and mAP improves by 0.2%. SIoU achieves the lowest loss value of 0.04229 and the highest mAP value of 80.5%, showing optimal performance compared with the other loss functions. It also shows that, compared with the YOLOv7-Tiny model, the YOLOv7-Ship model converges faster in training and accurately captures the orientation matching information between the target bounding boxes.

5.3. Ablation Experiment

In order to assess the efficacy of our proposed enhancement approach in optimizing ship inspection performance, we performed a sequence of ablation experiments on our self-constructed ship dataset using YOLOv7-Tiny as the baseline model. The corresponding improvement method is denoted in the table by "✓" if it was implemented and "✕" if it was not. The data of the ablation experiments are shown in Table 9.

Table 9. Comparison experiments of different object detection algorithms.

Model	Group	CA-M	OD-ELAN	CARAFE	SIoU	AP@.5/%	AP@.5:.95/%	GFLOPS	FPS
YOLOv7-Tiny	1	×	×	×	×	78.3	53.5	13.1	77
	2	✓	×	×	×	78.9	53.9	13.1	79
	3	✓	✓	×	×	79.6	54.2	12.7	71
	4	✓	✓	✓	×	80.3	55	12.8	64
	5	✓	✓	✓	✓	80.5	55.4	12.8	75

According to Table 9, the first group of experiments utilized the original YOLOv7-Tiny model, with an AP@0.5 and AP@.5:.95 of 78.3% and 53.5%, respectively. In the second group of experiments, we introduced an improved CA-M attention mechanism in the backbone section. Compared with the baseline model, mAP@0.5 increased by 0.6%. These results indicate that the model's capability to extract pertinent target depth features is enhanced due to the improved network structure.

Subsequently, in the third group of experiments, we introduced ODconv and replaced the efficient aggregation module in the neck section with the OD-ELAN model, which led to a further increase in mAP@0.5 by 0.7% while reducing GFLOPS by 0.4 M. Next, in the fourth group of experiments, we replaced the upsampling method with the CARAFE method, resulting in another 0.7% increase in mAP@0.5. This suggests that the CARAFE upsampling improvement method can more accurately capture semantic information of images. Finally, we improved the loss function to SIoU in the fifth group, mAP@0.5 was increased by 0.2%, while the detection speed was increased to 75 frames per second.

The comprehensive performance results of the YOLOv7-Tiny and YOLOv7-Ship models are shown in Table 10. The mAP@0.5 and mAP@0.5:0.95 of the YOLOv7-Ship model are 80.5% and 55.4%, which are improved by 2.2% and 1.9%, respectively, compared with the baseline model. In particular, the AP_S value of 37.7% for small target detection is improved by 2.5% compared with the baseline model, resulting in a more accurate identification of small-sized targets. However, the APL value decreased by 0.8%. Additionally, the detection speed of YOLOv7-Ship is maintained at 75 FPS, and GFLOPS are reduced by 0.3 M. In conclusion, the YOLOv7-Ship model greatly improves the detection accuracy of small ship objects in complex marine scenarios while meeting the real-time detection needs of embedded marine equipment and the requirements of a lightweight model. However, the detection performance of large targets at different scales still needs further improvement.

Table 10. Performance evaluation results of YOLOv7-Tiny and YOLOv7-Ship.

Model	P/%	R/%	AP@.5/%	AP@.5:.95/%	AP_S/%	AP_L/%	Params/M	GFLOPS	FPS
YOLOv7-Tiny	80.1	74.1	78.3	53.5	35.2	68.4	6.0	13.1	77
YOLOv7-Ship	81.4	75.8	80.5	55.4	37.7	67.6	6.1	12.8	75

5.4. Comparison Experiment

In this section, we select the two-stage object detection model and other mainstream YOLO series models to conduct comparative analysis experiments on the self-constructed ship dataset. The models contain faster R-CNN, SSD, YOLOv3 [47], YOLOv4, YOLOv5s, YOLOv5m, YOLOv7, YOLOv7-Tiny, and YOLOv8. Table 11 displays the outcomes of the experiments, which were all conducted in the identical training environment.

Table 11. Comparison experiments between YOLOv7-Ship and other object detection algorithms.

