

Article

CSD-YOLO: A Ship Detection Algorithm Based on a Deformable Large Kernel Attention Mechanism

Tao Wang ¹, Han Zhang ^{2,*} and Dan Jiang ¹

¹ School of Shipping and Marine Engineering, Chongqing Jiaotong University, Chongqing 400074, China; taowang0819@outlook.com (T.W.); jiangdan@cqjtu.edu.cn (D.J.)

² School of Tourism and Media, Chongqing Jiaotong University, Chongqing 400074, China

* Correspondence: hanzhang309@outlook.com

Abstract: Ship detection and identification play pivotal roles in ensuring navigation safety and facilitating efficient maritime traffic management. Aiming at ship detection in complex environments, which often faces problems such as the dense occlusion of ship targets, low detection accuracy, and variable environmental conditions, in this paper, we propose a ship detection algorithm CSD-YOLO (Context guided block module, Slim-neck, Deformable large kernel attention-You Only Look Once) based on the deformable large kernel attention (D-LKA) mechanism, which was improved based on YOLOv8 to enhance its performance. This approach integrates several innovations to bolster its performance. Initially, the utilization of the Context Guided Block module (CG block) enhanced the c2f module of the backbone network, thereby augmenting the feature extraction capabilities and enabling a more precise capture of the key image information. Subsequently, the introduction of a novel neck architecture and the incorporation of the slim-neck module facilitated more effective feature fusion, thereby enhancing both the accuracy and efficiency of detection. Furthermore, the algorithm incorporates a D-LKA mechanism to dynamically adjust the convolution kernel shape and size, thereby enhancing the model's adaptability to varying ship target shapes and sizes. To address data scarcity in complex marine environments, the experiments utilized a fused dataset comprising the SeaShips dataset and a proprietary dataset. The experimental results demonstrate that the CSD-YOLO algorithm outperformed the YOLOv8n algorithm across all model evaluation metrics. Specifically, the precision rate (precision) was 91.5%, the recall rate (recall) was 89.5%, and the mean accuracy (mAP) was 91.5%. Compared to the benchmark algorithm, the Recall was improved by 0.7% and the mAP was improved by 0.4%. These results indicate that the CSD-YOLO algorithm can effectively meet the requirements for ship target recognition and tracking in complex marine environments.

Keywords: deep learning; ship detection; attention mechanism; improve YOLOv8

MSC: 68U10



Citation: Wang, T.; Zhang, H.; Jiang, D. CSD-YOLO: A Ship Detection Algorithm Based on a Deformable Large Kernel Attention Mechanism. *Mathematics* **2024**, *12*, 1728. <https://doi.org/10.3390/math12111728>

Academic Editor: Jonathan Blackledge

Received: 9 May 2024

Revised: 25 May 2024

Accepted: 27 May 2024

Published: 2 June 2024

Correction Statement: This article has been republished with a minor change. The change does not affect the scientific content of the article and further details are available within the backmatter of the website version of this article.



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Waterway transportation plays a crucial role in comprehensive transportation, with a variety of ship types and numbers showing diversification and growth. As this growth continues, waterway transportation also demands a higher accuracy and efficiency in ship inspection. Real-time ship inspection technology based on deep learning has demonstrated significant potential in practical settings. The utilization of this technology is essential for improving the safety of ship navigation and the effective management of ports.

Driven by the rapid advancement of computer technology, deep learning has made substantial strides in target detection. Target detection networks primarily fall into two major categories, as shown in Table 1. The first category comprises two-stage target detection networks, exemplified by architectures such as R-CNN (Region-based Convolutional

Network) [1], Faster-RCNN (Faster Region-based Convolutional Network) [2], and Mask-RCNN (Mask Region-based Convolutional Network) [3]. These models adopt a staged approach to target detection, initially generating potential regions of interest through predefined anchor frames, followed by pinpointing the features of these regions using classification and regression networks. Drawing inspiration from these algorithms, Qi et al. [4] introduced a ship detection algorithm based on an enhanced Faster R-CNN, which improved the ship feature information extraction by reducing the image size, thereby enhancing the model speed and performance in ship detection. Yu et al. [5] proposed a novel algorithm based on an enhanced Faster-RCNN, integrating multiple scale region suggestion, ROI pooling, visual attention mechanisms, and rotating region regression suppression to acquire multi-scale information. Leveraging Res2Net for rich multi-scale information, Chai et al. [6] proposed an improved R-CNN algorithm, demonstrating enhanced accuracy and recall. Lin et al. [7] introduced an enhanced Faster R-CNN network, further enhancing the detection performance through the utilization of squeezing and excitation mechanisms. Subsequently, addressing dense distribution and diverse size constraints, Gou et al. [8] presented an R-Libra CNN method that effectively integrated high-level semantic information with lower-level features. Park et al. [9] proposed a Mask-RCNN-based AI model for warship detection, leveraging MobileNetV2 to achieve model lightweighting. Another class of target detection networks includes SSD (Single Shot MultiBox Detector) [10] and YOLO [11]. These models follow an end-to-end detection approach, combining target localization and classification in a single stage. This design significantly reduces the time needed for target detection and enhances efficiency. Ye et al. [12] introduced an improved attention mechanism in the YOLOv4 algorithm to decrease the missed detections of overlapping vessels while maintaining efficiency. Wang et al. [13] proposed an enhanced YOLOv5 algorithm that decreased overlapping vessels by integrating a sensory field enhancement module and a spatial pyramid fast pooling module, along with adding an attention mechanism to boost the detection speed. Ting et al. [14] addressed the issue of incomplete feature capture in the original YOLOv5 algorithm by merging the feature extraction process with the GhostbottleNet algorithm to account for the uneven distribution of ship image features in lateral and vertical directions. Cen et al. [15] improved the detection speed of overlapping ships by combining the convolutional attention mechanism and residual connection in YOLOv7, enabling the accurate localization of ships in challenging environments with darkness and noise. Xing et al. [16] achieved real-time detection and tracking of ship targets by incorporating FasterNet and GSConv convolution in the YOLOv8 algorithm. Zhu et al. [17] accomplished the real-time detection and tracking of ship targets by optimizing the attention mechanism and adding detection heads to enhance the accuracy in detecting small targets at sea, while Cheng et al. [18] enhanced the accuracy without increasing the network width and depth by incorporating full-dimensional convolution into the YOLOv5 backbone network.

Table 1. Comparison table of existing target detection algorithms.

Algorithm Name	Backbone Network	Specificities
YOLOv3	Darknet-53	Poor detection of small targets
YOLOv4	CSPDarknet53	Larger number of model participants
YOLOv5	CSPDarknet	Poor detection of dense targets
YOLOv7	Darknet-53	Requires greater computational resources
YOLOv8	Transformer-CNN Backbone	Suitable for a wide range of scenarios with high accuracy
R-CNN	AlexNet	Slow detection speed
Faster-RCNN	VGGNet, ResNet, Inception	Requires greater computational resources
SSD	VGGNet, ResNet	False detection in complex situations

Ship detection models currently face challenges such as a large number of parameters, high hardware requirements, low accuracy, and poor real-time performance. The focus of attention is on ensuring detection accuracy while reducing model complexity. More-

over, the complexity and variability of the marine environment present a challenge to ship detection performance. To tackle these issues, this paper introduces the CSD-YOLO ship detection algorithm, which is based on the deformable large kernel attention mechanism. This algorithm enhances the network structure of YOLOv8n, aiming to optimize the model structure, reduce parameters, improve computational efficiency, and accurately identify ship targets in complex environments. Specifically, it performs well under adverse conditions like fog, rain, and darkness. Overall, the proposed algorithm demonstrates excellent performance. The key advancements of this research can be outlined as follows:

- (1) Enhancement of the feature extraction capabilities of the backbone network's c2f module was achieved through the implementation of the CG block module. This advancement enables the model to more accurately capture the crucial information in images.
- (2) To enhance feature fusion and improve the accuracy and efficiency of ship target detection, the neck part was redesigned to incorporate the slim-neck module.
- (3) The D-LKA mechanism, which combines large convolution kernels with deformable convolution, was adopted. It significantly enhanced the model's adaptability to target shapes and sizes by dynamically adapting the shape and scale of the convolution kernel.
- (4) This study presents a fusion dataset for ship target detection by combining the Sea-Ships dataset with a self-made dataset. The dataset comprised 8000 visible ship images accurately labeled in a diverse marine setting.

