IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 July 2024, accepted 12 August 2024, date of publication 16 August 2024, date of current version 29 August 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3445154*

RESEARCH ARTICLE

# Real-Time Long-Distance Ship Detection Architecture Based on YOLOv8

**YANFENG GONG[1], ZIHAO CHEN[1], WEN DENG[1], JIAWAN TAN[1], AND YABIN LI[2]**
[1]School of Shipping and Naval Architecture, Chongqing Jiaotong University, Chongqing 400074, China
[2]Qingdao Shipping Development Research Institute, Qingdao, Shandong 266200, China

Corresponding author: Jiawan Tan (990020812051@cqjtu.edu.cn)

This work was supported by the Graduate Research and Innovation Project of Chongqing Jiaotong University under Grant 2023S0075.

**ABSTRACT** Long-distance detection of maritime ships is pivotal for the development of intelligent collision avoidance systems. Despite significant advancements in target detection achieved through deep learning, the identification of long-distance ships poses a substantial challenge due to their small pixel size in images. Consequently, the recognition of long-distance ships essentially amounts to small object detection. In response to these challenges in small object detection, this paper proposes Ship-YOLOv8, a modified architecture derived from You Only Look Once version 8 (YOLOv8). First, we developed the C-Bottleneck Transformer neural network (C-BoTNet), which is integrated at the end of the backbone, to enhance the global receptive field and facilitate feature fusion. Additionally, we incorporated shallow features with deep features and introduced a dedicated detection layer for small objects into the original structure. Furthermore, we optimized the C2f in the neck using the cross stage partial network (VoVGSCSP) based on GSConv. Finally, we conducted optimization using the Wise-IoU loss function. Extensive experiments conducted on a self-created dataset of long-distance ships demonstrate the remarkable capabilities of Ship-YOLOv8. The proposed method achieves an $AP_{0.5}$ of 91.8%, significantly outperforming YOLOv8's $AP_{0.5}$ of 70.6%. Moreover, our method attains a detection speed of 4.8 ms per image during inference, showcasing its efficiency in real-time applications. To validate the algorithm's broad applicability, comparative experiments were conducted on a public maritime dataset SeaShips. Ship-YOLOv8 achieved an $AP_{0.5}$ score of 99.3%, surpassing YOLOv8's 98.6%. Code is available at https://github.com/zihaohao123/Ship-YOLOv8.

**INDEX TERMS** Long-distance ship, small object detection, deep learning, YOLOv8.

## I. INTRODUCTION

Human activities in the maritime domain have become increasingly diverse, including maritime traffic, trade, fisheries, and military operations [1]. Given that ships serve as primary facilitators of these activities, their regulation is imperative. With advancements in autonomous vehicle technology, autonomous ships, especially those cargo transportation and high-risk military operations, have garnered significant attention [2]. Ship detection holds significant implications for applications, including automated fisheries oversight, port emergency response operations, and the optimization of maritime traffic [3], [4], [5]. Accurate ship detection directly impacts the safety and real-time operations of both military and civilian applications.

The utilization of electro-optical images for ship detection is a current research focus. Visual images are intuitive and easily understandable for human vision [6]. Moreover, their rich spectral content caters to various imaging needs across diverse scenarios. Visible-light spectral imaging offers high resolution, enabling the acquisition of abundant color and texture information [7]. Ship detection methods using image processing can be divided into two categories: traditional and deep-learning methods.

Traditional ship detection in visual images typically involves two primary steps: candidate area extraction and classification [8], [9]. Initially, the candidate area for the ship target is determined based on factors such as scale, shape, and

The associate editor coordinating the review of this manuscript and approving it for publication was Zhan-Li Sun.

other inherent characteristics, or by using a visual attention mechanism [10]. Features corresponding to the candidate area are then extracted for training. However, conventional methods often lack high-dimensional semantic information, leading to low accuracy [2].

In recent years, deep learning has showcased its versatility and marked progress in the realm of ship object detection [11], [12], [13]. Presently, prevailing methodologies are predominantly classified into two categories: CNN-based and Transformer-based architectures. Within the CNN-based paradigm, methodologies are further subcategorized into anchor-based and anchor-free approaches [14]. The anchor-based algorithms are subdivided into two distinct types: single-stage and two-stage methods. One-stage algorithms employ a direct regression approach to concurrently obtain bounding box coordinates and class probabilities [15]. Notable examples of such neural network architectures include the You Only Look Once (YOLO) series [16], [17], [18], [19], SSD [20], and RetinaNet [21]. Generally, the foremost advantage of single-stage detection algorithms is their swift detection speed. However, this advantage is offset by their relatively lower accuracy. While single-stage algorithms prioritize speed, two-stage algorithms excel in terms of accuracy, albeit at the expense of real-time performance. These two-stage algorithms begin by generating region proposal boundaries, which are then followed by the classification of these boundaries. Employing a methodical approach, they interpretively select sliding target windows, creating multiple windows that potentially contain detected objects. These windows are then identified, and any redundant ones are eliminated. Noteworthy examples of two-stage algorithms include RCNN [22], Fast RCNN [23], Faster RCNN [24], Mask RCNN [25], and Cascade R-CNN [26]. In the realm of anchor-free structures, notable algorithms include CenterNet [27] and Soft Anchor-Point Object Detection (SAPD) [28]. The primary advantage of anchor-free architectures lies in their ability to reduce the network's parameter count, streamlining the model. On the Transformer side, network architecture can be divided into two categories. One type directly employs Transformer as a detector, with DETR [29] being a principal example. DETR operates as an end-to-end, direct prediction model, distinguishing itself in the field. The other category utilizes transformer as a backbone for detection, exemplified by the Vision Transformer (ViT) model. This approach leverages the Transformer's capacity to process global dependencies, thereby enhancing the model's overall detection capabilities. The ViT has been extensively adopted in computer vision, where the attention mechanism establishes global relationships between image pixels, employing serialization and positional embeddings to describe spatial relationships and preserve spatial information in images [30]. Because of these advantages, ViT technology has found widespread application in semantic segmentation, object detection [29], and various cross-modal tasks [32], [33], [34]. However, all transformer-based models, including ViT, share a common drawback: they are computationally intensive, requiring significant hardware resources and often suffering from reduced real-time performance [35].

In the field of intelligent navigation, the primary goal of long-distance detection is to improve the performance of detecting small targets. To achieve this, several advanced functional modules have been integrated into the detection systems. Among these, the path aggregation network (PANet) [36] and the channels and spatial attention module (CSAM) [37] are instrumental in enhancing the accuracy and efficiency of target recognition, which is critical for navigating through complex maritime environments. Notably, PANet remains crucial for addressing multi-scale small-target detection [15]. The transformer model, initially prevalent in natural language processing, gained popularity for handling long sequence data [38]. Zhu et al. [33] have introduced Deformable DETR, which represents an enhanced network architecture based on the original DETR framework. This modification effectively augments its capability in detecting small targets. Dosovitskiy et al. [30] proposed a multilevel detection network based on the ViT specifically designed for the detection of small ships.

Despite the various methods currently available to enhance the accuracy of detecting long-distance ships, accurate detection remains challenging. Firstly, during convolution, pooling, and other operations within the backbone network, the image experiences a reduction in pixel count, leading to decreased resolution and a blurring of target details. During the downsampling process, multiple pixel values are merged into one, causing the features of the target to be confounded with those of other objects. This often results in an uneven integration of global and local information, thereby compromising the clarity and distinctiveness of the target features. Secondly, the deep feature maps generated by the backbone lack the detailed information of small targets. Furthermore, the complexity of the network structure affects the real-time of the detection neural network architecture. Lastly, the imbalance in the quality of data samples affects the model's learning of small targets. Therefore, this paper will focus on optimizing the model based on the aforementioned four aspects.

This study encompasses the following primary contributions:

- To augment the receptive field of the backbone network and enhance the extraction of detailed features by deep feature maps, we propose the integration of the C-Bottleneck Transformer Neural network (C-BoTNet) based on YOLOv8. This integration effectively combines local and global information.
- To address the issue of missing detailed information in the higher-layer feature maps, a large feature map of $160 \times 160$ is integrated into the neck, thereby enhancing the network's performance in detecting small targets.
- In order to improve the real-time of the model, the original C2f of YOLOv8 is optimized through the cross stage

partial network (VoVGSCSP). This module not only reduces the model's parameter size but also enhances the accuracy of detecting targets.
- In an effort to mitigate the influence of samples of varying qualities in the dataset, we have adopted Wise-IoU (W-IoU) to optimize the Complete IoU (C-IoU) loss function. This optimization accelerates algorithm convergence and effectively enhances the overall performance of the detector.
- Images were collected by team members on the sea surface, and a self-made dataset of long-distance ship images was created, encompassing various weather conditions.

The following sections in this study are structured as outlined below: Section II outlines the foundational work in this study, which encompasses an introduction to YOLOv8 as well as a detailed description. In Section III, we delve into the structure and principles of Ship-YOLOv8, encompassing the C-BoTNet module, the VoVGSCSP module, the added micro-target detection layer, and the principles of the W-IoU loss function. Section IV details the conducted experiments to substantiate the efficacy of the architecture proposed in this paper. Finally, Section V summarizes our study, providing an overview of our research content and contributions.

## II. RELATED WORK

### A. SHIP DETECTION

Before deep learning became popular, ship detection mainly focused on traditional image processing methods. Krüger and Orlov [39] developed a ship target detection method, which utilizes various filters and edge-detection algorithms to improve-accuracy in detecting ships within blurry images, Liang and Liang [40] proposed a neural network architecture based on probability distribution, building upon the Canny edge detector. Yet, this method exhibits reduced accuracy in images with significant occlusions. Li et al. [41] introduced a superpixel-level based detector that effectively identifies densely populated ships near coastlines. However, it falls short in multi-scale detection. Liu et al. [42] introduced a method combining the Laplacian of Gaussian with Kalman filtering, addressing some challenges of detecting ships under low light and substantial occlusion. However, it still struggles with accurately identifying smaller ships. Addressing the impact of environmental factors like clouds and sea waves on ship detection in optical images, Wang et al. [43] integrated a maximum symmetric surround model and non-subsampled contourlet transform for enhancing low-frequency signals. Despite these advancements, traditional ship detection methods, requiring manual feature extraction, are not only time-consuming but also lack robustness, making them unsuitable for effectively detecting small ships in complex scenarios [35].