Model	AP@.5/%	AP@.5:.95/%	AP_S/%	AP_M/%	AP_L/%	Params/M	GFLOPS	FPS
Faster R-CNN	74.9	51.1	29.8	51.3	64.9	72.1	47.6	21
SSD	72.2	47.1	25.3	48.6	61.3	38.6	28.8	43
YOLOv3	75.1	48.1	27.7	52.1	64.2	12.6	19.9	56
YOLOv4	74.8	50.1	29.1	50.9	64.7	52.5	54	31
YOLOv5s	77.2	52	31.6	53.1	66.4	7.1	13.2	59
YOLOv5m	78	55.8	32.0	54.8	69.2	20.9	47.9	41
YOLOv7-Tiny	78.3	53.5	35.2	55.6	68.4	6.0	13.1	77
YOLOv8	78.5	55.7	31.2	56.3	68.9	3.0	8.1	120
YOLOv7-Ship (Ours)	80.5	55.4	37.7	56.4	67.6	6.1	12.8	75

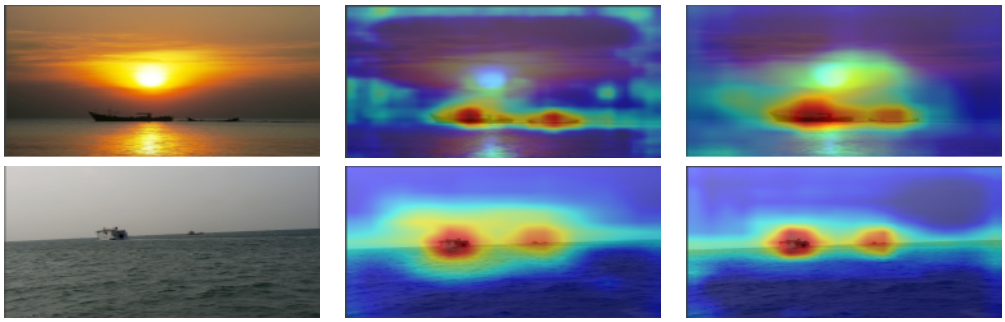
Our proposed model outperforms widely used models in ship target detection. The detection accuracy of faster R-CNN, SSD, and YOLOv4 algorithms is relatively low due to the anchor frame-fixed parameters that cannot be fully adapted to the multiscale ship target. Compared with YOLOv5s and YOLOv3, the YOLOv7-Ship model exhibits significant accuracy gains: 3.2% above YOLOv5s and 5.4% above YOLOv3, while maintaining similar detection speeds. Although YOLOv7-Tiny and YOLOv8 have faster detection speeds (77 and 120 FPS), their detection accuracies remain comparatively modest, 78.3% and 78.5%, respectively. The YOLOv7-Ship model not only achieves the highest detection accuracy but also preserves real-time performance, demonstrating superior overall performance in ship target detection within complex environments compared with similar algorithms.

Our proposed model outperforms widely used models in ship target detection. The faster R-CNN, SSD, and YOLOv4 algorithms have relatively low detection accuracies with an AP@.5 of 74.9%, 72.2%, and 74.8%, respectively, which is due to the anchor frame-fixed parameters that cannot be fully adapted to the multiscale ship target. The AP@.5 of the YOLOv7-Ship model is improved by 5.4% and 3.2% compared with that of YOLOv3 and YOLOv5s, respectively, and demonstrates significant accuracy improvement, while its GFLOPS are reduced by 6.5 and 1 M, respectively. Although YOLOv7-Tiny and YOLOv8 have faster detection speeds (77 and 120 FPS), their AP@.5 values remain comparatively modest, 78.3% and 78.5%, respectively. Compared with similar algorithms, AP_S is the highest at 37.7%, and real-time performance is maintained. YOLOv7-Ship demonstrates superior overall performance in ship target detection within complex environments.

Overfitting is a problem to be aware of in deep learning and may lead to a worse generalization ability of the model. In the experimental preparation phase above, we have taken many regularization methods. From the experimental results, we found that the training and testing errors of the YOLOV7-Ship model decreased synchronously with the increase in training rounds, so the model did not suffer from overfitting problems during training.

5.5. Analysis of the Detection Results

In this section, we employ the Grad-CAM visualization to assess the performance of the YOLOv7-Ship model in ship detection [48]. Grad-CAM is a technique used to visualize the degree of contribution to the prediction results. We randomly chose three images from the ship dataset and used Grad-CAM to visualize the output features of the YOLOv7-Tiny and the YOLOv7-Ship models. The computational results of the corresponding hidden layer feature maps are shown in Figure 12.