2. Methods

2.1. Algorithm Overview

YOLOv8 is a state-of-the-art target object detection algorithm that has demonstrated excellent performance in pedestrian detection [19], ship recognition [20], and vehicle detection [21]. The YOLOv8 model consists of four main parts: the input end, trunk, neck, and head, as illustrated in Figure 1. In the backbone network, it retains the CSPDarkNet (Cross Stage Partial DarkNet) structure of YOLOv5, but innovates the C3 module into a c2f module [21]. The c2f module draws on the ELAN (Efficient and Light-weight Attention Network) design of YOLOv7, optimizing the convergence speed and effectiveness while achieving lightweight performance by increasing parallel gradient flow branches. It utilizes a PANnet (path aggregation network) structure in the neck network design, which effectively transmits shallow positioning information to deeper layers, improving the multi-scale positioning performance. In contrast to the coupling head and anchor-based strategy of YOLOv5, YOLOv8 utilizes a decoupling head and anchor-free approach. This involves discarding the objectness branch and focusing solely on the decoupled classification and regression branches. As a result, YOLOv8 achieves a faster convergence speed and improved detection effectiveness.

For classification and regression branching, YOLOv5 uses BCE loss (binary cross entropy loss) [22] and CIOU loss (complete intersection over union loss) [23], respectively, and YOLOv8 uses VFL loss (varifocal loss) and DFL loss (distribution local loss) + CIOU loss. VFL loss is designed based on BCE loss, introducing asymmetric weighting operations for positive and negative samples, as shown in Equations (1) and (2).

$$Loss_{BCE} = -\frac{1}{n} \sum (y_n \log x_n + (1 - y_n) \log(1 - x_n)) \quad (1)$$

where x_n represents the predicted value; y_n represents the labeled value; n represents the sample number.

$$VFL(p, q) = \begin{cases} -q(q \log(p) + (1 - q) \log(1 - p)) & q > 0 \\ -\alpha p^\gamma \log(1 - p) & q = 0 \end{cases} \quad (2)$$

[illegible]

DFL loss was mainly proposed to address the inflexibility of generated bounding box representations [24]. Traditional object detection struggles to accurately define the true bounding boxes of target objects in complex scenes due to boundary ambiguity and uncertainty caused by factors such as subjective labeling, occlusion, and blurriness. Traditional regression methods approximate the predicted value to a discrete and certain value: the label position y , as shown in Figure 2. For complex scenes, the regression distribution range is more realistic compared to approximating discrete values. As shown in Figure 3, based on a similar concept, DFL loss models the box position as a general distribution, enabling the network to quickly focus on the vicinity of the label position y and improve the detection accuracy.

$$\text{DFL}(\mathcal{S}_i, \mathcal{S}_{i+1}) = -((y_{i+1} - y) \log(\mathcal{S}_i) + (y - y_i) \log(\mathcal{S}_{i+1})) \quad (3)$$

$$S_i = \frac{y_{i+1} - y}{y_{i+1} - y_i} \quad (4)$$

$$S_{i+1} = \frac{y - y_i}{y_{i+1} - y_i} \quad (5)$$

where y represents the labeled position of the detected object, while y_i and y_{i+1} represent the left and right positional coordinates of the predicted bounding box ($y_i \leq y \leq y_{i+1}$), respectively.

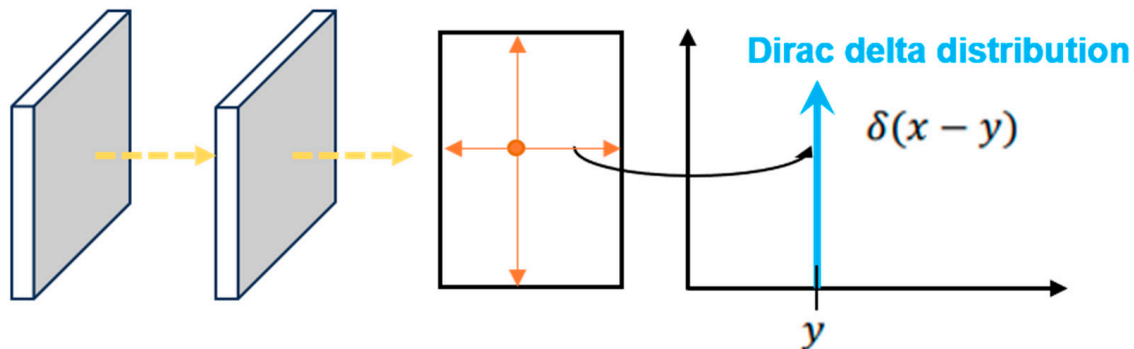


Figure 2. Schematic diagram of the traditional regression method.

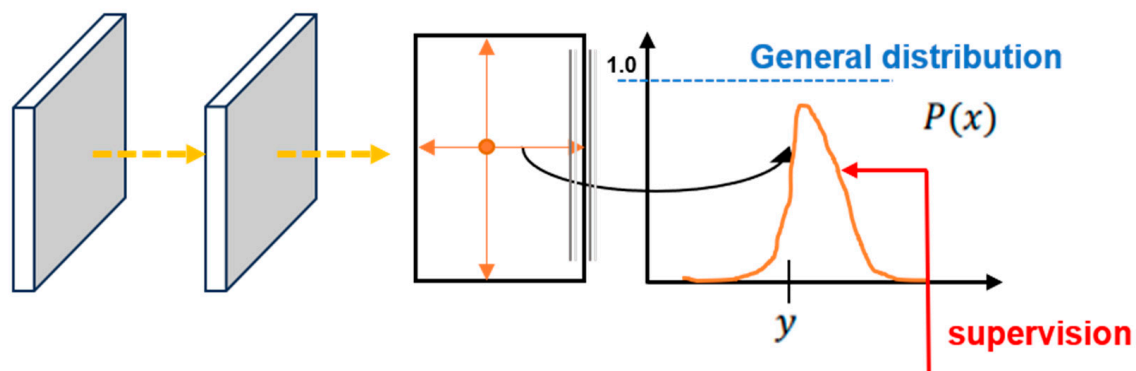


Figure 3. Schematic of the range of the regression distribution.

The CIOU loss function can be calculated using the following formula:

$$Loss_{CIOU} = 1 - IoU + \frac{\rho^2(b^x, b^y)}{c^2} + \alpha v \quad (6)$$

$$\alpha = \frac{v}{(1 - IoU) + v} \quad (7)$$

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w_y}{h_y} - \arctan \frac{w_x}{h_x} \right)^2 \quad (8)$$

The parameter α is utilized for trade-off, while the parameter v is used to assess the aspect ratio consistency. The variables b^x and b^y denote the centers of the prediction box and the marked box, respectively. ρ represents the Euclidean distance between the two center points, and c represents the length of the diagonal of the minimum outer rectangle that encompasses both the prediction box and the marked box.

In this article, we selected the YOLOv8 algorithm as the research foundation for several reasons. First of all, YOLOv8 has shown remarkable performance in target detection due to its sophisticated deep learning architecture and optimization techniques. It is capable of achieving rapid detection speeds without compromising accuracy, making it well-suited for real-time target detection applications. Second, YOLOv8 has outstanding generalization abilities, adapting to object detection tasks in different scenes and backgrounds. Regardless

of the size, shape, or occlusion level of the target, YOLOv8 can achieve effective detection, making it widely applicable in various practical scenarios. Finally, YOLOv8 provides a simple and easy-to-use interface and tools, facilitating developers to integrate it into various application scenarios. Simultaneously, it supports cross-platform deployment including PCs, mobile devices, and embedded systems, further broadening the deployment possibilities of YOLOv8 in practical applications.

2.2. Improved YOLOv8n

In the field of ship target recognition, the YOLOv8 algorithm faces challenges in maintaining detection accuracy and efficiency in complex environments. This study chose YOLOv8n as the reference model and proposed enhancements. The network structure diagram of the CSD-YOLO algorithm is illustrated in Figure 4. To begin with, the CG block module was integrated to enhance the c2f module of the backbone network, thereby improving the model's feature extraction capability to accurately capture the crucial information in images. Second, the introduction of the slim-neck module optimized the neck section, achieving more efficient feature fusion and improving the accuracy and efficiency of ship target detection. Finally, in the neck network, the D-LKA mechanism was introduced. This mechanism combines the broad receptive field of large convolutional kernels with the flexible adaptability of deformable convolution. It not only efficiently processes complex visual information but also significantly enhances the model's adaptability to target shapes and sizes, thereby improving the detection performance for small target objects in complex environments.