With the development of neural networks, deep-learning-based methods have attracted increasing attention in ship detection. Among these methods, object-detection neural network architectures, especially the YOLO framework, shine with its remarkable equilibrium between speed and accuracy, facilitating the swift and dependable identification of objects within images. Over the years, the YOLO series has undergone multiple iterations, with each successive version building upon its predecessor to overcome limitations and enhance performance. For instance, to address the challenge of low accuracy in the identification of small boats at sea, Chen et al. [44] proposed a novel approach that utilizes Gaussian Mixture Wasserstein Generative Adversarial Networks (GAN) for generating samples of small ships, thereby enhancing the precision of the algorithm. To mitigate the impact of noise interference on ship detection, Li et al. [45] proposed a minimal pooling detection method that effectively suppresses noise without introducing excessive parameters. However, during the feature fusion stage, some information loss occurs. Zhou et al. [46] optimized the performance of small-ship detection based on YOLOv5, enabling the application of the object-detection neural network architecture on maritime equipment. Tang et al. [47] introduced an attention mechanism for multi-scale receptive fields convolution block to enhance the lightweight and high-precision detection of ships at various scales. This module efficiently captures inter-channel relationships within feature maps, thereby improving the learning of ship-background relationships. Wang et al. [48] introduced a Small Proposal Detection Convolution (SPD-Conv) method aimed at improving the detection accuracy of small targets and low-resolution ship images. However, the model's excessive parameterization has led to suboptimal real-time performance. Addressing challenges such as limited feature information for small ships in maritime images and low detection accuracy, Chen et al. [49] introduced a dual-channel attention mechanism aimed at enhancing small object-detection capabilities. However, the recognition performance of this method is still inadequate in complex environments. Furthermore, Wang et al. [50] leveraged a multi-path aggregation network to reuse shallow-level features during the feature fusion stage, thereby optimizing small-ship detection in complex water environments. However, the effectiveness of this approach in identifying small ships at long distances remains suboptimal, and the real-time detection speed is relatively slow. To mitigate the impact of preset anchor box sizes and imbalanced ship samples in their dataset, Zhang and Hou [51] reinforced small object feature information in deep feature aggregation and devised a lightweight convolutional module. Li et al. [52] proposed a feature fusion model incorporating the Swin Transformer, which effectively integrates various detailed features of small vessels. However, the extended inference time of the neural network architecture limits its suitability for real-time detection. To enhance the capability of real-time detection, Zhang et al. [53] introduced an architecture that integrates a coordinate attention mechanism into the base of YOLOv5. While this approach maintains real-time performance, it notably improves accuracy. However, the neural network architecture still demonstrates limited accuracy in identifying targets at long distances or in environments with

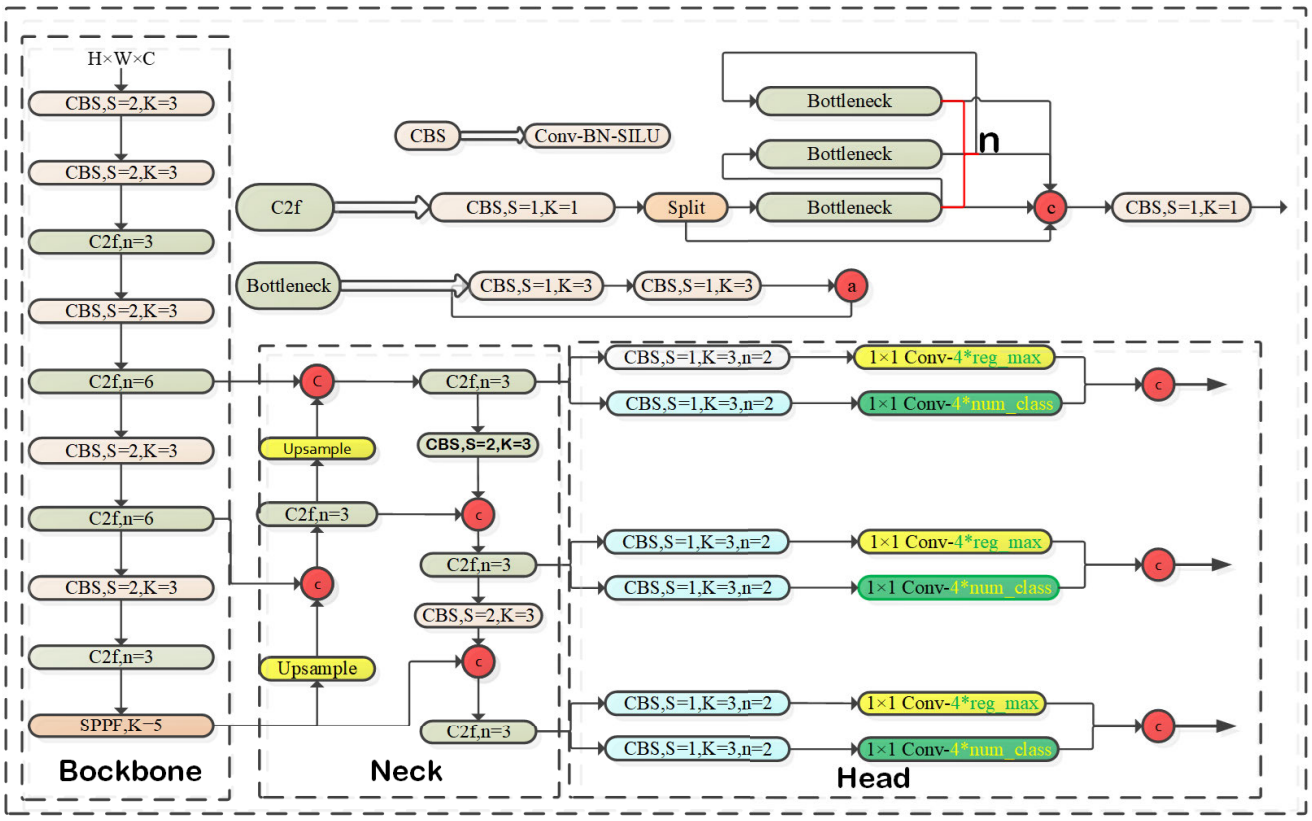


**FIGURE 1.** The structure of the YOLOv8 model.

significant interference. In recent years, numerous scholars have explored how to integrate traditional target detection methods with deep learning techniques. Among these efforts, Li et al. [52] developed a multi-level detection method that achieves high accuracy in detecting distant ships at sea. However, this method has its limitations: it can only detect ships near the sea horizon and is significantly affected by weather conditions.

### B. REVIEW OF YOLOv8

Among the YOLO series methods, YOLOv8 has high accuracy, fast speed while maintaining a relatively small model size, making it suitable for deployment on edge devices. FIGURE 1 illustrates the specific structure of YOLOv8, which comprises three sections: the backbone, neck, and head. The primary function of the backbone is to extract features. The neck integrates extracted features to generate feature maps at various scales, while the head is responsible for target detection output. The core of the back-bone is the Cross Stage Partial Darknet (CSPDarkNet) structure, including the Conv-BN-SILU (CBS), C2f, and spatial pyramid pooling fast (SPPF) modules. The CBS module integrates convolution, batch normalization, and the SiLU activation function. The C2f module, inspired by the structure of Densely Connected Convolutional Networks (Densenet), includes more skip connections, removes convolution operations within branches, and introduces additional split operations. These enhancements aim to enrich features while reducing computational complexity. The SPPF module conducts three consecutive pooling operations, reducing computational demands and combining the output of each layer to achieve multi-scale fusion while enlarging the receptive field. In the neck, a path aggregation feature pyramid network (PAFPN) is implemented, significantly enhancing the detection capabilities for objects of various scales. The head employs decoupled structures and incorporates the CIoU loss function and distribution focal loss (DFL) function for bounding box loss, along with binary cross-entropy for classification loss.

### C. SELF-ATTENTION MECHANISM

In traditional neural networks, each neuron relies solely on outputs from the preceding layer, potentially overlooking specific image details. However, attention mechanisms enable neurons to not only consider outputs from the previous layer, but also to selectively weigh different parts of the input data. This selective weighting allows the model to prioritize important information within the input sequence, enhancing accuracy and efficiency [54].

Dot-product attention is a commonly used attention mechanism widely applied in natural language processing. In recent years, it has also demonstrated excellent

performance in various tasks of image processing. The principle of dot-product attention is as follows:

If there are two vectors $Q$ and $K$, respectively, Dot-product attention measures the similarity between $Q$ and $K$ by taking their dot product, and then normalizes the weight of each $K$ using the softmax function, as follows:

$$Attention(Q, K, V) = softmax(QK^T)V \tag{1}$$

where $Q$ represents the query vector, $K$ represents the key vector, and $V$ represents the value vector. The advantage of dot-product attention is effectiveness in capturing local relationships within input sequences. However, there is a drawback in handling scale differences between query and key vectors, which may lead to numerical stability issues in the output results. Thus, dot-product attention is typically scaled by dividing the dot product by $\sqrt{d_k}$, where $d_k$ represents the dimensionality of the key vectors. This normalization helps stabilize the variance of attention weights, thereby enhancing the robustness and accuracy of the model. Consequently, the commonly used formula for dot-product attention is as follows:

$$softmax(QK^T/(\sqrt{d_k})V \tag{2}$$

The self-attention mechanism, widely employed currently, establishes global dependencies by associating different positions within a single sequence to compute the attention mechanism of the same sequence. This expands the receptive field of the image, enabling the acquisition of more contextual information.

The fundamental principle of the self-attention mechanism is to compute attention weights based on the relationship between queries, keys, and values. These weights are then multiplied with the values to obtain a weighted sum at each time step, thereby yielding the final output result.

The basic principle of the self-attention mechanism involves a sequence $X = [x_1, x_2, \ldots, x_n]$, where $x_i$ denotes the $i^{th}$ element of the sequence. For each element $x_i$, its query, key, and value are computed individually:

$$Q_i = x_i W_Q \tag{3}$$
$$K_i = x_i W_K \tag{4}$$
$$V_i = x_i W_v \tag{5}$$

where $W_Q$, $W_K$ and $W_v$ are the weight matrices for query, key, and value linear transformations, respectively. Their dimensions are $d_Q \times d_X$, $d_K \times d_X$, and $d_v \times d_X$ respectively, where $d_X$ represents the dimensionality of the input sequence, and $d_Q$, $d_K$, and $d_v$ represent the dimensions of query, key, and value, respectively.

## III. METHODS

In this section, we first briefly introduce the basic structure of Ship-YOLOv8. Subsequently, we provide detailed explanations for each improvement, including the C-BoTNet, the VoVGSCP module based on GSConv [58], the integration of an additional small-target detection head, and the optimization of the W-IoU loss function [59].

### A. SHIP-YOLOv8

FIGURE 2 shows the overall schematic of Ship-YOLOv8. The modules within the colored solid-line boxes represent improvements to the original model; the rest remain unchanged. The model comprises three main components: the backbone network, the feature enhancement part, and the prediction part.

(1) We integrated C-BoTNet into the backbone, merging local and global object features, enriching the detection process, and improving the identification of long-distance maritime ships [60]. (2) In the neck, we replaced the C2f with the VoVGSCSP [58]. Depthwise separable convolution was applied to the feature maps, facilitating interaction between channels. This optimization not only reduces model complexity but also enhances the neural network architecture's ability to detect small ships. (3) In the neck network, we performed resampling and merged the generated $160 \times 160$ feature map with the feature map of the same size from the backbone network, adding an additional layer for small object detection. The added detection layer improves the architecture's accuracy significantly in detecting long-distance maritime ships. (4) The W-IoU loss function was employed for optimization, effectively balancing samples of varying quality in the dataset, and further enhancing the overall accuracy of the neural network architecture [59].

### B. C-BOTTLENECK TRANSFORMER

The original Bottleneck Transformer (BoT) [60] is a simple but powerful backbone. It merely replaces the $3 \times 3$ convolutions in ResNet with a multi-head self-attention mechanism (MHSA). This modification enhances network performance significantly for tasks such as instance segmentation and object detection. The structures of the original ResNet and the BoT are illustrated in FIGURE 3 [61]. The $\oplus$ symbol indicates element-wise addition across each channel.

The C-BoTNet and C-Bottleneck Transformer (C-BoT) structures designed in this paper are illustrated in FIGURE 4. As depicted in FIGURE 4(b), our newly proposed residual connection structure, C-BoT, is based on the BoT. This structure merges the CBS module with the MHSA. C-BoT is outlined as follows. Initially, a $1 \times 1$ convolutional kernel is employed to adjust the channel count, facilitating the aggregation of features from distinct small ships. Subsequently, the MHSA mechanism is adeptly utilized to integrate both local and global features from the feature maps, specifically focusing on amalgamating distinct characteristics of small ships. This strategic fusion significantly enhances the system's ability to detect small maritime targets at long distances. Ultimately, a residual structure is integrated to enable the network to discern variations in distinct ship features. This facilitates effective gradient propagation, mitigates the vanishing gradient issue, and ensures the successful aggregation of diverse ship characteristics. The symbol $\oplus$ indicates element-wise addition across each channel. Taking inspiration from the C2f module, we introduce the C-BoTNet design, building upon the foundation of C-BoT.