(**a**) (**b**) (**c**)

Figure 12. Visualization results of the Grad-CAM feature heat map: (**a**) original image, (**b**) feature heat map of the YOLOv7-Tiny model, and (**c**) feature heat map of the YOLOv7-Ship model.

Through the heat map image, we can intuitively observe that the YOLOv7-Ship model focuses on the critical features of the ship, especially for the parts of small targets, which showcases the effectiveness of our suggested approach in enhancing the precision and accuracy of ship object detection.

5.6. Qualitative Analysis of Detection Effects

In this section, we experiment the YOLOv7-Ship model with other models on a self-constructed ship dataset. Images from three different scenarios were selected for the experiments, from top to bottom: partially occluded multiship detection, small ship detection, and harbor ship detection. The visual detection results of YOLOv5s, YOLOv5m, YOLOv7-Tiny, YOLOv8, and YOLOv7-Ship are presented in Figure 13.

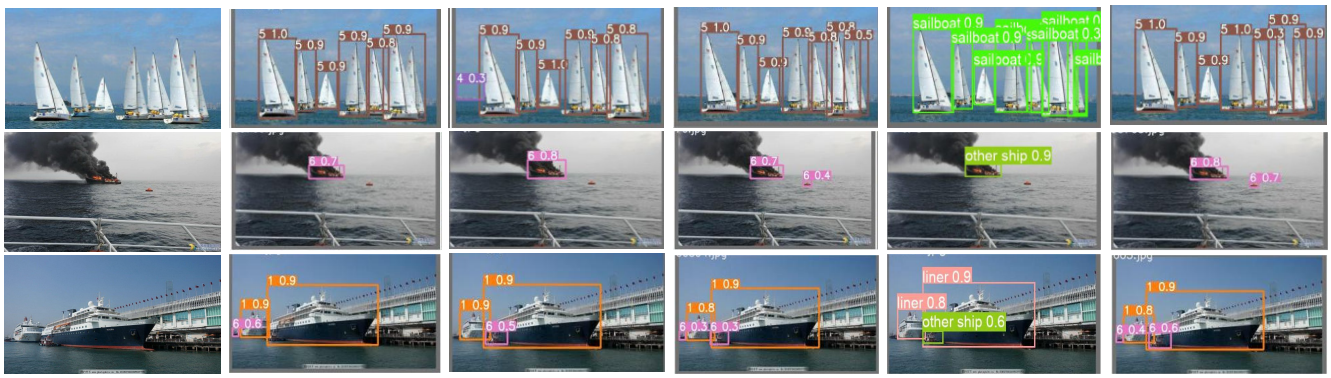


Figure 13. Visual comparison of the YOLOv7-Ship model with the YOLO family.

Figure 13 illustrates that under complex conditions, most models yield unsatisfactory ship detection results due to frequent instances of both leakage and misdetection. YOLOv5s cannot recognize the obscured ship target during partial occlusion, resulting in missed detection. Small target detection is also a challenge, as YOLOv5s, YOLOv5m, and YOLOv8 fail to detect small target ships due to the limited extractable features, which are susceptible to interference from waves and water reflections. In intricate harbor settings, where ships often exhibit multiscale dimensions, numerous pseudo-targets, and background interference, detection difficulty is amplified. However, the YOLOv7-Ship model excels in detecting partially occluded ship targets and accurately identifies small ship targets even at considerable distances. In summary, the YOLOv7-Ship model can detect ship targets more accurately and reduce the ship object detection miss rate in complex environments characterized by multiscale dimensions, high noise levels, and small targets.

6. Conclusions and Discussions

This paper proposes an improved YOLOv7-Ship model, which can accurately detect ship targets in complex marine environments. First, we introduced an improved CA-M attention mechanism after each aggregated network module of backbone, which weakens the interference of irrelevant background noise. Next, we introduced the OD-ELAN module in the neck part, which significantly improves the information mining ability of the detected targets in space and depth. Then, we improved the upsampling method to the CARAFE algorithm, which increases the network sensory field and retains more detailed semantic information. Subsequently, we used SIoU in the loss function part, and the training convergence of the YOLOv7-Ship model was further accelerated. In addition, we have self-constructed a ship dataset in complex environments, aiming to promote the research and development of maritime safety. Experimental results showed that the YOLOv7-Ship model improves the average detection accuracy by 2.2% compared with the baseline model on the self-built ship dataset and adds the computation and parameters only slightly. As a result, the YOLOv7-Ship model provides better detection accuracy for multiscale, partially obscured, and small vessel targets, helping mariners to provide more accurate and comprehensive detection information.