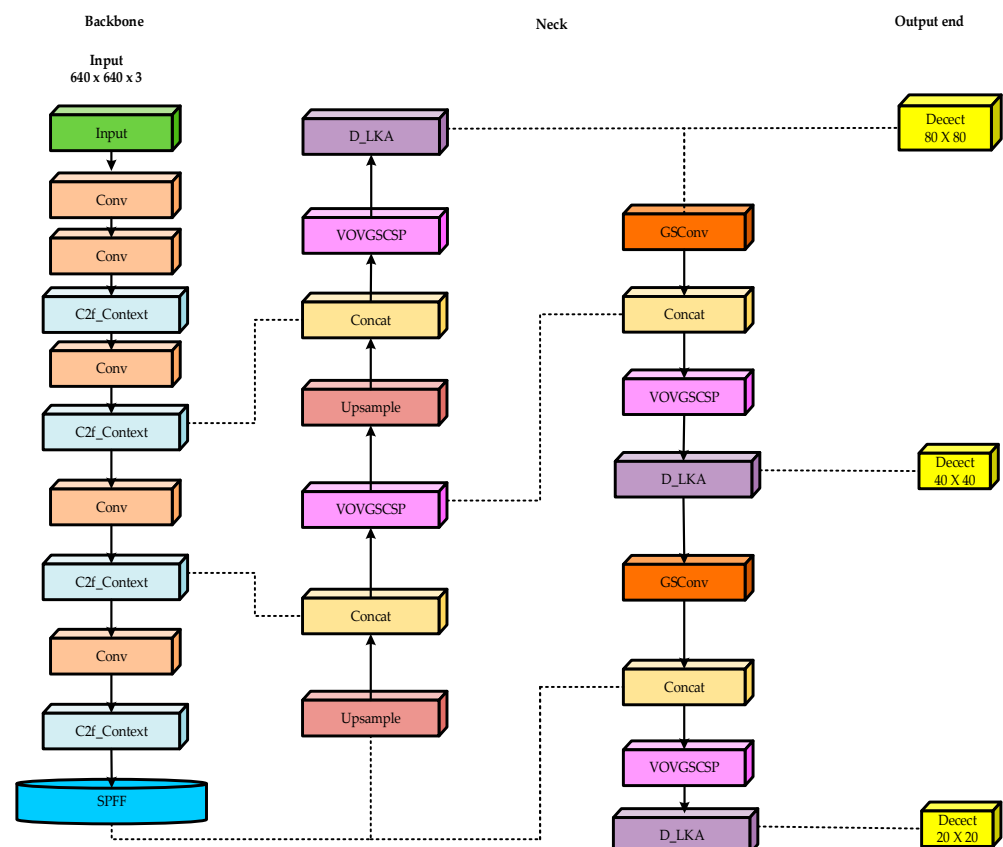


Figure 4. CSD-YOLO algorithm network structure diagram.

2.3. Context Guided Block Module

The CG block is a lightweight module designed to capture local features, surrounding contexts, and global contexts. By effectively fusing this information, it signifi-

cantly improves the accuracy of semantic segmentation. This module consists of several key components:

- (1) Local feature extractor $f_{loc}(\cdot)$: This learns local features from the input data through standard convolutional layers. These features become important information within the module and are subsequently combined with surrounding contextual features to build a comprehensive understanding of various regions of the image, thereby enhancing the network's ability to grasp the image content.
- (2) Surrounding context extractor $f_{sur}(\cdot)$: This utilizes dilated convolution to expand the receptive field and enhance the processing capability of multi-scale information. This feature is particularly important when handling high-resolution images and semantic segmentation tasks in complex scenes.
- (3) Joint feature extractor $f_{joi}(\cdot)$: Through concatenation layers, batch normalization (BN), and parametric ReLU (PReLU) operations, it effectively fuses local features and surrounding contextual features to form joint features. These features not only retain the details of the input data, but also incorporate information from a wider area, enabling the network to comprehensively consider local and contextual information, thereby significantly improving the accuracy of semantic segmentation.
- (4) Global context extractor $f_{glo}(\cdot)$: This integrates global contexts through a global average pooling layer and subsequently utilizes a multilayer perceptron to deeply extract global features. Furthermore, a scaling layer is employed to weight the joint features, strengthening beneficial features and weakening redundant information. This mechanism effectively captures and utilizes global information from the input image, greatly enhancing the learning efficiency of the joint feature extractor.

The components collaborate to enhance the network's capability in understanding multi-scale features within intricate scenes, thereby improving the accuracy of the backbone network in tasks like semantic segmentation and target detection. By integrating local and global contexts, the module achieves pixel-level classification accuracy, which is essential for image analysis. Figure 5 displays the structure diagram of the CG block module.

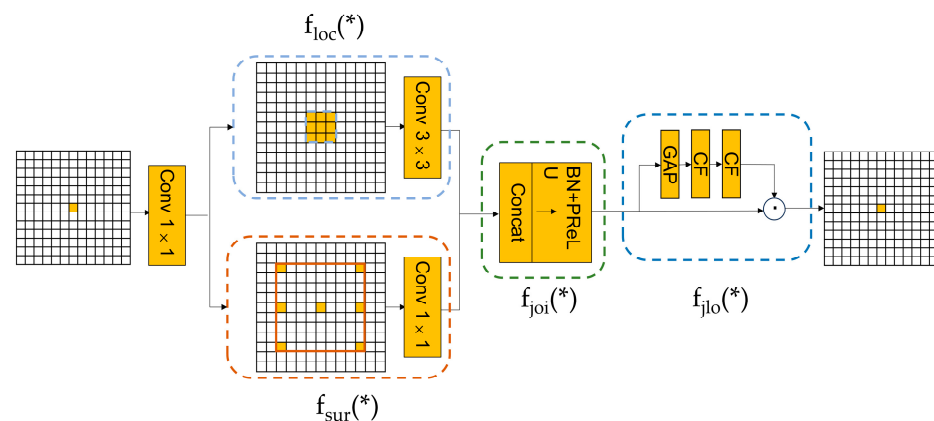


Figure 5. Context guided block module structure.

2.4. Slim-Neck Construction

In the design of the YOLOv8n algorithm, standard convolution and the C2f module are widely used to improve prediction accuracy. However, this increase in accuracy is often accompanied by a decrease in computational speed and an increase in model parameters. This paper introduces the slim-neck structure to integrate feature information from the feature maps of different scales in the backbone network, reducing the model complexity without compromising accuracy. As a result, YOLOv8n achieves a more balanced performance in terms of efficiency and accuracy.

The slim-neck is a unique structure that optimizes the neck area, combining group-wise spatial convolution (GSConv) and modular elements. The GSConv module integrates spa-

tial convolution (SC), depthwise separable convolution (DSC), and shuffle operations [25]. The parameter quantity and computational complexity of the convolutional layer can be expressed using Formulas (9) and (10).

$$Paramconv2d = C_{in} * C_{out} * K_h * K_w \quad (9)$$

$$FLOPconv2d = Paramconv2d * M_{out-h} * M_{out-w} \quad (10)$$

where C_{in} represents the number of input channels, C_{out} represents the number of output channels, K_h and K_w represent the height and width of the convolution kernel, respectively, and M_{out-h} and M_{out-w} represent the height and width of the output feature map, respectively.

Compared with standard convolution, GSConv reduces its computational cost by approximately 30% to 40% while maintaining competitive performance. To enhance the model's performance further, we replaced the standard convolution in the neck layer with GSConv and introduced a vision over vision-group-wise spatial convolution and shuffle pooling (VoV-GSCSP) module based on GSConv. This module can further improve the network's lightweight and efficiency while ensuring accuracy. The calculation process of the SC and DSC is illustrated in Figure 6, and the calculation formulas are presented in Equations (11) and (12). The detailed structure of the GSConv and VoV-GSCSP modules is elaborated in Figures 7 and 8.

$$Time_{SC} \sim O(W \cdot H \cdot K_1 \cdot K_2 \cdot C_1 \cdot C_2) \quad (11)$$

$$Time_{DSC} \sim O(W \cdot H \cdot K_1 \cdot K_2 \cdot 1 \cdot C_2) \quad (12)$$