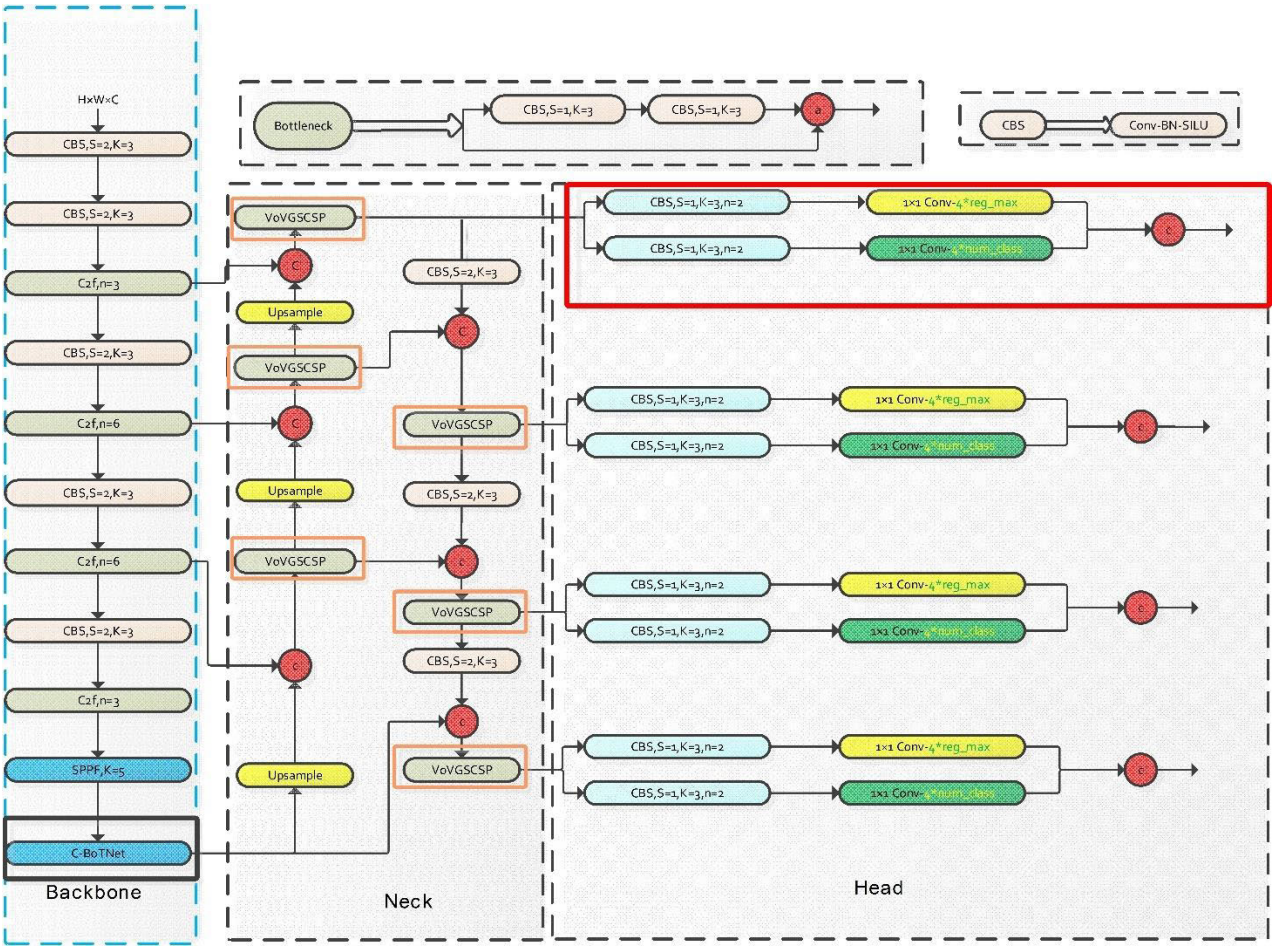


**FIGURE 2.** The structure of Ship-YOLOv8. The modules within the solid-line boxes are improvements to the original model.

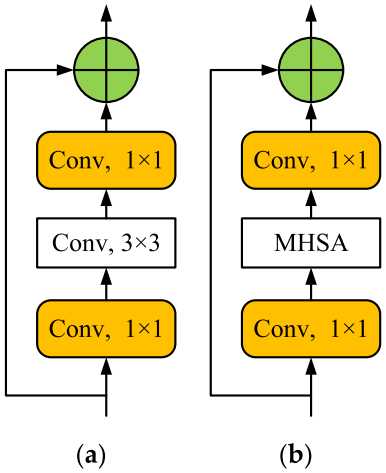


**FIGURE 3.** (a) ResNet Bottleneck block; (b) BoT block.

The structure of C-BoTNet is depicted in FIGURE 4(a). The input feature map $F \in R^{h \times w \times C_1}$ undergoes processing by the CBS module, where the number of channels. is adjusted to $0.5 \cdot C_2$ using a $1 \times 1$ convolution. Following this, computations are performed using the C-BoT module, an operation that leaves the size of the feature map unchanged. Subsequently, the input feature map undergoes operations through the CBS module, concatenating via a residual connection to adjust the channel number to $C_2$. Finally, a CBS module operation is used to reshape the feature map to $h \times w \times C_1$. C-BoTNet merges local and global object features, enriching the detection process and improving the identification of small maritime ships.

The integration of C-BoTNet enhances the process of detecting small maritime ships by combining local and global object features, leading to improved long-distance identification.

This integration takes place in the concluding phase of the backbone, a determination derived from experiments identifying optimal accuracy in small ship recognition. Detailed experiments pertaining to this aspect are expounded in Section F.

In FIGURE 4, the core of C-BoTNet is Multi-Head Self-Attention (MHSA), as depicted in FIGURE 5. Here, q, k, v, and r respectively represent query, key, value, and positional

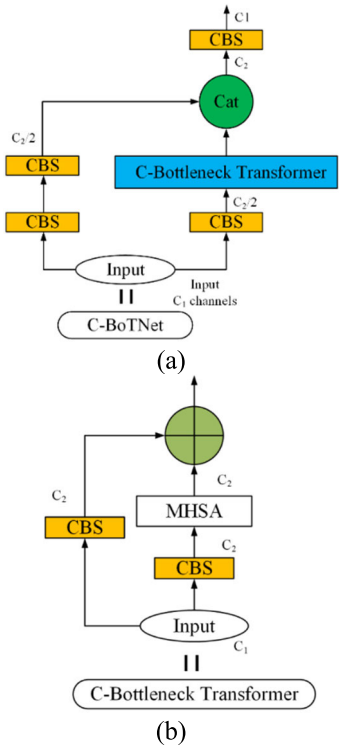


**FIGURE 4.** (a) The structure of C-BoTNet; (b) The structure of the C-BoT.

encoding. The symbol $\oplus$ denotes element-wise sum, $\otimes$ signifies matrix multiplication, and $1 \times 1$ denotes a pointwise convolution.

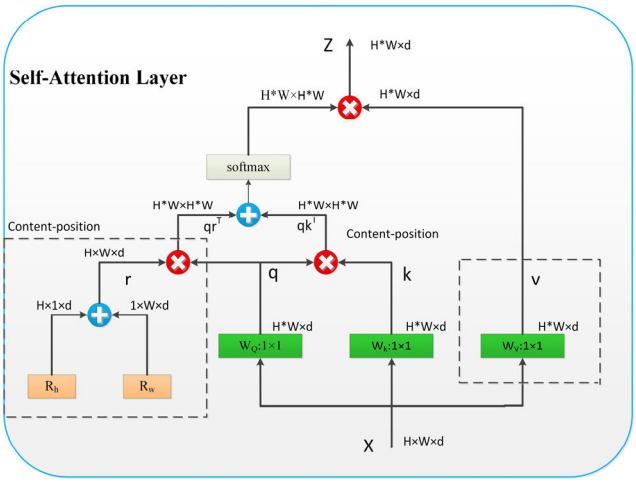


**FIGURE 5.** Multi-head self-attention mechanism.

### C. ADDIYIONAL DETECTION LAYER

In YOLOv8, small-target detection effectiveness is hampered by the diminutive sizes of target samples and the relatively high downsampling factor. The latter poses challenges for deeper feature maps in capturing the intricate details of these smaller targets. Therefore, we added an extra detection layer to the three original detection heads. In our enhanced backbone, downsampling was employed to generate feature maps of five distinct sizes. In the neck, concatenation fusion began with the up-sampling of large feature maps of $160 \times 160$, $80 \times 80$, and $40 \times 40$ pixels generated by the backbone. Subsequently, concatenation fusion was carried out with the $20 \times 20$-pixel feature map after downsampling. This process resulted in the creation of four feature maps that integrated diverse semantic information. Finally, four detection layers were output separately, with the additional detection layer integrating the $160 \times 160$-pixel feature maps from the fusion backbone. This integration specifically enhances the detection of small targets, thereby enabling more effective long-distance recognition.

### D. VoVGSCSP

The introduction of a new detection layer, specifically designed for small-target recognition, has inevitably increased the model's parameter count, posing challenges for real-time detection. To address this, GSConv [58], representing a novel convolution mode, has been employed. GSConv merges the lightweight characteristics of depth-separable convolution (DSC) with the accuracy of standard convolution (SC). It initially performs a DSC operation on the information obtained from the SC, and then reorganizes this information across various channels to produce the final output. This strategy not only facilitates the efficient reutilization of feature information but also significantly reduces computational complexity. Consequently, it achieves a notable improvement in the model's balance between accuracy and speed, maintaining high detection accuracy while catering to real-time processing requirements. FIGURE 6 illustrates the structural design of GSConv.

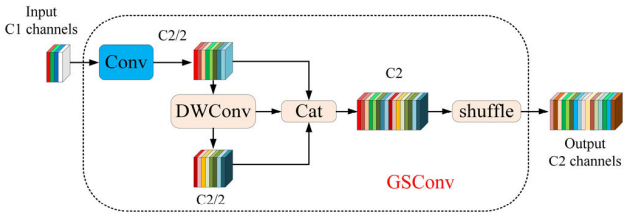


**FIGURE 6.** The structure of GSConv.

Based on GSConv, Hu et al. make an improvement by introducing GSbottleneck, as illustrated in FIGURE 7 (a). Subsequently, as illustrated in FIGURE 7 (b), a one-time aggregation approach was employed to design the inter-level subnetwork VoVGSCSP module [58].

VoVGSCSP enhances GSConv by splitting the input feature map into two segments. In this architecture, the feature map is bifurcated: one path processes features through the hybrid Conv and GSConv structure, and a parallel path applies a solitary Conv layer, acting as a residual connection. Subsequently, the two segments are merged and linked to the output via Conv convolution. The unique architecture of VoVGSCSP enables easy dimensionality manipulation

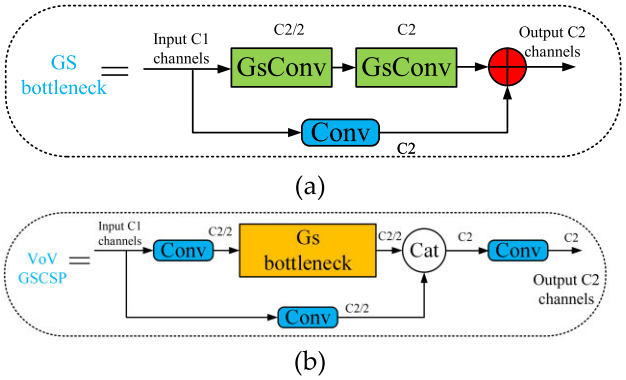


**FIGURE 7.** (a) Structure of GS bottleneck module; (b) Structure of VoVGSCSP module.

and reductions in feature dimensions, decreasing computational load. Typically, the time complexity of a convolutional computation is quantified in terms of giga-floating-point operations (GFLOPs).Table 1 compares the different levels of VoVGSCSP module optimizations applied to YOLOv8, with parameters and GFLOPs listed.