The model proposed in this paper preliminarily achieves the detection of ships in complex marine environments, but there are still the following deficiencies:

1. The research in this paper is limited to the algorithm level, and the algorithm has not yet been deployed to the embedded computing platform.
2. In this paper's self-constructed dataset, there is an imbalance in the category labels. The number of liner and container ship labels is small, leading to insufficient feature extraction and model training for these two categories. In addition, the virtual dataset may not be able to fully simulate the actual scenario, which may lead to the performance degradation of the model in real applications.
3. Although the YOLOv7-Ship model improves the accuracy on small targets, there are still problems of ship missed detection in foggy and dark day scenarios.
4. Compared with the latest YOLOv8 model, the network structure of the YOLOv7-Ship model is more complex and requires more computational resources in the inference stage.

Therefore, our future research directions and work include the following:

1. We plan to design a complete object detector embedded system so that it can execute the YOLOv7-Ship model.
2. We will expand the dataset by including more images of real scenes in different environments and improve the problem of unbalanced category labeling by data augmentation and resampling.
3. We will investigate defogging algorithms and multimodal information fusion techniques. We plan to fuse multisource information from infrared or radar into the model to enhance perception in different environments.
4. We will consider using a more lightweight network structure and employ methods such as pruning and knowledge distillation to reduce the number of model parameters. We aim to maintain the higher detection accuracy of the model while compensating for its shortcomings in detection speed.
5. We will also explore applying the optimized algorithms to complex tasks such as ship object tracking and ship trajectory planning to offer more dependable technical support for realizing intelligence and safety in the maritime field.

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

Funding: This research was funded by the Research Project on Key Technologies for the Development of Intelligent Technology Test Ships (Grant No. CJ01N20).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the author Z.J. (jzkjzk@hrbeu.edu.cn) upon request.

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

References

1. Zhou, W.; Peng, Y. Ship detection based on multi-scale weighted fusion. *Displays* **2023**, *78*, 102448. [CrossRef]
2. Xing, B.; Wang, W.; Qian, J.; Pan, C.; Le, Q. A Lightweight Model for Real-Time Monitoring of Ships. *Electronics* **2023**, *12*, 3804. [CrossRef]
3. Zhang, M.; Rong, X.; Yu, X. Light-SDNet: A Lightweight CNN Architecture for Ship Detection. *IEEE Access* **2022**, *10*, 86647–86662. [CrossRef]
4. Xu, F.; Liu, J.; Sun, M.; Zeng, D.; Wang, X. A hierarchical maritime target detection method for optical remote sensing imagery. *Remote Sens.* **2017**, *9*, 280. [CrossRef]
5. 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.
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
7. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
8. 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 Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
9. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
10. 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, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
11. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
12. Liang, Q.; Dong, W.; Kai, C.L.; Wei, W.; Liang, D. Ship target detection method based on SRM segmentation and hierarchical line segment features. In Proceedings of the 2019 Chinese Control and Decision Conference (CCDC), Nanchang, China, 3–5 June 2019. [CrossRef]
13. Zhu, C.; Zhou, H.; Wang, R.; Guo, J. A Novel Hierarchical Method of Ship Detection from Spaceborne Optical Image Based on Shape and Texture Features. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 3446–3456. [CrossRef]
14. Yang, F.; Xu, Q.; Li, B. Ship Detection from Optical Satellite Images Based on Saliency Segmentation and Structure-LBP Feature. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 602–606. [CrossRef]
15. Yang, Y.; Chen, P.; Ding, K.; Chen, Z.; Hu, K. Object detection of inland waterway ships based on improved SSD model. *Ships Offshore Struct.* **2023**, *18*, 1192–1200. [CrossRef]
16. Li, D.; Zhang, Z.; Fang, Z.; Cao, F. Ship detection with optical image based on CA-YOLO v3 Network. In Proceedings of the 2023 3rd International Conference on Frontiers of Electronics, Information and Computation Technologies (ICFEICT), IEEE, Yangzhou, China, 26–29 May 2023; pp. 589–598. [CrossRef]
17. Huang, Q.; Sun, H.; Wang, Y.; Yuan, Y.; Guo, X.; Gao, Q. Ship detection based on YOLO algorithm for visible images. *IET Image Process.* **2023**. [CrossRef]
18. Zhou, S.; Yin, J. YOLO-Ship: A Visible Light Ship Detection Method. In Proceedings of the 2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE), IEEE, Guangzhou, China, 14–16 January 2022; pp. 113–118. [CrossRef]
19. Gao, Z.; Zhang, Y.; Wang, S. Lightweight Small Ship Detection Algorithm Combined with Infrared Characteristic Analysis for Autonomous Navigation. *J. Mar. Sci. Eng.* **2023**, *11*, 1114. [CrossRef]
20. Wu, W.; Li, X.; Hu, Z.; Liu, X. Ship Detection and Recognition Based on Improved YOLOv7. *Comput. Mater. Contin.* **2023**, *76*, 489–498. [CrossRef]
21. Cen, J.; Feng, H.; Liu, X.; Hu, Y.; Li, H.; Li, H.; Huang, W. An Improved Ship Classification Method Based on YOLOv7 Model with Attention Mechanism. *Wirel. Commun. Mob. Comput.* **2023**, *2023*, 7196323. [CrossRef]