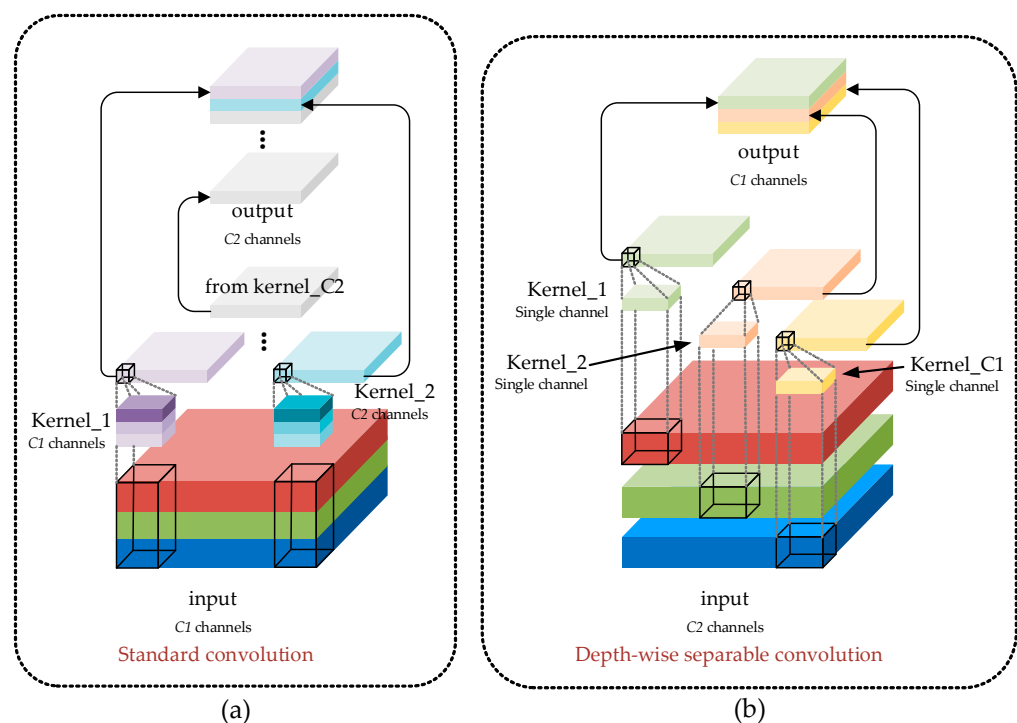


Figure 6. (a) SC calculation process. (b) DSC calculation process.

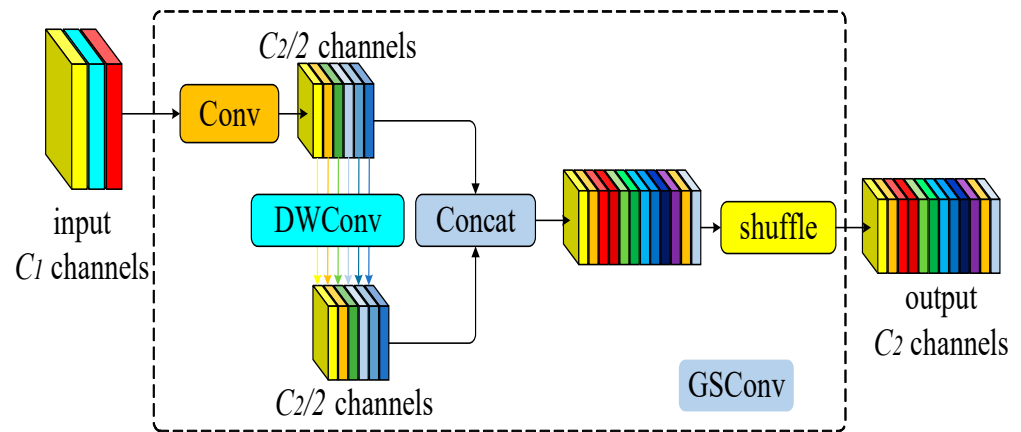


Figure 7. GSConv module structure diagram.

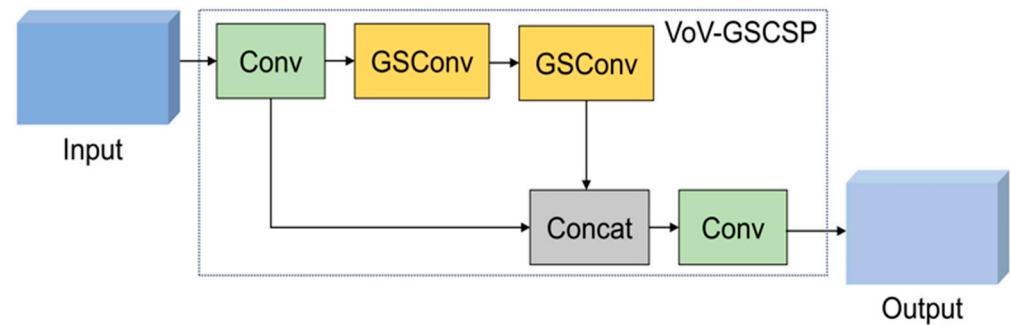


Figure 8. VoV-GSCSP module structure diagram.

Among these parameters, W represents the width of the output feature map, H represents the height of the output feature map, $K_1 \cdot K_2$ denotes the size of the convolution kernel, C_1 indicates the number of channels of each convolution kernel and the input feature map, and C_2 represents the number of channels of the output feature map.

2.5. Deformable Large Kernel Attention Mechanism

The core idea of D-LKA lies in combining large convolution kernels with the attention mechanism of deformable convolution. By introducing large convolution kernels, it simulates the extensive receptive field achieved by the self-attention mechanism, while cleverly avoiding the high computational costs associated with traditional self-attention mechanisms. This design not only enhances the model's ability to capture spatial information, but also ensures computational efficiency, allowing D-LKA to achieve higher efficiency while maintaining performance. The D-LKA module can be represented by Formulas (13) and (14).

$$\text{Attention} = \text{Conv}_{1 \times 1}(\text{Conv}_{\text{DDW-D}}(\text{Conv}_{\text{DDW}}(F')))) \quad (13)$$

$$\text{Output} = \text{Conv}_{1 \times 1}(\text{Attention} \otimes F') + F \quad (14)$$

The input feature is denoted as $F \in \mathbb{R}^{C \times H \times W}$ and $F' = \text{GELU}(\text{Conv}(F))$. The attention component $\mathbb{R}^{C \times H \times W}$ represents the attention map, where each value indicates the relative importance of the corresponding feature.

D-LKA mainly consists of the following three major components. First, large convolution kernels are used to capture extensive contextual information in images, simulating the large receptive field of the self-attention mechanism. Second, deformable convolution technology is combined to enable the model's sampling grid to flexibly adjust based on image features, adapting to diverse data patterns. Finally, the 2D and 3D versions of D-LKA demonstrate excellent performance at different data depths. Additionally, D-LKA flexibly adjusts the sampling grid through deformable convolution, allowing the model to better

adapt to different data patterns. This can effectively solve the difficulty of detecting small and multiple targets in complex backgrounds or under different lighting conditions. The architecture diagram of D-LKA is shown in Figure 9.

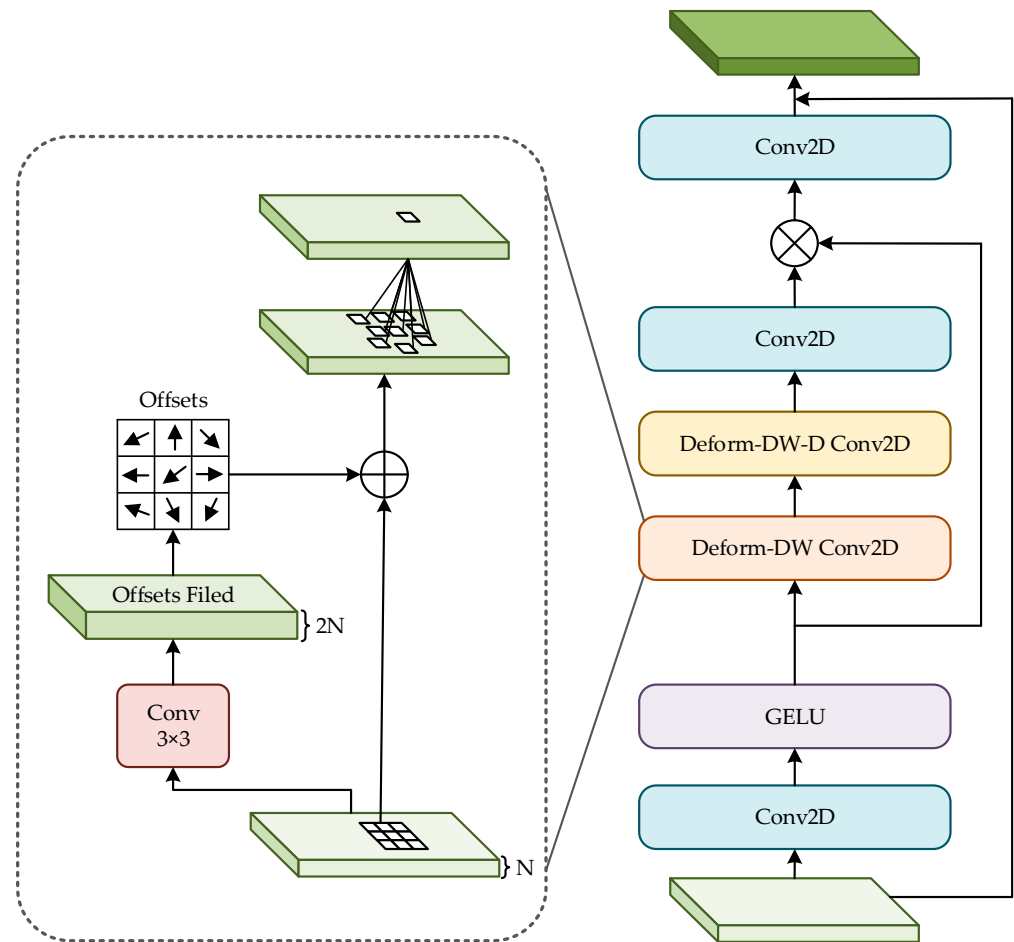


Figure 9. Deformable-LKA architecture diagram.