**TABLE 1.** Parameter and GFLOP comparison: YOLOv8 with different VoVGSCSP optimizations.

| Layers in Neck | Parameters | GFLOPs |
|---|---|---|
| Base | 3011043 | 8.2 |
| VoVGSCSP×1 | 2932833 | 7.9(-3.7%) |
| VoVGSCSP×2 | 2924211 | 7.7(-6.1%) |
| VoVGSCSP×3 | 2910003 | 7.6(-7.3%) |
| VoVGSCSP×4 | 2891379 | 7.5(-8.5%) |

In Table 1 "base" refers to the model without VoVGSCSP optimization. "VoVGSCSP×1" represents the model optimized with one C2f layer using VoVGSCSP, and "VoVGSCSP×4" represents the model optimized with four C2f layers using VoVGSCSP. The data show that incorporating the VoVGSCSP module for neck optimization significantly reduces the model's parameters. The degree of optimization with VoVGSCSP is directly correlated with the model's lightweight design. Subsequent experiments confirm that VoVGSCSP not only streamlines the model but also enhances its detection performance for long-distance ships.

In our neural network architecture, the C2f component within the neck is optimized using the VoVGSCSP module. This refinement not only reduces the model's complexity but also significantly enhances the algorithm's precision in long-distance detection of small targets. It is important to note that the GSConv is exclusively utilized in the neck, where it processes feature maps characterized by maximal channel numbers and minimal spatial dimensions. At this juncture, these feature maps exhibit minimal redundancy, obviating the need for compression and thus allowing the module to operate more effectively.

### E. LOSS FUNCTION

The total loss of Ship-YOLOv8 comprises two components: classification loss and regression box loss:

$$Loss = k_1 * loss_{rect} + k_2 * loss_{cls} \tag{6}$$

where $loss_{rect}$ represents the regression box loss and $loss_{cls}$ represents the classification loss. The coefficients $k_1$ and $k_2$ weigh the two losses and are typically set to 0.5. The variable $loss_{cls}$ is calculated as follows:

$$loss_{cls} = \frac{1}{N}\sum_{i}^{N}\left[-y_i \cdot \log(p_i) + (1-y_i)\cdot \log(1-p_i)\right] \tag{7}$$

Distribution focal loss (DFL) and W-IoU are used to compute $loss_{rect}$. DFL is calculated as follows:

$$DFL(S_i, S_{i+1}) = -((y_{i+1}-y)\log(S_i) + (y-y_i)\log(S_{i+1})) \tag{8}$$

where $S_i$ and $S_{i+1}$ denote the probabilities corresponding to the predicted values $y_i$ and $y_{i+1}$ in the vicinity of the label $y$.

In line with the characteristics of the LDSD, the determination of the bounding box loss in our approach is governed by the W-IoU loss function [59]. This methodology is crafted to achieve equilibrium in the detection of model training images with diverse qualities, ultimately elevating the precision of detection results.

W-IoU introduces a dynamic focusing mechanism (FM) designed to estimate the extent of outliers within the anchor box. The FM enhances anchor box regression by assigning small gradient gains to precision ship bounding anchors, directing the focus towards regular-quality ship bounding anchors. Simultaneously, it allocates limited gradient gains to inferior anchor boxes, effectively channeling the loss function towards regular-quality ship instances.

The W-IoU formula is shown in Equations (9) – (12):

$$L_{WIoU} = r \times \exp\left(\frac{(x-x_{gt})^2+(y-y_{gt})^2}{\left(W_g^2+H_g^2\right)^*}\right)(1-IoU) \tag{9}$$

$$\beta = \frac{L_{IoU}^*}{\overline{L_{IoU}}} \in [0, +\infty) \tag{10}$$

$$r = \frac{\beta}{\delta\alpha^{\beta-\delta}} \tag{11}$$

$$L_{IoU} = 1 - IoU \tag{12}$$

In this context, W-IoU introduces an outlier parameter, denoted as $\beta$, for assessing anchor box quality and establishes a non-monotonic focusing factor, denoted as r. A lower outlier value indicates higher precision for ship bounding anchors. Consequently, we allocate a smaller gradient gain to such anchors, directing the bounding box regression towards those of standard quality. Due to the intricate characteristics of LDSD, there are instances with relatively high outlier values representing imprecise ship bounding anchors. We similarly assign a smaller gain to these anchors, preventing the neural network architecture from overfitting to such

specific instances and thereby improving its generalization performance. In addition, In the formula, $\delta$ and $\alpha$ are hyper-parameters that can be tuned to suit various models. Terms $W_g$ and $H_g$ correspond to the width and height, respectively, of the smallest encompassing box. The superscript $*$ denotes detachment from the computational graph, aiming to reduce computational burden, enhance algorithmic robustness, and expedite convergence. W-IoU is depicted in FIGURE 8, A denotes the predicted box, and B represents the real box.

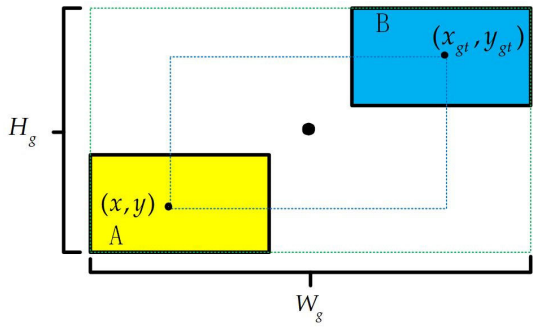


**FIGURE 8.** Schematic representation of formula parameters.

Owing to the intricate nature of the environment and the diminutive pixel size of the ships, LDSD comprises anchor boxes of lower quality. To address this issue, we incorporate W-IoU into the object-bounding box regression loss. This enhancement enables the model to focus more on learning anchor boxes of average quality, thereby improving its object localization capabilities.

## IV. EXPERIMENTS AND RESULTS

To evaluate the effectiveness of our proposed object-detection model, we conducted extensive experiments on the LDSD.

This section is organized as follows: Initially, we outline the experimental setting and training specifics. Then, the efficacy of each enhanced structure is confirmed through multiple ablation experiments, followed by the verification of Ship-YOLOv8's efficiency in comparative experiments. Finally, we present a visual analysis of the detection performances of both YOLOv8 and Ship-YOLOv8 on the LDSD to validate our methodology.

### A. EXPERIMENTAL PLATFORM

The experimental setup employed for this study is detailed in Table 2. In these experiments, we used the stochastic gradient descent with a weight decay of 0.0005 and a momentum of 0.937 to train for 300 epochs on the LDSD.

### B. METRICS

A set of evaluation criteria has been selected to more effectively compare the performance among various algorithms. This study primarily employs accuracy and speed as key metrics for measurement. The specific details are outlined below.

**TABLE 2.** Configuration parameters for the experiment.

| Configuration | Parameter |
|---|---|
| CPU | Intel(R) Core (TM) i7-10700 |
| GPU | NVIDIA GeForce RTX3090ti |
| Operating System | Linux |
| Frame | Pytorch |
| Torch | 1.8.1 |
| CUDA | 11.1 |
| Batch Size | 16 |
| Image Size | 640 |
| $\delta$ | 3 |
| $\alpha$ | 1.9 |

Precision ($P$) is the most intuitive performance evaluation metric, which is defined as the number of correctly predicted positive instances divided by the total number of identified objects. It can be expressed as:

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

where $TP$ is the number of ships recognized as ships and $FP$ is the number of backgrounds recognized as ships. However, dependence solely on accuracy proves insufficient. In object detection, the imbalance in positive and negative sample distribution impacts the detection accuracy of the neural network architecture significantly. This study also employed recall $R$ to evaluate the neural network architecture:

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

where $FN$ is the number of ships detected as backgrounds.

The Precision-Recall (PR) curve delineates the precision and recall values computed across various confidence thresholds. Average Precision (AP) quantifies the area under this curve, signifying the average accuracy at distinct recall points. For neural network architectures.

That detect multiple categories simultaneously, the mean average precision (mAP) is used to measure their overall performance. This work comprised only one category, so $AP$ was used, described in the following manner:

$$AP = \int_0^1 P(R) \cdot d(R) \tag{15}$$

Given the potential bias of relying on a single metric, the F1 score, encompassing both precision $P$ and recall $R$, served as a comprehensive measure to reflect the detection accuracy and recall performance of the algorithm:

$$F1 = \frac{2 \times R \times P}{R + P} \tag{16}$$

### C. DATASET

Currently, there are few publicly available datasets for maritime ships in visible light images, with notable examples

being the Sea Ships dataset [55] and the Singapore Maritime dataset. The Sea Ships dataset comprises 31,455 images and covers a wide range, making it a comprehensive collection for maritime ship studies. The Singapore Maritime dataset [56], consisting of images captured in the waters near Singapore, includes a variety of complex scenarios and is also an extensive dataset. However, these public datasets feature a limited number of long-distance ship instances. The reliability of a dataset is crucial for the neural network architecture's detection performance. In response to this limitation, we developed the Long-Distance Small Maritime Ship Dataset (LDSD), utilizing high-definition cameras mounted on a ship to capture images, specifically focusing on small and distant maritime ships. We labeled images manually to identify small maritime ships and converted them into the COCO [57] format. This resulting dataset comprises 1871 images and 4348 ships, and we randomly partitioned it at a 9:1 ratio for training and validation. In other words, 90% of the data were allocated for training, while the remaining 10% were reserved for validation. Images of small ships within the LDSD are depicted in FIGURE 9.

As depicted in FIGURE 9, the example images from our collected dataset of long-distance ships include various challenging scenarios. These scenarios encompass images with few pixels, ships obscured by waves, and ships easily confused with the sea–sky background. The diverse and complex images under different conditions pose significant challenges for the algorithm's recognition capabilities.

We conducted a statistical analysis of the target count in each image of the LDSD. FIGURE 10 illustrates the distribution of number of ships in each image. The horizontal axis represents the number of ships per image, while the vertical axis indicates the corresponding number of images. Our analysis revealed that each image contains between 1 and 14 ships. Specifically, the first cylinder cluster has 958 images comprising only one ship, resulting in a total number of ships equal to $1 \times 958 = 958$. Additionally, in the last cylinder cluster, the four images have 14 ships each, containing a total of $4 \times 14 = 56$ ships. FIGURE 11 displays a pie chart to depict the proportions of images with different ship counts. The statistical chart shows that the distribution of ship counts varies across the images in the dataset, with the majority having four or fewer targets per image.

As illustrated in FIGURE 11 and FIGURE 12, we conducted a pixel size analysis of all targets within the LDSD. The minimum pixel value in the dataset was 4 ($2 \times 2$), with the majority falling below 150. The definition of small targets was 9 established based on the absolute pixel sizes of the targets, with the most widely accepted criteria derived from the commonly used MS COCO dataset for object detection. Targets with dimensions below $32 \times 32(1024)$ pixels were considered small. The average pixel value for the targets in the LDSD was 229, significantly below the defined threshold for small-target pixels. The statistical analysis indicated that an overwhelming majority of ship targets in LDSD qualified as extremely small, presenting a significant challenge for detection.

