IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 February 2024, accepted 29 February 2024, date of publication 8 March 2024, date of current version 14 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3373893

RESEARCH ARTICLE

SHIP-YOLO: A Lightweight Synthetic Aperture Radar Ship Detection Model Based on YOLOv8n Algorithm

YONGHANG LUO, MING LI, GUIHAO WEN, YUNFEI TAN, AND CHAOSHAN SHI
College of Computer and Information Sciences, Chongqing Normal University, Shapingba, Chongqing 401331, China
Corresponding author: Ming Li (55613163@qq.com)

ABSTRACT In response to the challenges posed by small objects, high noise, and complex backgrounds in synthetic aperture radar (SAR) ship detection, we proposed a lightweight model called SHIP-YOLO. In the neck of YOLOv8n, we replaced ordinary convolution (Conv) with a lighter ghost convolution (GhostConv) and introduced reparameterized ghost (RepGhost) bottleneck structure in C2f module. We then introduced Wise-IoU (WIoU) into the algorithm to improve the localization ability of the detection box. Finally, shuffle attention (SA) modules were added to the backbone and neck of YOLOv8n to enhance the perception capability of the target area. The results confirm that, compared with YOLOv8n, the proposed SHIP-YOLO on SAR Ship Detection dataset (SSDD) reduces the parameters and floating-point operations (FLOPs) by 17% and 11%, respectively, and improves the precision, recall, and mean average precision (mAP) (0.5) by 1.7%, 0.1%, and 0.2%, respectively. The proposed model also showed strong generalization ability on another Sar-Ship-Dataset.

INDEX TERMS Computer vision, deep learning, object detection, radar remote sensing, synthetic aperture radar.

I. INTRODUCTION

With the continual growth of maritime traffic and the increasing complexity of ship activities, conventional ship detection methods face limitations owing to the impact of weather and lighting conditions as well as the challenge of effectively locating targets in expansive water regions. Synthetic aperture radar (SAR) technology, which is characterized by its relative insensitivity to weather and lighting conditions, has been widely applied in ship detection. Traditional SAR ship detection algorithms, such as constant false alarm rate (CFAR) [1], Gaussian model-based two-parameter CFAR [2], template matching [3], trial detection [4], and wavelet-based [5] detection methods, primarily rely on manually crafted classifiers. Despite their fast computational speeds, these methods exhibit poor detection performance, and the design process of the detection algorithms is intricate. With the rapid acquisition of extensive SAR data and impetus for practical tasks, traditional detection methods no longer meet the current demands for timeliness and accuracy of SAR ship detection.

Currently, object detection techniques based on deep learning is a popular research topic in the field of computer vision. SAR ship detection typically involves image or video analysis, and object detection techniques can be effectively applied in this domain. Several researchers have provided SAR ship datasets for this field. Li et al. [6] constructed a SAR ship detection dataset (SSDD) based on the PASCAL VOC template. This dataset consists of 1,160 images captured by RadarSat-2, TerraSAR-X, and Sentinel-1 satellites, containing a total of 2,540 ship targets. The images in this dataset have various polarization modes and resolutions with a high scene complexity, which can effectively test the performance of algorithm. Wang et al. [7] proposed a SAR-Ship-Dataset based on multimodal SAR images. Using this dataset, they developed an integrated deep learning processing system for ship detection and classification in complex backgrounds, achieving near-real-time automatic detection and classification of merchant ships without distinguishing between sea

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Santi.

and land. Sun et al. [8] introduced a high-resolution, large-scale SAR ship detection dataset called AIR-SAR Ship, with an image size of approximately $3000px \times 3000px$, and resolutions of 1m and 3m. Lei et al. [9] proposed a high-resolution SAR images dataset (HRSID) composed of images captured by Sentinel-1 and TerraSAR-X satellites. This dataset contains 5,604 image slices with 16,591 ship targets suitable for ship detection and instance segmentation.