3. Experiments

3.1. Experimental Environment

The experimental environment in this study was run on Windows 10 Professional Edition, utilizing an Intel Core i7-10750K CPU @ 2.60 GHz. Python was the chosen compilation language, with PyTorch used as the deep learning framework and YOLOv8n as the benchmark model. Table 2 presents the detailed configuration of the basic parameters in the algorithm.

Table 2. Configuration parameters of the experimental platform.

Configure	Releases
Operating system	Windows10
CPU	Intel (R) Core (TM) i7-10750K CPU @ 2.60 GHz
GPU	NVIDIA GeForce RTX 3080
RAM	32 G
Toolboxes	CUDA 11.2
Compilers	Python 3.9.12
Frameworks	PyTorch 2.0.0

3.2. Training Parameter Settings

The CSD-YOLO network was trained using carefully selected training parameters, as detailed in Table 3 following the experimental methodology.

Table 3. Detailed parameter list for model training.

Parameter Name	Parameter Value
Training round	200
Sample batch size	32
Input image size	640×640
Learning rate	0.01
Learning threshold IOU-t	0.2
Weight decay	0.0005
Learning rate momentum	0.937

3.3. Datasets

The experimental dataset in this paper was comprised of two parts: one was a self-made dataset containing 1194 ship images, and the other was the publicly available SeaShips dataset [26], which contains 7000 ship images, as shown in Figure 10. To enhance data diversity and experimental accuracy, this paper combined these two datasets to form a mixed dataset. This mixed dataset covered nine common types of ships and island targets at sea including container ships, ore carriers, passenger ships, bulk carriers, fishing boats, cruise ships, general cargo ships, islands, and other ships.

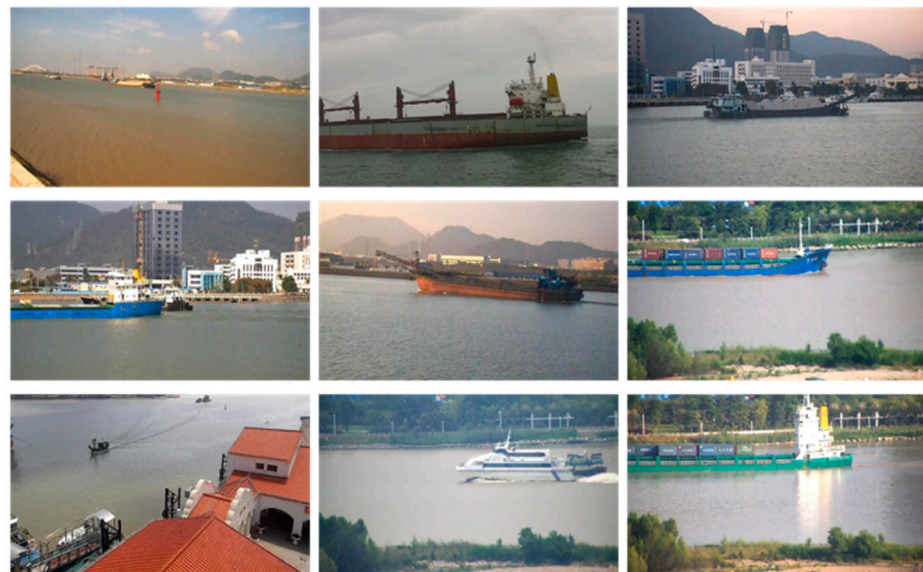


Figure 10. Partial image of the training set.

To ensure the rigor of the experiment, this paper randomly split the 8194 images in the mixed dataset into training, validation, and test sets in a 6:3:1 ratio. This splitting method ensured that each set had sufficient data volume while maintaining the randomness and representativeness of the data.

The mixed dataset used in the experiment included a rich variety of ship types and diverse navigation states covering various weather conditions and shooting environments (Table 4). This enabled a more accurate simulation of the actual situation of the camera's environmental perception and image acquisition during the intelligent navigation of ships. This dataset not only provides a wide range of authentic data samples for this paper, but also lays a solid foundation for subsequent model training and evaluation, which is

beneficial for improving the robustness and accuracy of an intelligent navigation system for ships.

Table 4. Distribution of the number of datasets.

Type of Dataset	Training Set/Sheet	Validation Set/Sheet	Test Set/Sheet
Mixed datasets	4916	2458	820

3.4. Indicators for Model Evaluation

In order to objectively assess the effectiveness of the CSD-YOLO ship recognition network model in this paper, reasonable evaluation indices were developed. Five types of metrics were used to evaluate the performance of the algorithm: precision (P) [27], recall (R) [28], average precision map@.5 [29] and map@.5:.95 [30], and frames per second (FPS) [31]. The specific formula for each metric is as follows:

$$Precision = \frac{TP}{TP + FP} \quad (15)$$

TP represents the number of correctly predicted positive samples, FP represents the number of incorrectly predicted positive samples, and P represents the proportion of correctly predicted positive samples among all of the detected positive samples.

$$Recall = \frac{TP}{TP + FN} \quad (16)$$

FN represents the number of falsely predicted negative samples and R represents the proportion of all correctly predicted positive samples out of all the samples detected.

$$AP = \int_0^1 P(R) dR \quad (17)$$

AP represents a comprehensive evaluation metric for multiple detection results of each category of targets. The value of AP is the area under the precision–recall (P–R) curve, which means that for each recall value, the corresponding precision value is calculated, and finally, the average value is taken as the AP value for that category.

$$mAP = \frac{1}{C} \sum_{i=1}^C AP_i \quad (18)$$

$$mAP@0.5 = \frac{\sum_{t=1}^N AP_t}{N}, IoU = 0.5 \quad (19)$$

$$mAP@0.5 : 0.95 = \frac{\sum_{t=1}^N AP_t}{N}, IoU \in [0.5, 0.95] \quad (20)$$

AP_i represents the average precision of the i th target category detection, and mAP represents the mean average precision for multiple categories, which is the average of AP for all categories. $mAP@0.5$ refers to the AP value when the IoU threshold is 0.5. The AP is calculated for each IoU threshold from 0.5 to 0.95 with an increment of 0.05, and the average is taken to obtain $mAP@.5 : 0.95$.

$$FPS = \frac{FigureNumbers}{TotalTime} \quad (21)$$

FPS stands for frames per second, indicating the number of image frames that can be detected per second. It refers to the number of images that the model can process per second, which is a measure of the model's inference speed.

4. Results and Discussion

4.1. Ablation Experiment

4.1.1. Context Guided Block Module

Table 5 displays a comparison of the recognition outcomes between C2f integrated with the context guided block module and the original C2f model. The findings indicate that the overall performance index of C2f with the context guided block module was 0.002 points higher than the original YOLOv8n model's mAP@0.5.

Table 5. Comparison of detection results for different c2f structures.

Arithmetic	P/%	R/%	map@0.5	mAP@0.5:0.95
YOLOv8n-c2f	0.924	0.888	0.911	0.683
YOLOv8n-c2f-CG block	0.928	0.883	0.913	0.668

To better illustrate the impact of the CG block module on c2f enhancement and optimization, this study showcases the changes in the feature heat map with and without the introduction of the CG block module. In the first row in Figure 11, it is evident that the inclusion of the CG block module resulted in the extracted features being predominantly focused around the ship, whereas the baseline c2f structure still showed significant attention toward the sea surface and coast. The second row demonstrates that the CG block module enhanced the model's ability to detect small target objects, with a more pronounced focus on the ship compared to the baseline c2f structure. Moreover, the integrated CG block module structure effectively suppressed the background features that were not part of the main object, as depicted in the last row of Figure 11. In conclusion, integrating the CG block module into the C2f structure enhanced the model's feature extraction capability.