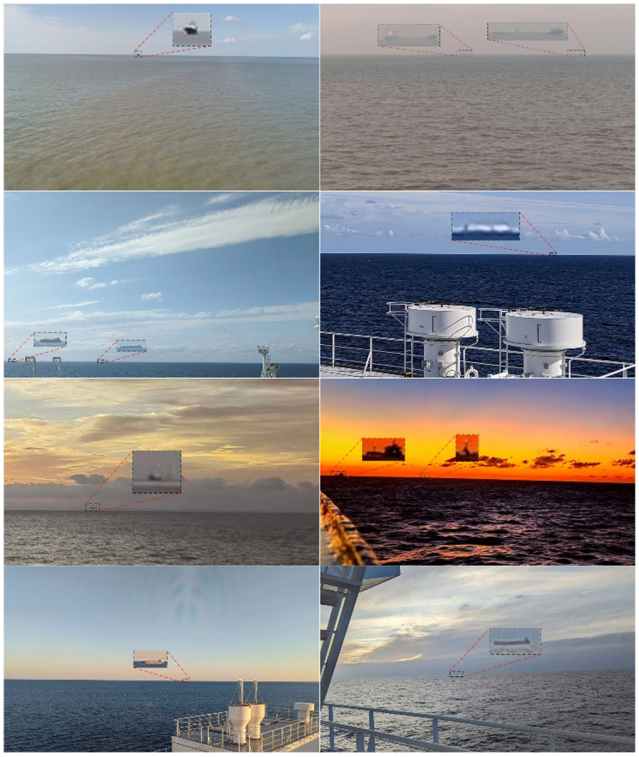

**FIGURE 9.** Samples from the LDSD.

### D. ABLATION EXPERIMENTS

To gauge the efficacy of the improved modules in Ship-YOLOv8, we conducted ablation experiments to analyze the impacts of different modules on the detection performance. The results of the experiments are presented in Table 3, where $\surd$ indicates the use of the improved strategy, ADL indicates an additional detection layer and $AP_{0.5}$ represents the AP at an IoU threshold of 0.5.

The ablation experiments demonstrate that integrating C-BoTNet, VoVGSCSP, additional detection layers, and the W-IoU loss function into the baseline YOLOv8 algorithm significantly enhances the detection accuracy of long-distance maritime ships in the LDSD. These enhancements are primarily evidenced by the improved $AP_{0.5}$ scores on the validation set, indicating a notable rise in the model's precision for ship detection tasks.

The primary reason for the baseline model's low AP0.5 in detecting ships on the LDSD is the small pixel representation of distant ships and smaller targets. This results in a loss of detailed features after downsampling through the backbone network. The integration of the C-BoTNet module effectively merges local and global features, improving the $AP_{0.5}$ by 0.2 percentage points. The introduction of additional detection layers significantly enhanced the model's accuracy

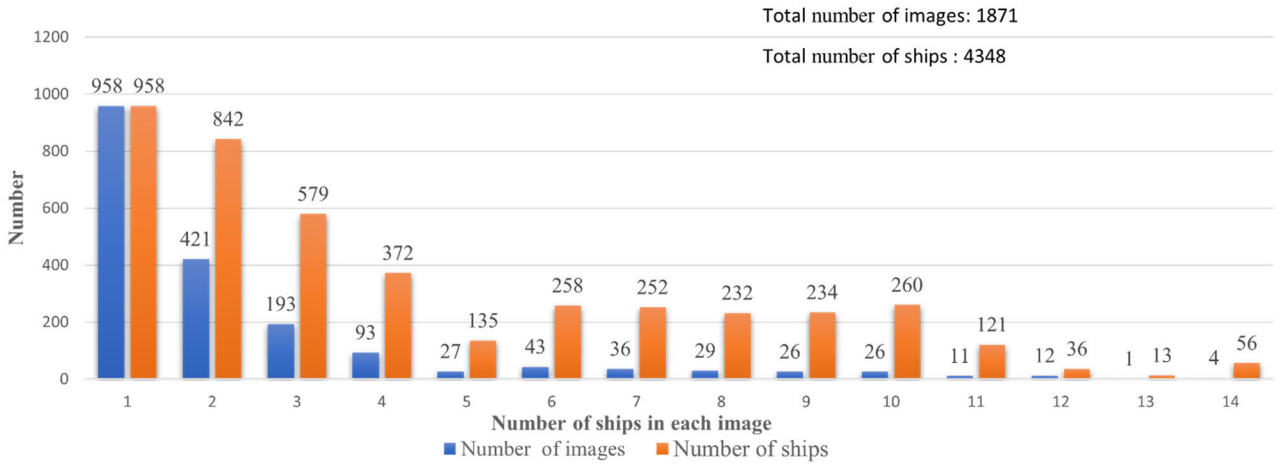


**FIGURE 10.** Ship count distribution in images.

**TABLE 3.** Ablation experiments on the LDSD.

| YOLOv8 | C-BoTNet | VoVGSCSP | ADL | W-IoU | $AP_{0.5}$ (%) | Recall (%) | F1 (%) |
|---|---|---|---|---|---|---|---|
| √ | | | | | 70.60 | 51.80 | 65.49 |
| √ | √ | | | | 70.80(+0.20) | 60.30 | 70.27 |
| √ | | √ | | | 72.60(+2.00) | 52.10 | 66.18 |
| √ | | | √ | | 87.50(+16.90) | 77.60 | 83.98 |
| √ | | | | √ | 72.90(+2.30) | 60.20 | 71.76 |
| √ | √ | √ | | | 74.00(+3.40) | 59.10 | 69.72 |
| √ | √ | √ | √ | | 90.20(+19.40) | 81.70 | 86.23 |
| √ | √ | √ | √ | √ | 91.80(+21.20) | 85.20 | 88.42 |

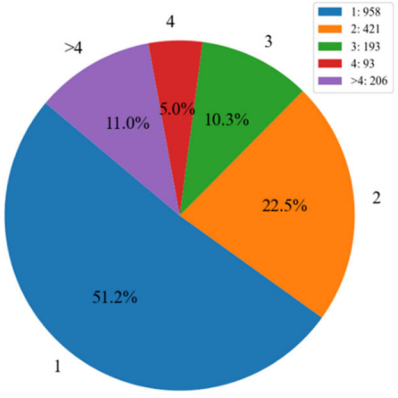


**FIGURE 11.** Proportional distribution of ship numbers in images.

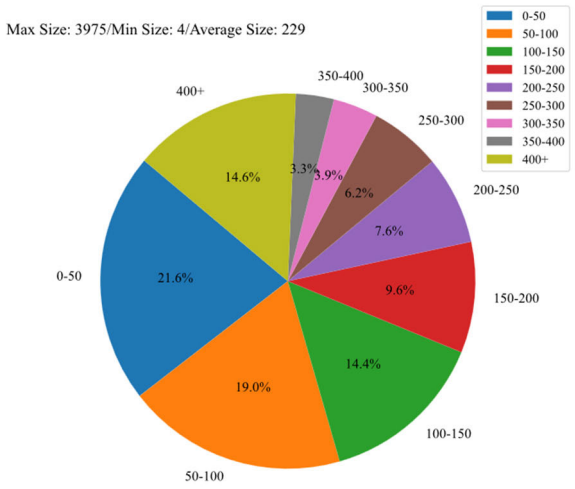


**FIGURE 12.** Distribution of ship pixel sizes in the LDSD.

on the LDSD, achieving an $AP_{0.5}$ of 87.50. This indicates that incorporating the 160 × 160 large feature maps from the backbone network into the neck network effectively supplement the detailed information of the images. In essence, the baseline model's insufficiency in recognizing smaller targets is primarily due to the loss of detailed information of the target objects, leading to an inability to correctly identify distant ships. The inclusion of both the C-BoTNet and additional detection layers increases the model's parameter count and complexity. To address this, we integrated the VoVGSCSP module in the neck network to lighten the overall

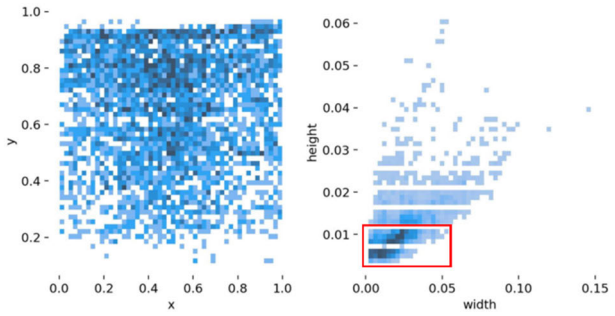


**FIGURE 13.** Distribution of ship pixel sizes and positions in the LDSD.

model and reduce complexity. Notably, the VoVGSCSP optimization not only reduced the model's complexity but also improved the $AP_{0.5}$ by two percentage points. Additionally, employing the W-IoU loss function resulted in a 2.3 percentage point improvement in $AP_{0.5}$ compared to the baseline model. This indicates that the W-IoU loss function effectively balances the quality of different samples in the LDSD, mitigating the impact of poor-quality samples on the algorithm's performance.

Notably, the detection model incorporating all four optimizations—C-BoTNet, VoVGSCSP, additional detection layers, and the W-IoU loss function—achieved the highest overall detection accuracy with an $AP_{0.5}$ score of 91.8. This result demonstrates that the proposed approach of enhancing target detection and balancing the dataset's sample quality from two perspectives is effective.

In summary, the effectiveness of the four optimization schemes has been demonstrated through eight sets of ablation experiments. It has been shown that utilizing any single optimization or combining multiple optimizations can effectively enhance the performance of the algorithm.

### E. COMPARATIVE EXPERIMENTS

This study compared the proposed Ship-YOLOv8 with several prominent object detection algorithms, including one-stage algorithms, such as YOLO and the SSD series [20], and representative two-stage algorithms, such as Faster R-CNN [24]. Furthermore, our comparison included both anchor-based and anchor-free models.

Table 4 presents a comparison of Ship-YOLOv8 with other mainstream algorithms based on three metrics: $AP_{0.5}$, Recall, and F1. Our proposed algorithm achieved an $AP_{0.5}$ of 91.8%, surpassing YOLOv8 by 21.2 percentage points, and exceeding the well-known small object detection algorithm TPH-YOLOv5 [62] by 14.6 percentage points. In comparison with numerous state-of-the-art algorithms, our model demonstrated a distinct advantage. Our algorithm achieved the highest recall of 85.2%, surpassing YOLOv8 by 33.4 percentage points. Specifically, it outperformed Centernet and YOLO-Fastestv2 by 12.26 and 0.5 points, respectively, and significantly outperformed algorithms such as Faster-RCNN and SSD. For the F1 score, our algorithm also achieved the highest F1 of 88.37%, surpassing the YOLOv8 algorithm by 22.93% points. Considering these three metrics, our method performs better in the long-distance ship detection, providing assurance for effective remote obstacle avoidance in intelligent vessels.

**TABLE 4.** Comprehensive detection performance comparison of different algorithms on the LDSD.

| Method | $AP_{0.5}$ (%) | Recall (%) | F1 (%) |
|---|---|---|---|
| Centernet | 64.58 | 72.94 | 75.00 |
| Faster-RCNN | 39.70 | 41.88 | 46.00 |
| SSD | 63.20 | 35.50 | 47.27 |
| TPH-YOLOv5 | 77.20 | 68.50 | 77.47 |
| YOLOv5 | 39.46 | 41.88 | 46.00 |
| YOLO-Fastestv2 | 81.04 | 84.70 | 81.91 |
| YOLOv8 | 70.60 | 51.80 | 65.49 |
| Ours | **91.80** | **85.20** | **88.42** |

To further demonstrate the enhanced performance of our optimized model, we created a visualization of the detection results. In FIGURE 14, the first column displays the original images, the second column presents the recognition results obtained using YOLOv8, and the last column shows the results of our proposed Ship-YOLOv8.