Target detection algorithms are primarily divided into two-stage and single-stage detection algorithms. Common two-stage detection algorithms include region convolutional neural network (R-CNN) [10] and faster region convolutional neural network (Faster R-CNN) [11]. These algorithms generate candidate boxes, and then classify and perform bounding box regressions on these candidate boxes. In contrast, single-stage detection algorithms typically have a lower computational complexity and are more suitable for real-time applications. Representative algorithms include you only look once (YOLO) [12], single-shot multibox detector (SSD) [13], and Retina-Net [14]. Zhang et al. [15] proposed a SAR ship detection algorithm that improved Faster R-CNN. On the fusion dataset of GF-3 and Sentinel-1 satellites, the mean average precision (mAP)(0.95) increased by 6.2% compared with Faster R-CNN. Zhang et al. [16] also proposed a SAR ship detection algorithm that improves real-time models for object detection (RTMDet). Experimental comparisons on rotated ship detection dataset (RSDD) revealed that the improved algorithm achieved significant performance enhancement. Zhang and Zhang [17] proposed a method for detecting high-speed SAR ship based on a grid convolutional neural network (G-CNN). The purpose of improving the detection speed was achieved through grid processing of the image and depth-separable convolution. Zhang et al. [18] proposed a new high-speed SAR ship detection method using a deep separable convolutional neural network (DS-CNN). This method significantly improved detection speed while maintaining high accuracy through a lightweight network architecture integrating multi-scale detetction, serial, and anchor frame mechanisms. Xu et al. [19] proposed a method called group-wise feature enhancement-and-fusion network with dual-polarization feature enrichment (GWFEF-Net) to enrich the feature library of SAR ship detection network through dual polarization characteristics and realized more accurate detection of ship targets in SAR images by using intra-group feature enhancement, feature fusion, mixed pool channel attention, and other technologies. After extensive experiments on Sentinel-1 dual-polarization SAR ship dataset, the mAP of GWFEF-Net was 94.18%, which is 2.51% higher than that of the second-best method. Xu et al. [20] proposed a method called shadow-background-noise 3D spatial decomposition model (SBN-3D-SD), which enhances moving target shadows in SAR images through 3D spatial decomposition, thereby improving the accuracy of object detection and tracking. Huang et al. [21] proposed a lightweight SAR ship detection algorithm based on YOLOv5, using channel pruning and knowledge distillation. Experimental results on the restructured large-scale multi-class SAR image target detection dataset (MSAR) and SSDD multi-class target datasets showed improved target detection accuracy while maintaining a lightweight model volume of only 3.73M. He et al. [22] introduced an improved detection algorithm based on YOLOv5, using a modified bi-directional feature pyramid network (BiFPN). On HRSID, the recall and mAP increased by 2% and 2.7%, respectively, compared with the original YOLOv5 algorithm. Zhang et al. [23] based their work on YOLOv7, incorporating shuffle attention (SA) module and introducing dynamic snake convolution (DSConv). They conducted experiments on HRSID, and the results showed a 16.7% improvement in the mAP in detection tasks, with a 62.55% reduction in the model volume compared with the original YOLOv7 algorithm. Ren et al. [24] proposed the YOLO-Lite model for SAR ship detection. By designing effective lightweight feature enhancement networks, position information capture modules, and multi-scale feature fusion networks, the YOLO-Lite model can significantly reduce computational complexity while ensuring detection accuracy and providing an effective solution for real-time ship detection. Guo et al. [25] proposed a lightweight SAR ship detection model called LMSD-YOLO. By introducing an activation function module (DBA), a mobile net with a stem block (S-MobileNet) backbone network, depth-wise adaptively spatial feature fusion module (DSASFF), and SCYLLA-IoU (SIoU), better adaptability, and smaller model volume in multi-scale object detection are realized. Tang et al. [26] proposed a SAR ship detection model based on YOLOv7, which utilizes a multi-scale receptive field convolution block (AMMRF) to fully utilize the positional information of feature maps and effectively capture the relationship between feature map channels, enabling the network to better learn the relationship between ships and the background. On HRSID and LS-SSDD-v1.0, compared with YOLOv7, the mAP (0.5) was improved by 2.6% and 3.9%, respectively.

Currently, efforts in deep learning algorithms are directed towards achieving smaller and faster models. The pursuit of lightweight solutions without compromising the algorithmic accuracy has become a crucial research direction. Cholet [27] believes that channel and spatial correlation should be treated separately. Depth-wise convolution (DWConv) is performed by splitting the convolution operation into two steps: depth convolution and point-by-point convolution. Han et al. [28] harnessed the redundancy characteristics of feature maps to introduce ghost convolution (GhostConv), which is a lighter alternative achieved through a series of cost-effective linear transformations. Chen et al. [29] presented a reparameterized ghost (RepGhost), which is an innovative, hardware-efficient solution. Implicit feature reuse is achieved through reparameterization rather than cascade operators, thereby reducing parameters and delays. Tong et al. [30] proposed Wise-IoU (WIoU), which establishes a dynamic focusing mechanism

(FM) by estimating the outlierness of anchor boxes. This approach utilizes abnormality instead of IoU to assess the quality of anchor boxes and implements a prudent gradient gain allocation strategy to mitigate the competitiveness of low-quality anchor boxes and detrimental gradients produced by low-quality examples. In computer vision research, spatial and channel attention mechanisms are widely employed to capture pixel-level pairwise relationships and channel dependencies. Although combining these mechanisms may enhance performance, it inevitably introduces a computational overhead. Zhang and Zhang [31] addressed this challenge by introducing SA module, an effective solution that seamlessly integrates both types of attention mechanisms using shuffle units. Hu et al. [32] proposed a squeeze-and-excitation (SE) module that used global information to learn the weight of each channel. By compressing the dimensions of the feature graph and using multilayer perceptrons to recalibrate the feature response of the channel, the learning ability of the model for the correlation between features is enhanced, and then the representation ability and performance of the convolutional neural network are improved.