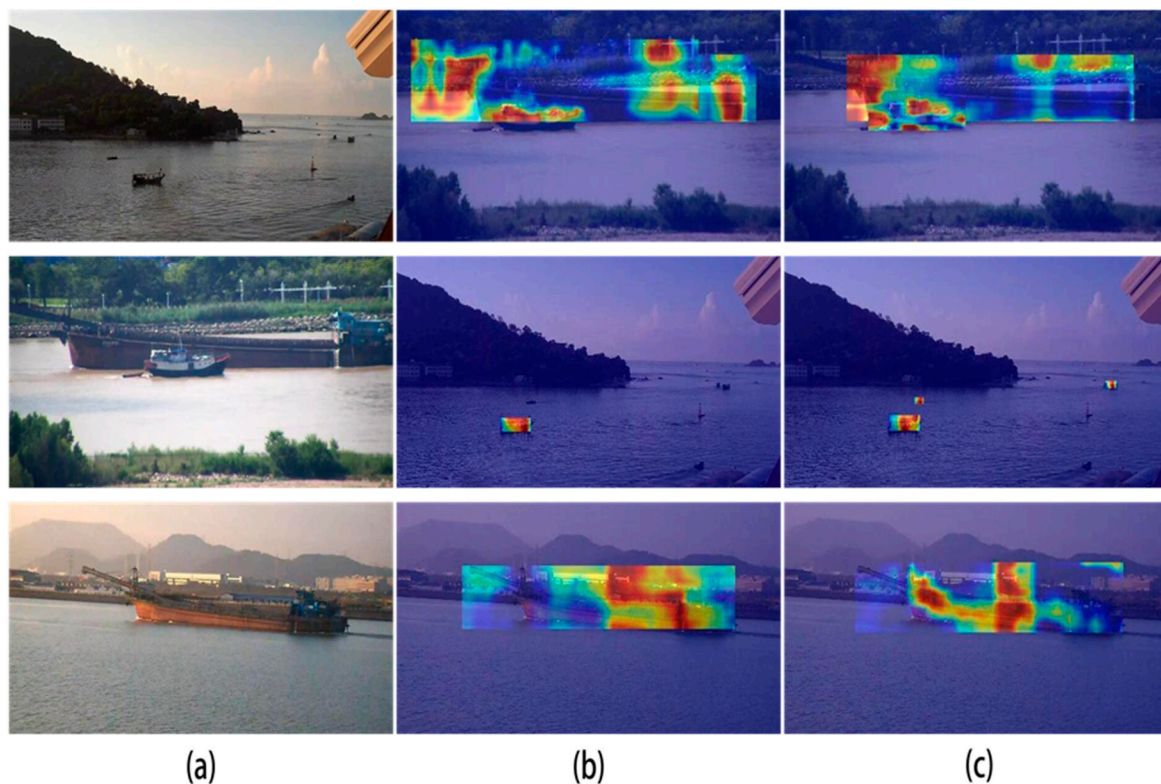


Figure 11. Heatmap of the c2f-CG block structure showing the (a) original image, (b) YOLOv8-c2f, and (c) YOLOv8-c2f-CG block.

4.1.2. Slim-Neck Construction

Table 6 displays the evaluation metrics for whether or not the slim-neck structure is introduced. The results indicate that the introduction of the slim-neck structure effectively improved the overall performance of the model. Specifically, there was an increase of 0.004 in mAP@0.5 and an increase of 0.007 in mAP@0.5:0.95.

Table 6. Comparison of detection results with and without the introduction of the slim-neck structure.

Arithmetic	P/%	R/%	map@0.5	mAP@0.5:0.95
YOLOv8n	0.924	0.888	0.911	0.683
YOLOv8n-Slim-neck	0.919	0.901	0.915	0.69

Figure 12 shows randomly selected image heatmaps. The design of the slim-neck structure takes full advantage of the benefits of depthwise separable convolution (DSC) while minimizing its drawbacks. Through the introduction of methods like GSConv, slim-neck can enhance detection performance while maintaining model accuracy, particularly excelling in detecting small targets and achieving a higher average precision. The results indicate that compared to the original structure, the slim-neck structure pays more attention to the tested objects. Additionally, by comparing the brightness of small targets, it can be observed that the introduction of the slim-neck structure enabled the better detection of small targets.



Figure 12. Whether or not to introduce the slim-neck structure heat map display.

4.1.3. Deformable-LKA Attention Mechanism

Table 7 displays the impact of adopting the deformable-LKA attention mechanism on detection accuracy. The results indicate that compared to the original benchmark model without the D-LKA attention mechanism, the D-LKA attention mechanism significantly improved the detection accuracy of the existing model.

Table 7. Impact of whether or not to use the deformable-LKA attention mechanism on the detection results.

Arithmetic	P/%	R/%	map@0.5	mAP@0.5:0.95
YOLOv8n	0.924	0.888	0.911	0.683
YOLOv8n-D-LKA	0.913	0.905	0.915	0.692

As shown in the heatmaps in Figure 13, the model adopting the deformable-LKA attention mechanism was highly focused on the detected objects. However, the original model tended to focus more on features in the background and peripheral areas. The results indicate that the adopted deformable-LKA attention mechanism can effectively improve the accuracy of detecting different ship targets and can adapt well to changes in complex environments.

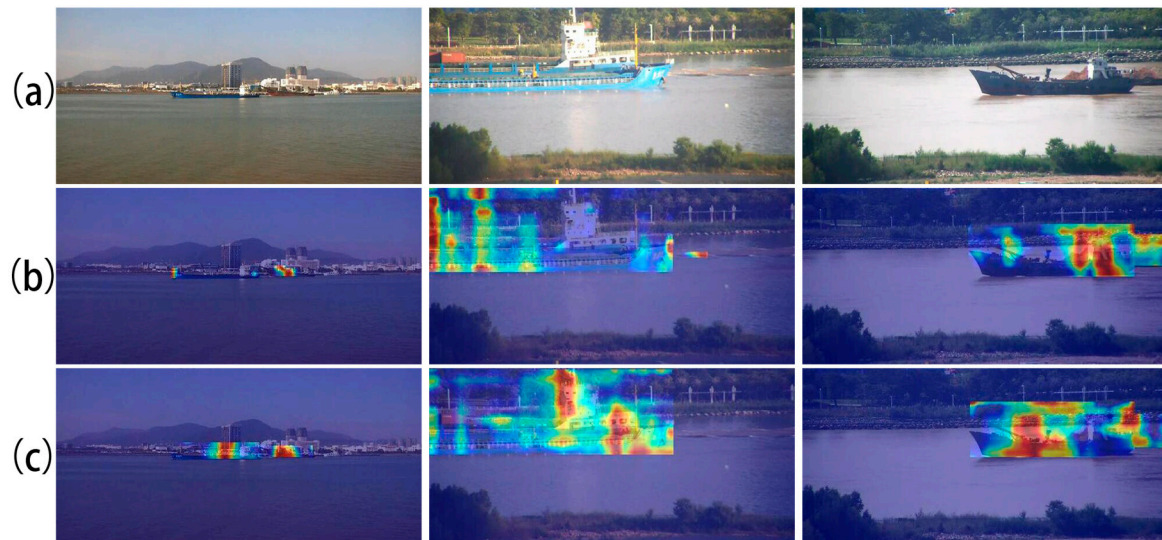


Figure 13. Heatmap with and without the D-LKA attention mechanism showing the (a) original image, (b) YOLOv8n, and (c) YOLOv8n-D-LKA.

4.2. Analysis of Test Results

To more intuitively reflect the performance differences between the improved algorithm and the baseline algorithm, this paper compared the main ship target detection effects and the ship detection results. As shown in Tables 8 and 9, the CSD-YOLO algorithm outperformed the original algorithm in terms of detection accuracy for most major types of ship targets, indicating that the improved algorithm has enhanced performance and accuracy.

Table 8. YOLOv8n algorithm detection of ship type performance results.

Vessel Type	P/%	R/%	mAP@0.5	mAP@0.5:0.95
Container ship	0.98	1	0.994	0.851
Fishing vessel	0.965	0.922	0.963	0.735
Bulk carrier	0.97	0.959	0.981	0.777
Ocean liner	0.98	0.923	0.982	0.726
Ore carrier	0.953	0.976	0.983	0.778
General cargo ship	0.966	0.97	0.978	0.784
Passenger ship	0.962	0.979	0.975	0.794

Table 9. CSD-YOLO algorithm detection of ship type performance results.

Vessel Type	P/%	R/%	mAP@0.5	mAP@0.5:0.95
Container ship	0.942	1	0.995	0.833
Fishing vessel	0.967	0.943	0.97	0.736
Bulk carrier	0.938	0.956	0.977	0.761
Ocean liner	0.969	0.949	0.982	0.721
Ore carrier	0.944	0.976	0.977	0.761
General cargo ship	0.959	0.979	0.986	0.806
Passenger ship	0.951	1	0.985	0.774

As shown in Figure 14, this paper selected detection results from four different environments for comparison, from top to bottom: foggy environment, nighttime environment, multi-target environment, and small target environment. From the comparison chart, it can be seen that the CSD-YOLO algorithm outperformed the baseline algorithm in detection accuracy and efficiency in various environments.