FIGURE 14 also shows that our proposed algorithm demonstrated higher accuracy and lower error rates in testing on the LDSD compared to the original YOLOv8. In the (a) group comparison, the original image had a target of 17 × 3 pixels. YOLOv8 failed to detect it, while our algorithm correctly identified it. In the (b) group comparison, two targets were in the original image, but in the evening environment, where the color of the ships was close to that of the water surface, YOLOv8 detected only one target. In contrast, our algorithm identified both targets accurately. In the (c) group comparison, a target in the original image had dimensions of 18 × 3 pixels, which YOLOv8 failed to recognize. However, our detection algorithm could correctly identify it. In the (d) group comparison, three targets were in the original image, and the smallest target was 4 × 3 pixels. YOLOv8 could not recognize the smallest target, while our detection algorithm could accurately identify all three targets. The (e) group comparison had seven targets in the original image, with the smallest being 7 × 7 pixels. Additionally, multiple targets had short spacing, making recognition challenging. YOLOv8 recognized only five targets, whereas our detection algorithm correctly identified all seven targets. Through visualized image presentations, our proposed method performs better in long-distance ship detection in various scenarios, indicating its potential advantages in intelligent ship autonomous obstacle avoidance.

### F. COMPARISON EXPERIMENT FOR C-BoTNet POSITION OPTIMIZATION

In our optimized backbone, we introduce the C-BoTNet module in the final feature stage for enhanced feature recognition and extraction..

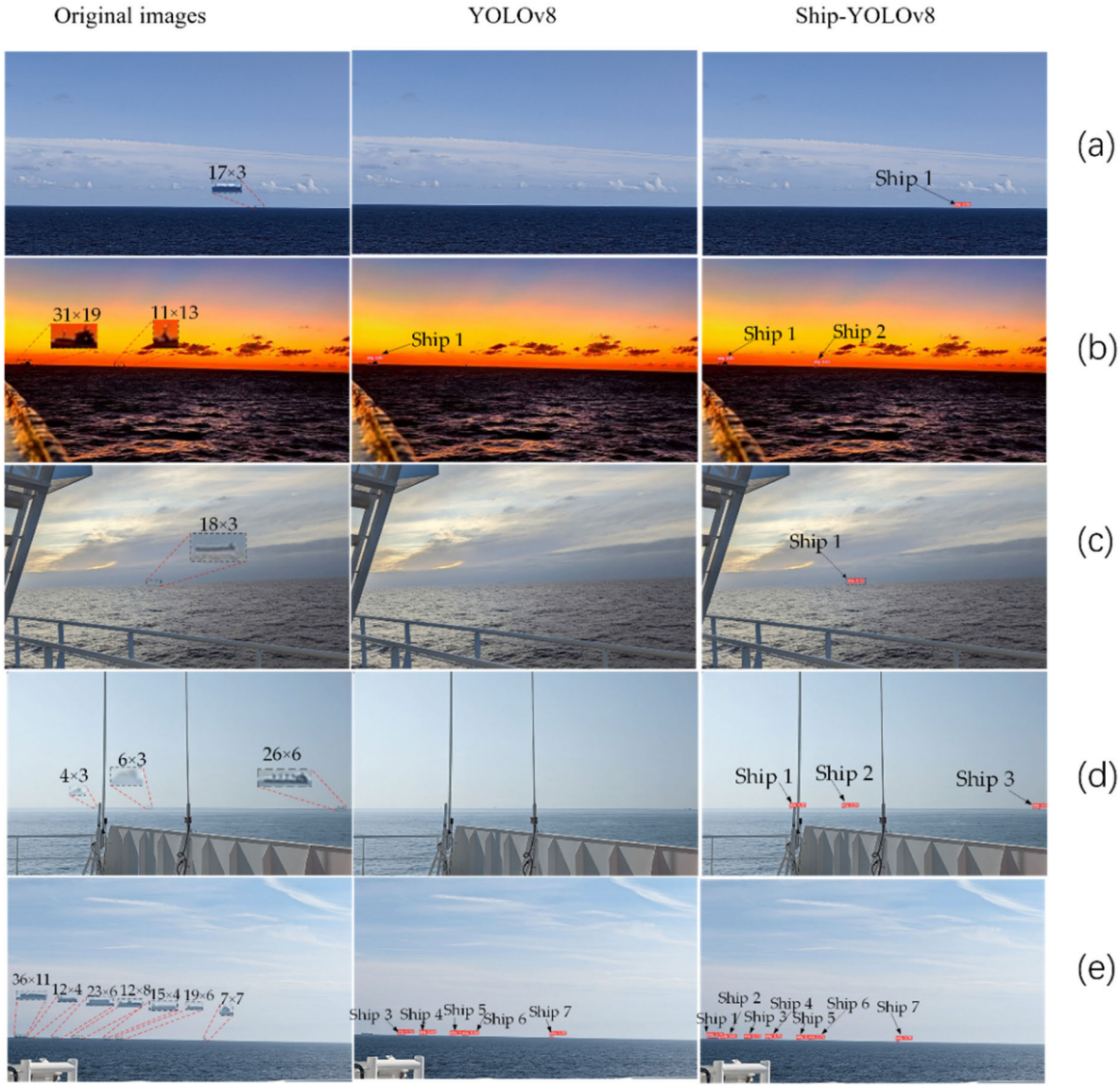


**FIGURE 14.** Comparison of detection results.

**TABLE 5.** Comparative experiments with different positions for C-BoTNet.

| Method | C-BoTNet Position | $AP_{0.5}$ (%) | Recall (%) | F1 (%) |
|---|---|---|---|---|
| C-BoTNet-1 | Middle of Backbone | 67.2 | 57.6 | 62.07 |
| C-BoTNet-2 | End of Backbone | **70.8** | **60.3** | **70.27** |

Table 5 compares the performance metrics between placing C-BoTNet in the middle (C-BoTNet-1) and at the final stage of the backbone (C-BoTNeT-2). The results indicate that C-BoTNet-2 outperformed C-BoTNet-1 in terms of *AP*, recall, and F1-score metrics. The results clearly indicate that positioning C-BoTNet at the end of the main backbone is the optimal choice for enhancing model detection accuracy.

### G. COMPARISON OF INFERENCE TIME

Detection speed is a key indicator to evaluate the real-time performance of the algorithm. Therefore, experiments were conducted on Ship-YOLOv8 and other algorithms, with results summarized in Table 6. Our proposed algorithm achieved a detection speed of 208 FPS and an inference time of 4.8 ms, meeting the requirements for real-time detection and supporting rapid obstacle avoidance in intelligent ships.

**TABLE 6.** Comparison of inference times of different methods in LDSD.

| Method | FPS (Image/Seconds) | Inference Time (Milliseconds/Image) |
|---|---|---|
| Centernet | 48 | 20.7ms |
| Faster R-CNN | 48 | 20.6ms |
| SSD | 143 | 7.0ms |
| TPH-YOLOv5 | 20 | 47.9ms |
| YOLOv5 | 36 | 27.4ms |
| YOLO-Fastestv2 | 3 | 279.1ms |
| YOLOv8 | **476** | **2.1ms** |
| Ours | **208** | **4.8ms** |

### *H. MODEL VISUALIZATION AND ANALYSIS*

FIGURE 15 displays the confusion matrices for YOLOv8 and Ship-YOLOv8. The x-axis of the confusion matrix represents true outcomes, while the y-axis represents predicted results. Examination of these matrices indicates that the enhanced Ship-YOLOv8 achieves superior precision and comprehensive recognition compared to the original YOLOv8.

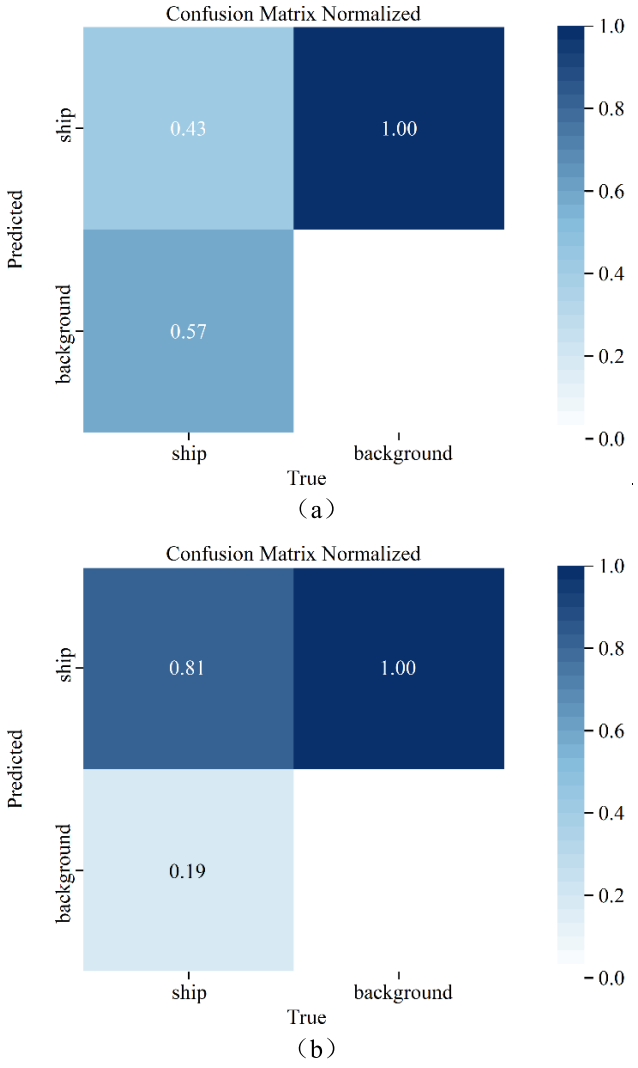


**FIGURE 15.** Confusion matrix comparison chart. (a) Confusion matrix chart for the original YOLOv8; (b) Confusion matrix chart for Ship-YOLOv8.

To evaluate the overall detection performance of Ship-YOLOv8 in LDSD, we con ducted a PR curve analysis for both YOLOv8 and Ship-YOLOv8. The results are depicted in FIGURE 16.

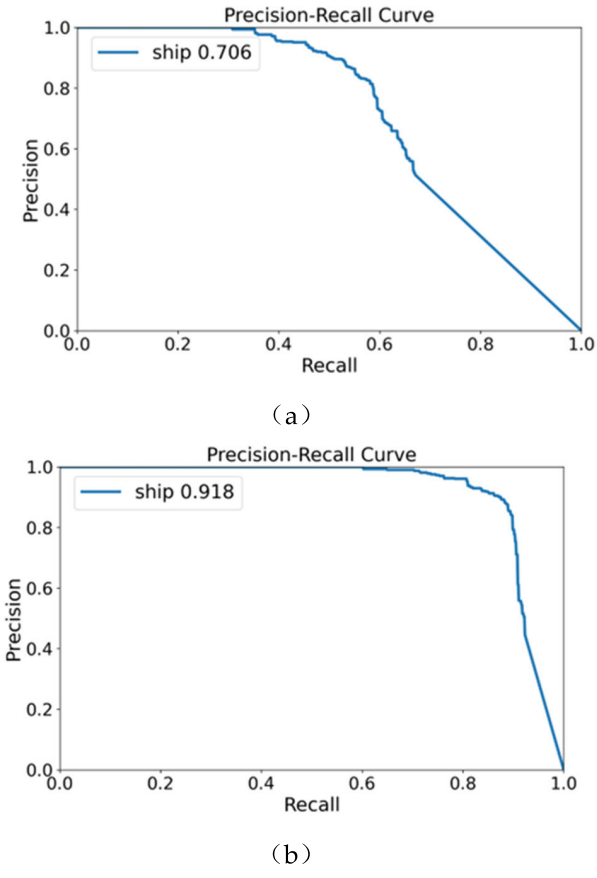


**FIGURE 16.** Comparison of PR curves for the algorithms in LDSD. (a) PR curve for YOLOv8. (b) PR curve for Ship-YOLOv8.