22. Lang, C.; Yu, X.; Rong, X. LSDNet: A Lightweight Ship Detection Network with Improved YOLOv7. *J. Real-Time Image Process.* **2023**. [CrossRef]
23. Er, M.J.; Zhang, Y.; Chen, J.; Gao, W. Ship detection with deep learning: A survey. *Artif. Intell. Rev.* **2023**, *56*, 11825–11865. [CrossRef]
24. Escorcia-Gutierrez, J.; Gamarra, M.; Beleño, K.; Soto, C.; Mansour, R.F. Intelligent deep learning-enabled autonomous small ship detection and classification model. *Comput. Electr. Eng.* **2022**, *100*, 107871. [CrossRef]
25. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934. [CrossRef]
26. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430. [CrossRef]
27. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976. [CrossRef]
28. 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, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
29. 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.
30. Yang, B.; Bender, G.; Le, Q.V.; Ngiam, J. Condconv: Conditionally parameterized convolutions for efficient inference. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
31. Chen, Y.; Dai, X.; Liu, M.; Chen, D.; Yuan, L.; Liu, Z. Dynamic convolution: Attention over convolution kernels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11030–11039.
32. Li, C.; Zhou, A.; Yao, A. Omni-dimensional dynamic convolution. *arXiv* **2022**, arXiv:2209.07947. [CrossRef]
33. 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.
34. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
35. 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.
36. Wang, J.; Chen, K.; Xu, R.; Liu, Z.; Loy, C.C.; Lin, D. Carafe: Content-aware reassembly of features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3007–3016.
37. 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]
38. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
39. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000. [CrossRef]
40. Gevorgyan, Z. SIoU loss: More powerful learning for bounding box regression. *arXiv* **2022**, arXiv:2205.12740. [CrossRef]
41. Zheng, J.; Wu, H.; Zhang, H.; Wang, Z.; Xu, W. Insulator-defect detection algorithm based on improved YOLOv7. *Sensors* **2022**, *22*, 8801. [CrossRef]
42. Shao, Z.; Wu, W.; Wang, Z.; Du, W.; Li, C. Seaships: A large-scale precisely annotated dataset for ship detection. *IEEE Trans. Multimed.* **2018**, *20*, 2593–2604. [CrossRef]
43. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
44. Liu, Y.; Shao, Z.; Hoffmann, N. Global attention mechanism: Retain information to enhance channel-spatial interactions. *arXiv* **2021**, arXiv:2112.05561. [CrossRef]
45. Yang, L.; Zhang, R.Y.; Li, L.; Xie, X. Simam: A simple, parameter-free attention module for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 11863–11874.
46. Zhang, Y.F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* **2022**, *506*, 146–157. [CrossRef]
47. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767. [CrossRef]
48. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.