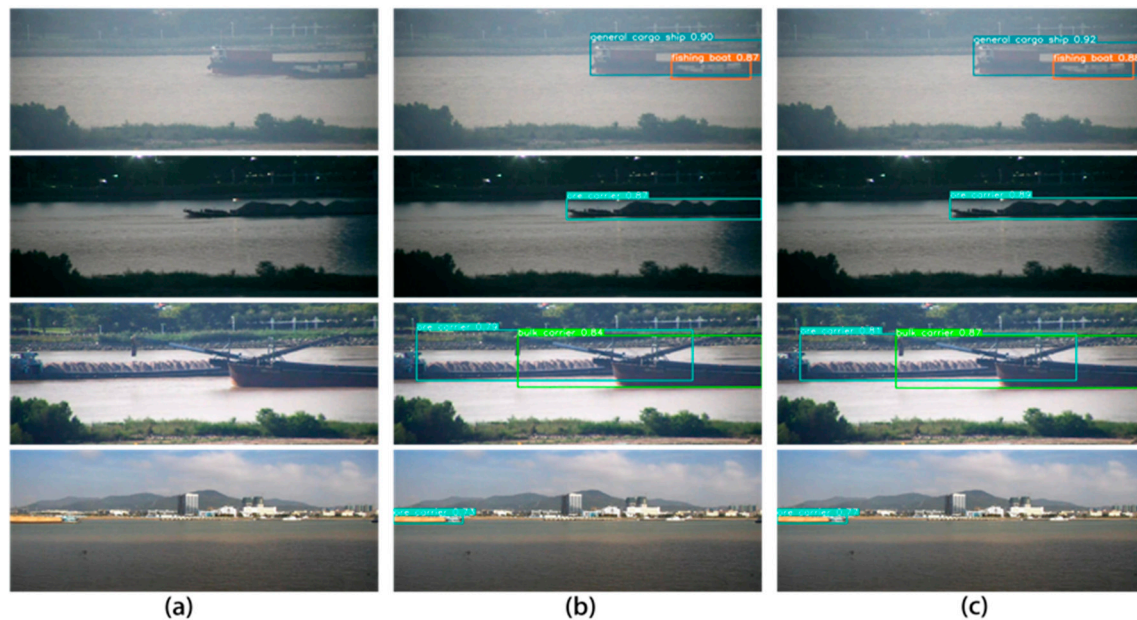


Figure 14. Comparison chart of the detection results. (a) Original image, (b) YOLOv8n, and the (c) CSD-YOLO algorithm.

4.3. Discussion

In order to further validate the efficacy of the proposed algorithm in this study, the CSD-YOLO algorithm was experimentally compared and analyzed with established algorithms like Faster R-CNN, SSD, YOLOv3, YOLOv4, YOLOv5, YOLOv7, YOLOv8n, etc. All of the experimental parameters remained consistent across the comparison, with 60% of fusion datasets comprising 4916 images used in the experiments and 200 training rounds conducted. The results of these comparison experiments are detailed in Table 10.

Table 10. Experimental results comparing different algorithms.

Arithmetic	P/%	R/%	mAP@0.5	FPS
Yolov3	88.5	90.2	90.8	57
Yolov4	87.7	85.6	86.8	53
Yolov5	90.3	89.7	89.6	47
Yolov7	92.5	90.1	90.8	54
Yolov8	92.4	88.8	91.1	102
SSD	89.9	72.7	74.9	41
Faster R-CNN	80.5	84.8	82.1	11
CSD-YOLO	91.5	89.5	91.5	23

Based on the experimental results presented in Table 10, the algorithms proposed in this study exhibited superior performance compared to the widely used algorithms in ship target detection. Notably, the Faster R-CNN and SSD algorithms are limited in their adaptability to multi-scale ship targets due to fixed anchor frame parameters, resulting in relatively low detection accuracy [32,33]. Although the YOLOv5 and YOLOv7 algorithms demonstrated faster detection speeds (47 and 54 FPS, respectively), their detection accuracies remained relatively low at 89.6% and 90.8%, respectively. In contrast, the CSD-YOLO

algorithm achieved the highest detection accuracy while retaining real-time performance. The CSD-YOLO algorithm distinguishes itself from similar algorithms in several key ways. First, it introduces the slim-neck module to innovatively design the neck structure, significantly enhancing the feature fusion efficiency and improving the detection accuracy and speed. Second, it optimizes the c2f module of the backbone network by integrating the context guided block module, thereby markedly enhancing the model's feature extraction capability and enabling a more accurate capture of essential information from the image. Finally, the introduction of the D-LKA mechanism dynamically adjusts the shape and size of the convolution kernel to better adapt to image features, consequently improving the model's adaptability to target shape and size. By elucidating these advancements, the study underscores the significant contributions of the CSD-YOLO algorithm to the field of ship target detection, marking a notable stride in the advancement of detection accuracy and real-time performance.

The CSD-YOLO algorithm shares numerous similarities and differences with existing algorithms.

- (1) Both the CSD-YOLO algorithm and the existing algorithms use a deep learning-based framework.
- (2) The CSD-YOLO algorithm uses a similar dataset to all existing algorithms: the Sea-Ships dataset.
- (3) The CSD-YOLO algorithm achieved a similar performance to the existing algorithms in terms of ship recall (e.g., YOLOv5 has a recall of 89.5, and the CSD-YOLO algorithm has a recall of 89.5).
- (4) Model architecture: Compared with the benchmark YOLOv8n algorithm architecture, the CSD-YOLO algorithm introduces the deformable large kernel attention mechanism, which significantly improved the ship detection performance, as can be seen from the data in Table 7.
- (5) Multi-scale fusion of features: Unlike traditional methods that use only single-scale features, the CSD-YOLO algorithm applies the slim-neck module to the neck part to achieve more efficient scale fusion to enhance the model's ability to characterize ships.
- (6) Speed and efficiency: As can be seen from the data in Table 10, compared to the benchmark algorithm, the CSD-YOLO algorithm significantly improved the accuracy of ship detection while maintaining the existing detection speed.

Overfitting is a problem to be aware of in deep learning and may lead to poor generalization of the model. In the preparation stage of the above experiments, many regularization methods were adopted in this paper. From the experimental results, it can be seen that the training and testing errors of the CSD-YOLO algorithm decreased synchronously with the increase in the number of training rounds, so the model did not suffer from the overfitting problem during the training process.

5. Conclusions

In response to the challenges faced in ship detection under complex marine environments, this study proposed a novel ship detection algorithm, termed CSD-YOLO, which addresses the limitations of maintaining detection accuracy and efficiency. The benchmark model utilized in this study was YOLOv8n. By innovatively redesigning the neck structure and incorporating the CG block module, the D-LKA mechanism was introduced and seamlessly integrated into the YOLOv8n algorithm. Experimental validation demonstrated that the CSD-YOLO algorithm significantly enhanced the performance of ship detection in complex navigational environments while maintaining real-time detection speeds. Specifically, the precision of the algorithm reached 91.5%, the recall achieved 89.5%, and the mAP attained 91.5%. Compared to the baseline algorithm, CSD-YOLO showed improvements in the recall rate (by 0.7%) and average precision (by 0.4%). These results provide robust technical support for the real-time tracking and identification of ships in challenging marine conditions.

In the future, we anticipate the enhancement of datasets representing various environmental scenarios at sea, thereby enriching the research presented in this paper. This will lay the groundwork for the development of a comprehensive target detection embedded system. The ultimate goal is to provide more reliable technical support for achieving intelligence, safety, and visualization in the maritime field. This advancement will contribute significantly to the realization of advanced capabilities in maritime applications.

Author Contributions: Conceptualization, T.W., H.Z. and D.J.; Methodology, T.W.; Software, T.W. and H.Z.; Validation, T.W. and H.Z.; Formal analysis, T.W.; Investigation, H.Z.; Resources, D.J.; Data curation, T.W.; Writing—original draft preparation, T.W. and H.Z.; Writing—review and editing, T.W. and D.J.; Visualization, H.Z.; Supervision, T.W. and D.J.; Project administration, T.W. and D.J. All authors have read and agreed to the published version of the manuscript.

Funding: The research of this thesis was supported by the Ministry of Education, Industry-University Cooperation Collaborative Education Project (NO. 202102104008); Xiamen Municipal Natural Science Foundation Upper-level Project (NO. 3502ZZ202373038); Chongqing Municipal Postgraduate Student Supervisory Team Construction Project (NO. JDDSTD2022009).

Data Availability Statement: Data included in the article.