The PR curve represents the relationship between precision and recall. The comparison with YOLOv8 indicates that the PR curve for our proposed Ship-YOLOv8 is notably smoother. The curve encompasses a larger area under the precision–recall space, indicating superior performance. The improved algorithm achieves a better balance between precision and recall across various thresholds.

FIGURE 17 presents a comparative analysis of metrics between Ship-YOLOv8 and YOLOv8. The first three

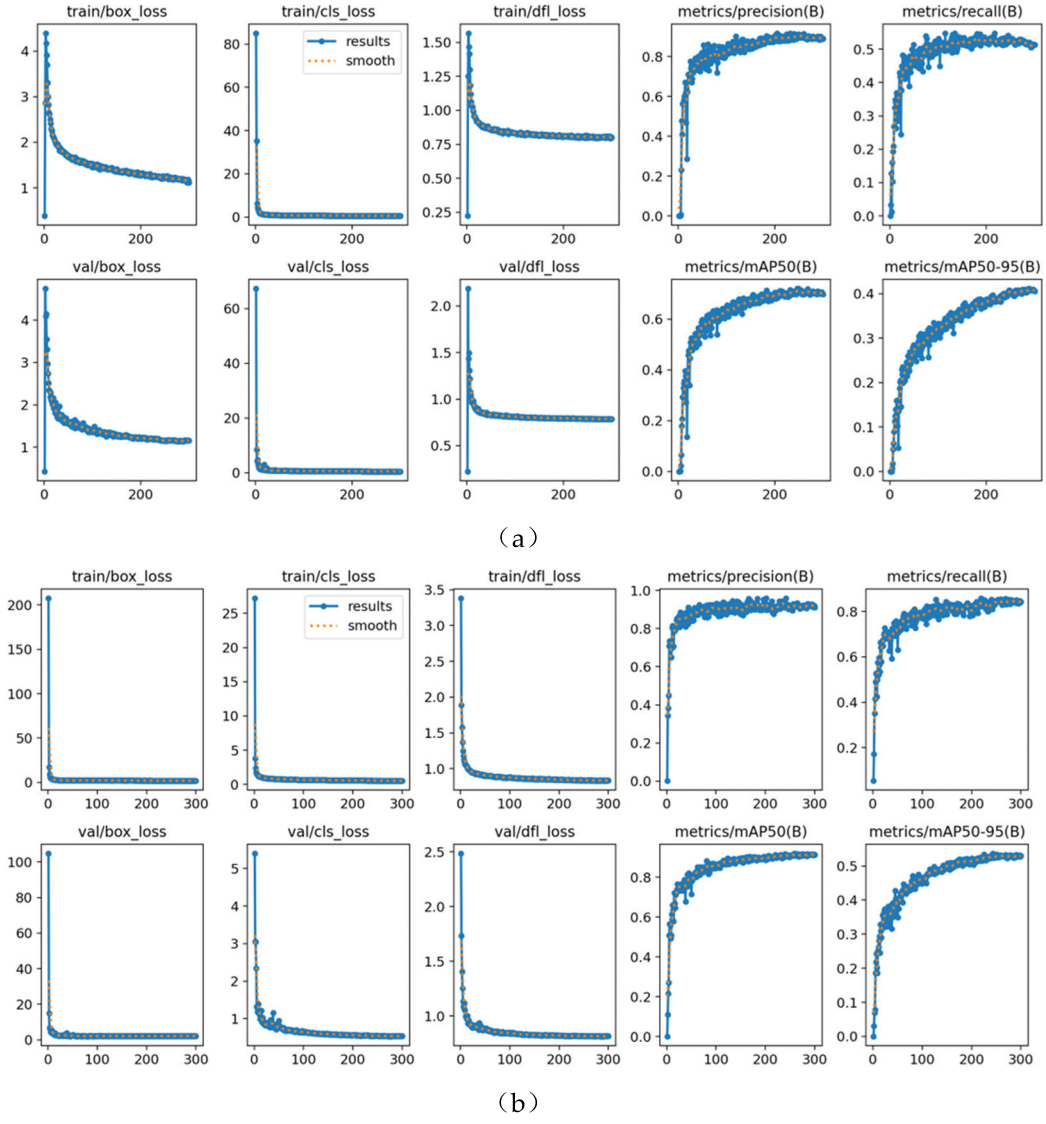


**FIGURE 17.** (a) Comprehensive Performance Curve for YOLOv8; (b) Comprehensive Performance Curve for Ship-YOLOv8.

columns show the model's box loss, cls loss, and dfl loss, with the x-axis representing training epochs and the y-axis representing the total loss. From the comparison results in (a) and (b), it can be observed that the Ship-YOLOv8, which incorporates the W-IoU loss function, exhibits faster reduction in box loss and smoother loss curves, indicating rapid convergence. The last two columns display the precision, recall, and $AP_{0.5}$ curves, with the x-axis representing training epochs and the y-axis representing their respective values. The experiment results demonstrate the superiority of our method.

FIGURE 18 comprehensively reflects the performances of different algorithms on the LDSD. Our proposed algorithm achieved the highest *AP*, sur-passing two small-target detection algorithms, TPH-YOLOv5 and YOLO-Fastestv2, and significantly outperforming a series of well-known algorithms, like Faster-RCNN. In terms of detection speed, our proposed algorithm reached 208 FPS, surpassing all algorithms except YOLOv8. This speed far exceeded the requirements for real-time detection. Considering both precision and speed, our proposed algorithm demonstrated superiority, meeting the demands for real-time detection of long-distance small maritime targets on the sea surface.

### *I. COMPARATIVE EXPERIMENTS ON SeaShips DATASET*

In order to comprehensively validate the applicability of our method, experiments were conducted on the public dataset SeaShips, which comprises 7000 images. This dataset encompasses various types of vessels, such as ore carriers (OC), bulk cargo carriers (BCC), general cargo ships (GCS),


**FIGURE 18.** Comparative analysis of comprehensive performances on LDSD.

container ships (CS), fishing boats (FB), and passenger ships (PS). The distribution of the dataset is presented in Table 7, wherein "Mixed" denotes instances where different categories of ships occlude each other within the images.

**TABLE 7.** Distribution of categories in the SeaShips dataset.

| Ship category | Images | Percentage |
|---|---|---|
| OC | 1141 | 0.1630 |
| GCS | 1188 | 0.1697 |
| BCC | 1129 | 0.1613 |
| CS | 814 | 0.1163 |
| FB | 1258 | 0.1797 |
| PS | 705 | 0.1007 |
| Mixed type | 765 | 0.1093 |

**TABLE 8.** Comparative experiment on the SeaShips dataset.

| Method | $AP_{0.5}$ | *AP%* for Each Category | | | | | |
|---|---|---|---|---|---|---|---|
| | (%) | OC | GCS | BCC | CS | FB | PS |
| YOLOv8 | 98.6 | 98.3 | 99.5 | 99.0 | 98.8 | 97.8 | 98.0 |
| Ship-YOLOv8 | 99.3 | 99.2 | 99.5 | 99.2 | 99.5 | 98.9 | 99.3 |

(a)

| Method | FPS (Image/Seconds) | Inference Time (Milliseconds/Image) |
|---|---|---|
| YOLOv8 | 769 | 1.3ms |
| Ship-YOLOv8 | 250 | 4.0ms |

(b)

Under consistent training conditions and parameters, 130 training epochs were conducted using YOLOv8 and

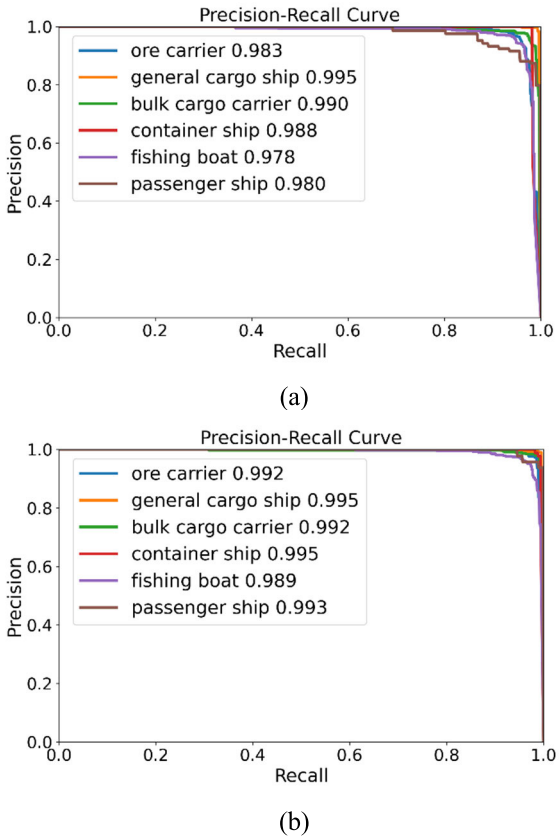


**FIGURE 19.** Comparison of PR curves for the algorithms in SeaShips. (a) PR curve for YOLOv8. (b) PR curve for Ship-YOLOv8.

Ship-YOLOv8. The results are shown in Table 8. The Ship-YOLOv8 continues to demonstrate excellent performance on the SeaShips dataset, achieving an average $AP_{0.5}$ of 99.3%. Moreover, it surpasses the $AP_{0.5}$ of the original YOLOv8

in each detection category. During single-image inference, the model achieves a processing time of 4.0 milliseconds, resulting in a frame rate of 250 FPS, meeting the requirements for real-time detection.

The PR curves for the two algorithms are as illustrated in FIGURE 19. Compared to the original YOLOv8, the Ship-YOLOv8 performs better for various categories of ship targets. Moreover, the curves of Ship-YOLOv8 are smoother, indicating the effectiveness of the proposed algorithm.

## V. CONCLUSION

This study explored methods to tackle the significant challenge of detecting long-distance ships at sea. To address this issue, we developed a novel algorithm, namely Ship-YOLOv8. The algorithm was applied to our self-made LDSD, and we conducted experiments to verify an $AP_{0.5}$ of 91.8%, representing a 21.2-point improvement over YOLOv8. Our proposed algorithm utilizes various techniques to improve the detection accuracy for long-distance ships at sea. First, we created the LDSD, a dataset for sea surface long-distance ship detection. Second, we added the combination module C-BoTNet, with a CNN and a transformer, to the backbone of YOLOv8, which could enhance the extraction of local information and global information at the same time. Then, we used the VoVGSCSP structure to optimize the extraction when performing feature fusion in the neck, which reduced the network's complexity and enhanced the recognition ability for small targets. We added a new small-target detection layer to the backbone, which fuses large feature maps, thereby enhancing the recognition accuracy for small targets and bolstering the capability for long-distance detection. Finally, due to the unique characteristics of the LDSD, W-IoU was incorporated into ship-YOLOv8. This strategic integration serves to effectively alleviate the challenges posed by varying quality levels of long-distance ship samples, thereby enhancing the algorithm's generalization capability.

Experimental results show that Ship-YOLOv8 can achieve higher detection accuracy using the same dataset. Our method demonstrates robustness in scenarios involving object blur and environmental confusion, providing effective sup-port for the intelligent navigation of maritime ships. Additionally, comparative experiments on the public dataset SeaShips show that Ship-YOLOv8 outperforms the original YOLOv8 in multiple metrics. This comprehensive demonstration showcases the good performance of our proposed method across different environments, laying the foundation for obstacle avoidance and autonomous navigation in long-distance ships. Nevertheless, there is still potential for enhancement in our algorithm, as it may erroneously classify birds in the sky or other objects on the sea surface as ships. Addressing this issue constitutes our next research endeavor.

In our forthcoming research endeavors, our primary focal points will encompass three key areas: refining the detection algorithm to further boost target identification accuracy and minimize false detections; enriching the LDSD with techniques such as data augmentation to strengthen the algorithm's capability in intricate scenarios involving small targets; and deploying our algorithm on embedded devices for real-world testing, particularly on custom-built uncrewed ships for detection and obstacle avoidance experiments. These efforts aim to significantly advance the field of object detection in maritime environments, contributing to safer and more efficient maritime navigation.