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

References

- Jiang, M.; Gu, L.; Li, X.; Gao, F.; Jiang, T. Ship Contour Extraction from SAR Images Based on Faster R-CNN and Chan–Vese Model. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–14. [\[CrossRef\]](#)
- Cai, J.; Du, S.; Lu, C.; Xiao, B.; Wu, M. Obstacle Detection of Unmanned Surface Vessel based on Faster RCNN. In Proceedings of the 2023 IEEE 6th International Conference on Industrial Cyber-Physical Systems (ICPS), Wuhan, China, 8–11 May 2023; pp. 1–6. [\[CrossRef\]](#)
- Yang, J.-R.; Hao, L.-Y.; Liu, Y.; Zhang, Y. SLT-Net: Enhanced Mask RCNN network for ship long-tailed detection. In Proceedings of the 2023 IEEE 2nd Industrial Electronics Society Annual On-Line Conference (ONCON), Online, 8–10 December 2023; pp. 1–6. [\[CrossRef\]](#)
- Qi, L.; Li, B.; Chen, L.; Wang, W.; Dong, L.; Jia, X.; Huang, J.; Ge, C.; Xue, G.; Wang, D. Ship Target Detection Algorithm Based on Improved Faster R-CNN. *Electronics* **2019**, *8*, 959. [\[CrossRef\]](#)
- Liu, Y.; Wang, Z.; Zhang, F.; Xie, J.; Xu, Z. Target scale information detection based on improved Faster R-CNN. In Proceedings of the Third International Seminar on Artificial Intelligence, Networking, and Information Technology (AINIT 2022), Shanghai, China, 23–25 September 2022. Proc. SPIE 12587, 125870V. [\[CrossRef\]](#)
- Chai, B.; Nie, X.; Zhou, Q.; Zhou, X. Enhanced Cascade R-CNN for Multi-scale Object Detection in Dense Scenes from SAR Images. *IEEE Sens. J.* **2024**, *early access*. [\[CrossRef\]](#)
- Lin, Z.; Ji, K.; Leng, X.; Kuang, G. Squeeze and excitation rank faster R-CNN for ship detection in SAR images. *IEEE Geosci. Remote Sens. Lett.* **2018**, *16*, 751–755. [\[CrossRef\]](#)
- Guo, H.; Yang, X.; Wang, N.; Song, B.; Gao, X. A rotational libra R-CNN method for ship detection. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 5772–5781. [\[CrossRef\]](#)
- Park, J.; Moon, H. Lightweight Mask RCNN for Warship Detection and Segmentation. *IEEE Access* **2022**, *10*, 24936–24944. [\[CrossRef\]](#)
- Wen, G.; Cao, P.; Wang, H.; Chen, H.; Liu, X.; Xu, J.; Zaiane, O. MS-SSD: Multi-scale single shot detector for ship detection in remote sensing images. *Appl. Intell.* **2023**, *53*, 1586–1604. [\[CrossRef\]](#)
- Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
- Ye, Y.; Zhen, R.; Shao, Z.; Pan, J.; Lin, Y. A novel intelligent ship detection method mased on attention mechanism feature enhancement. *J. Mar. Sci. Eng.* **2023**, *11*, 625. [\[CrossRef\]](#)
- Wang, L.; Zhang, Y.; Lin, Y.; Yan, S.; Xu, Y.; Sun, B. Ship Detection Algorithm Based on YOLOv5 Network Improved with Lightweight Convolution and Attention Mechanism. *Algorithms* **2023**, *16*, 534. [\[CrossRef\]](#)
- Ting, L.; Baijun, Z.; Yongsheng, Z.; Shun, Y. Ship Detection Algorithm based on Improved YOLO V5. In Proceedings of the 2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE), Dalian, China, 15–17 July 2021; pp. 483–487. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Xing, B.; Wang, W.; Qian, J.; Pan, C.; Le, Q. A Lightweight Model for Real-Time Monitoring of Ships. *Electronics* **2023**, *12*, 3804. [\[CrossRef\]](#)

17. Zhu, C.; Liu, W.; Liu, Y. Ship target detection method based on improved YOLOv8. In Proceedings of the International Conference on Computer Graphics, Artificial Intelligence, and Data Processing (ICCAID 2023), Qingdao, China, 1–3 December 2023. Proc. SPIE 13105, 131051E. [\[CrossRef\]](#)
18. Cheng, S.; Zhu, Y.; Wu, S. Deep learning based efficient ship detection from drone-captured images for maritime surveillance. *Ocean. Eng.* **2023**, *285*, 115440. [\[CrossRef\]](#)
19. Liu, Q.; Ye, H.; Wang, S.; Xu, Z. YOLOv8-CB: Dense Pedestrian Detection Algorithm Based on In-Vehicle Camera. *Electronics* **2024**, *13*, 236. [\[CrossRef\]](#)
20. Chen, W.-Z.; Hao, L.-Y.; Liu, H.; Zhang, Y. Real-Time Ship Detection Algorithm Based on Improved YOLOv8 Network. In Proceedings of the 2023 IEEE 2nd Industrial Electronics Society Annual On-Line Conference (ONCON), Online, 8–10 December 2023; pp. 1–6. [\[CrossRef\]](#)
21. Wang, H.; Liu, C.; Cai, Y.; Chen, L.; Li, Y. YOLOv8-QSD: An Improved Small Object Detection Algorithm for Autonomous Vehicles Based on YOLOv8. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 1–16. [\[CrossRef\]](#)
22. Wu, T.; Gao, G.; Huang, J.; Wei, X.; Wei, X.; Liu, C.H. Adaptive Spatial-BCE Loss for Weakly Supervised Semantic Segmentation. In *Computer Vision—ECCV 2022*; Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T., Eds.; ECCV 2022; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2022; Volume 13689. [\[CrossRef\]](#)
23. Du, S.; Zhang, B.; Zhang, P.; Xiang, P. An Improved Bounding Box Regression Loss Function Based on CIOU Loss for Multiscale Object Detection. In Proceedings of the 2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML), Chengdu, China, 16–18 July 2021; pp. 92–98. [\[CrossRef\]](#)
24. Zhao, S.; Tao, R.; Jia, F. DML-YOLOv8-SAR Image Object Detection Algorithm (Version 1). *Res. Sq.* **2024**, preprint. [\[CrossRef\]](#)
25. Jiang, T.; Chen, S. A Lightweight Forest Pest Image Recognition Model Based on Improved YOLOv8. *Appl. Sci.* **2024**, *14*, 1941. [\[CrossRef\]](#)
26. Su, L.; Chen, Y.; Song, H.; Li, W. A survey of maritime vision datasets. *Multimed Tools Appl.* **2023**, *82*, 28873–28893. [\[CrossRef\]](#)
27. Tang, X.; Zhang, J.; Xia, Y.; Xiao, H. DBW-YOLO: A High-Precision SAR Ship Detection Method for Complex Environments. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 7029–7039. [\[CrossRef\]](#)
28. Zhu, H.Y.; Huang, H.C.; Rong, Y.; Wang, C. DB-YOLO: A Duplicate Bilateral YOLO Network for Multi-Scale Ship Detection in SAR Images. *Sensors* **2021**, *21*, 8146. [\[CrossRef\]](#)
29. Liu, W.; Chen, Y. IL-YOLOv5: A Ship Detection Method Based on Incremental Learning. In *Advanced Intelligent Computing Technology and Applications*; Huang, D.S., Premaratne, P., Jin, B., Qu, B., Jo, K.H., Hussain, A., Eds.; ICIC 2023; Lecture Notes in Computer Science; Springer: Singapore, 2023; Volume 14087. [\[CrossRef\]](#)
30. Wang, N.; Wang, Y.; Feng, Y.; Wei, Y. MDD-ShipNet: Math-Data Integrated Defogging for Fog-Occlusion Ship Detection. *IEEE Trans. Intell. Transp. Syst.* **2024**, early access. [\[CrossRef\]](#)
31. Ju, M.; Luo, J.; Zhang, P.; He, M.; Luo, H. A Simple and Efficient Network for Small Target Detection. *IEEE Access* **2019**, *7*, 85771–85781. [\[CrossRef\]](#)
32. Li, X.; Fu, C.; Li, X.; Wang, Z. Improved Faster R-CNN for Multi-Scale Object Detection. *J. Comput.-Aided Des. Comput. Graph.* **2019**, *31*, 1095–1101. [\[CrossRef\]](#)
33. Sun, J.; Xu, Z.; Liang, S. NSD-SSD: A Novel Real-Time Ship Detector Based on Convolutional Neural Network in Surveillance Video. *Comput. Intell. Neurosci.* **2021**, *2021*, 7018035. [\[CrossRef\]](#) [\[PubMed\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.