## REFERENCES

[1] Q. Li, L. Mou, Q. Liu, Y. Wang, and X. X. Zhu, "HSF-net: Multiscale deep feature embedding for ship detection in optical remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 7147–7161, Dec. 2018, doi: 10.1109/TGRS.2018.2848901.

[2] M. Cafaro, I. Epicoco, M. Pulimeno, and E. Sansebastiano, "Toward enhanced support for ship sailing," *IEEE Access*, vol. 11, pp. 87047–87061, 2023, doi: 10.1109/ACCESS.2023.3303808.

[3] C. Dong, J. Liu, and F. Xu, "Ship detection in optical remote sensing images based on saliency and a rotation-invariant descriptor," *Remote Sens.*, vol. 10, no. 3, p. 400, Mar. 2018, doi: 10.3390/rs10030400.

[4] J. Hu, X. Zhi, W. Zhang, L. Ren, and L. Bruzzone, "Salient ship detection via background prior and foreground constraint in remote sensing images," *Remote Sens.*, vol. 12, no. 20, p. 3370, Oct. 2020, doi: 10.3390/rs12203370.

[5] F. Xu, J. Liu, C. Dong, and X. Wang, "Ship detection in optical remote sensing images based on wavelet transform and multi-level false alarm identification," *Remote Sens.*, vol. 9, no. 10, p. 985, Sep. 2017, doi: 10.3390/rs9100985.

[6] F. Ji, D. Ming, B. Zeng, J. Yu, Y. Qing, T. Du, and X. Zhang, "Aircraft detection in high spatial resolution remote sensing images combining multi-angle features driven and majority voting CNN," *Remote Sens.*, vol. 13, no. 11, p. 2207, Jun. 2021, doi: 10.3390/rs13112207.

[7] Z. Tan, Z. Zhang, T. Xing, X. Huang, J. Gong, and J. Ma, "Exploit direction information for remote ship detection," *Remote Sens.*, vol. 13, no. 11, p. 2155, May 2021.

[8] Z. Zou and Z. Shi, "Ship detection in spaceborne optical image with SVD networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 5832–5845, Oct. 2016, doi: 10.1109/TGRS.2016.2572736.

[9] Z. Shi, X. Yu, Z. Jiang, and B. Li, "Ship detection in high-resolution optical imagery based on anomaly detector and local shape feature," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4511–4523, Aug. 2014, doi: 10.1109/TGRS.2013.2282355.

[10] Y. Zhang, Q.-Z. Li, and F.-N. Zang, "Ship detection for visual maritime surveillance from non-stationary platforms," *Ocean Eng.*, vol. 141, pp. 53–63, Sep. 2017.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[12] X. Qian, X. Cheng, G. Cheng, X. Yao, and L. Jiang, "Two-stream encoder GAN with progressive training for co-saliency detection," *IEEE Signal Process. Lett.*, vol. 28, pp. 180–184, 2021, doi: 10.1109/LSP.2021.3049997.

[13] S. Lin, M. Zhang, X. Cheng, L. Wang, M. P. Xu, and H. Wang, "Hyperspectral anomaly detection via dual dictionaries construction guided by two-stage complementary decision," *Remote Sens.*, vol. 14, no. 8, p. 1784, Apr. 2022, doi: 10.3390/rs14081784.

[14] S. Bhattacharjee, P. Shanmugam, and S. Das, "A deep-learning-based lightweight model for ship localizations in SAR images," *IEEE Access*, vol. 11, pp. 94415–94427, 2023, doi: 10.1109/ACCESS.2023.3310539.

[15] W. Zhao, M. Syafrudin, and N. L. Fitriyani, "CRAS-YOLO: A novel multi-category vessel detection and classification model based on YOLOv5s algorithm," *IEEE Access*, vol. 11, pp. 11463–11478, 2023, doi: 10.1109/ACCESS.2023.3241630.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[17] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[18] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. 14th Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands. Cham, Switzerland: Springer, Oct. 2016, pp. 21–37.

[21] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[23] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[26] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

[27] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6568–6577.

[28] C. Zhu, F. Chen, Z. Shen, and M. Savvides, "Soft anchor-point object detection," in *Proc. 16th Eur. Conf. Comput. Vision (ECCV)*, Glasgow, U.K. Cham, Switzerland: Springer, Aug. 2020, pp. 91–107.

[29] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.

[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, and J. Uszkoreit, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–22.

[31] T. Wu, B. Li, Y. Luo, Y. Wang, C. Xiao, T. Liu, J. Yang, W. An, and Y. Guo, "MTU-Net: Multilevel TransUNet for space-based infrared tiny ship detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5601015, doi: 10.1109/TGRS.2023.3235002.

[32] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[33] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–16. [Online]. Available: https://openreview.net/pdf?id=gZ9hCDWe6ke

[34] Z. Dai, B. Cai, Y. Lin, and J. Chen, "UP-DETR: Unsupervised pre-training for object detection with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1601–1610.

[35] M. J. Er, Y. Zhang, J. Chen, and W. Gao, "Ship detection with deep learning: A survey," *Artif. Intell. Rev.*, vol. 56, no. 10, pp. 11825–11865, Oct. 2023.

[36] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7029–7038.

[37] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[38] N. Yu, H. Ren, T. Deng, and X. Fan, "Stepwise locating bidirectional pyramid network for object detection in remote sensing imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023, doi: 10.1109/LGRS.2022.3223470.

[39] W. Krüger and Z. Orlov, "Robust layer-based boat detection and multi-target-tracking in maritime environments," in *Proc. Int. WaterSide Secur. Conf.*, Nov. 2010, pp. 1–7.

[40] D. Liang and Y. Liang, "Horizon detection from electro-optical sensors under maritime environment," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 1, pp. 45–53, Jan. 2020.

[41] M.-D. Li, X.-C. Cui, and S.-W. Chen, "Adaptive superpixel-level CFAR detector for SAR inshore dense ship detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2021.3059253.

[42] L. Liu, G. Liu, X. M. Chu, Z. L. Jiang, M. Y. Zhang, and J. Ye, "Ship detection and tracking in nighttime video images based on the method of LSDT," *J. Phys., Conf.*, vol. 1187, no. 4, Apr. 2019, Art. no. 042074.

[43] W. Wang, J. Ren, C. Su, and M. Huang, "Ship detection in multispectral remote sensing images via saliency analysis," *Appl. Ocean Res.*, vol. 106, Jan. 2021, Art. no. 102448.

[44] Z. Chen, D. Chen, Y. Zhang, X. Cheng, M. Zhang, and C. Wu, "Deep learning for autonomous ship-oriented small ship detection," *Saf. Sci.*, vol. 130, Oct. 2020, Art. no. 104812.

[45] X. Li, L. Zhou, H. Wu, B. Yang, W. Zhang, J. Gu, and Y. Gan, "A min-pooling detection method for ship targets in noisy SAR images," *IEEE Access*, vol. 11, pp. 31902–31911, 2023, doi: 10.1109/ACCESS.2023.3262804.

[46] J. Zhou, P. Jiang, A. Zou, X. Chen, and W. Hu, "Ship target detection algorithm based on improved YOLOv5," *J. Mar. Sci. Eng.*, vol. 9, no. 8, p. 908, Aug. 2021. [Online]. Available: https://www.mdpi.com/2077-1312/9/8/908

[47] H. Tang, S. Gao, S. Li, P. Wang, J. Liu, S. Wang, and J. Qian, "A lightweight SAR image ship detection method based on improved convolution and YOLOv7," *Remote Sens.*, vol. 16, no. 3, p. 486, Jan. 2024. [Online]. Available: https://www.mdpi.com/2072-4292/16/3/486

[48] Y. Wang, B. Wang, L. Huo, and Y. Fan, "GT-YOLO: Nearshore infrared ship detection based on infrared images," *J. Mar. Sci. Eng.*, vol. 12, no. 2, p. 213, Jan. 2024. [Online]. Available: https://www.mdpi.com/2077-1312/12/2/213

[49] D. Chen, S. Sun, Z. Lei, H. Shao, and Y. Wang, "Ship target detection algorithm based on improved YOLOv3 for maritime image," *J. Adv. Transp.*, vol. 2021, pp. 1–11, Sep. 2021.

[50] W. Wang, Y. Li, Y. Zhang, P. Han, and S. Liu, "MPANet-YOLOv5: Multi-path aggregation network for complex sea object detection," *J. Hunan Univ.*, vol. 49, no. 10, pp. 69–76, 2022.

[51] F. Zhang and X. Hou, "Multi-site and multi-scale unbalanced ship detection based on CenterNet," *Electronics*, vol. 11, no. 11, p. 1713, May 2022.

[52] Z. Li, Q. Zhang, T. Long, and B. Zhao, "Ship target detection and recognition method on sea surface based on multi-level hybrid network," *J. Beijing Inst. Technol.*, vol. 30, pp. 1–10, Jan. 2021.

[53] J. Zhang, Z. Chen, G. Yan, Y. Wang, and B. Hu, "Faster and lightweight: An improved YOLOv5 object detector for remote sensing images," *Remote Sens.*, vol. 15, no. 20, p. 4974, Oct. 2023.

[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

[55] Z. Shao, W. Wu, Z. Wang, W. Du, and C. Li, "SeaShips: A large-scale precisely annotated dataset for ship detection," *IEEE Trans. Multimedia*, vol. 20, no. 10, pp. 2593–2604, Oct. 2018.

[56] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 1993–2016, Aug. 2017.

[57] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*, Zürich, Switzerland. Cham, Switzerland: Springer, Sep. 2014, pp. 740–755.

[58] H. Li, J. Li, H. Wei, Z. Liu, Z. Zhan, and Q. Ren, "Slim-neck by GSConv: A lightweight-design for real-time detector architectures," *J. Real-Time Image Process.*, vol. 21, no. 3, p. 62, Jun. 2024.

[59] Z. Tong, Y. Chen, Z. Xu, and R. Yu, "Wise-IoU: Bounding box regression loss with dynamic focusing mechanism," 2023, *arXiv:2301.10051*.

[60] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 16514–16524.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[62] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 2778–2788.

**YANFENG GONG** received the Ph.D. degree in instrument science and technology from Chongqing University, Chongqing, China, in 2021. He is currently a Lecturer with the Department of Maritime Technology, Chongqing Jiaotong University, Chongqing. His research interests include image processing, machine learning, deep learning, and intelligent ships.

**ZIHAO CHEN** received the bachelor's degree from Hunan Agricultural University. He is currently pursuing the master's degree with Chongqing Jiaotong University. His current research interests include deep learning and object detection.

**WEN DENG** received the bachelor's degree from Jiangsu University of Technology. He is currently pursuing the master's degree with Chongqing Jiaotong University. His current research interests include deep learning, machine learning, and object recognition.

**JIAWAN TAN** received the Ph.D. degree in traffic information engineering and control from Dalian Maritime University, Liaoning, China, in 2003. He was a Post-Doctoral Researcher with the State Key Laboratory of CAD and CG, Zhejiang University, from 2003 to 2005. He is currently the Associate Dean and an Associate Professor with the College of Navigation, Chongqing Jiaotong University. His research interests include image processing, intelligent navigation, virtual reality, computer simulation, and electronic chart systems.

**YABIN LI** received the Ph.D. degree in cartography and geographical information engineering from the Graduate University of Chinese Academy of Sciences, Beijing, China, in 2008. He is currently the Dean and a Professor with Qingdao Institute of Shipping Development Innovation. His research interests include intelligent navigation of ships, virtual reality, and computer simulation.