In the domain of SAR ship detection, many studies have focused solely on achieving high accuracy or lightweight algorithms without striking a balanced compromise between the two. Consequently, we propose a novel model called SHIP-YOLO, which aims to maintain high detection accuracy while achieving algorithmic lighting. The primary contributions of this study are summarized as follows:

(1) We integrated the RepGhost bottleneck structure into C2f module of YOLOv8 to create an updated and lighter module called C2f_RepGhost.

(2) Notable modifications include replacing Conv in the neck of YOLOv8n with GhostConv, replacing the C2f module in the neck of YOLOv8n with the proposed C2f_RepGhost module, replacing the Complete-IoU (CIoU) [33] with WIoU, and incorporating SA modules into the backbone and neck of the algorithm.

(3) We presented and performed a comprehensive analysis of the proposed SHIP-YOLO on SSDD and SAR-Ship-Dataset.

II. INTRODUCTION TO YOLOV8 ALGORITHM

YOLO is a one-stage real-time object detection model. In contrast to two-stage detection model, YOLO processes images by partitioning them into multiple grids, where each grid makes predictions for multiple bounding boxes and their confidences, subsequently determining the class of objects within the predictions. This unique approach allows YOLO to deliver rapid detection capabilities; the initial version achieved impressive 45 frames per second on a Titan X GPU. Over time, YOLO has evolved through various iterations, with YOLOv2 [34] to YOLOv8 witnessing enhancements in speed, accuracy, and model size.

Figure 1 depicts the architecture of YOLOv8, comprising three integral components: backbone, neck, and head. The backbone, akin to YOLOv5 [35], is rooted in the cross stage partial dark network-53 (CSPDarkNet-53), encompassing CBS, C2f, and spatial pyramid pooling-fast (SPPF) modules. However, YOLOv8 diverges from YOLOv5 by adopting the C2f module over the C3 module, leveraging insights from ELAN module of YOLOv7 [36]. This refinement bolsters feature fusion capabilities, thereby accelerating the inference speed. YOLOv8 integrates the SPPF module from YOLOv5, harmonizing the local and global features via spatial pyramid pooling layers. The neck of YOLOv8 echoes the path aggregation network-feature pyramid network (PAN-FPN) structure of YOLOv5 but refines it by removing the convolutional structure in the upsampling stage of PAN-FPN and replacing C3 module with the C2f module in YOLOv5. As for the head, YOLOv8 aligns with YOLOv6 [37] and YOLOX [38], employing a decoupled head, in contrast to the coupled head utilized in YOLOv3 [39], YOLOv4 [40], YOLOv5, and YOLOv7. This decoupled head strategically segregates the object position and class information, facilitating independent learning through distinct network branches, culminating in synergized fusion. This innovative approach efficiently reduced the parameter volume and computational complexity of the algorithm.

III. ALGORITHM IMPROVEMENT

In this study, we proposed the SHIP-YOLO SAR ship detection model based on YOLOv8n. Notably, in the neck part, Conv is replaced by GhostConv, and the RepGhost bottleneck structure is integrated into the C2f module, thereby contributing to lightweight algorithmic improvement. The proposed SHIP-YOLO further incorporates SA module into both SPFF module of the backbone of the algorithm and the second upsample module in the neck to capture more diverse feature information. Additionally, the loss function of the algorithm was replaced with WIoU to enhance the precision of the detection boxes and augment the recall of the detected targets. The overall network structure of SHIP-YOLO is shown in Figure 2.

A. LIGHTWEIGHT MODULES

1) GhostConv

Given the ongoing evolution of convolutional neural networks and escalating demand for deploying models on embedded devices with constrained memory and computational resources, the pursuit of more efficient and lightweight neural networks has emerged as a pivotal trend. This study introduces GhostConv, a lightweight alternative to a certain Conv within the YOLOv8 algorithm, aimed at curtailing both the computational and parameter requirements. The disparities between Conv and GhostConv are shown in Figure 3.

GhostConv involves cost-effective linear operations based on a minimal number of traditional convolutional operations. Conv is divided into two components: the first part executes a standard convolution while meticulously controlling its quantity and the second part employs the inherent feature maps generated by Conv to conduct a series of straightforward

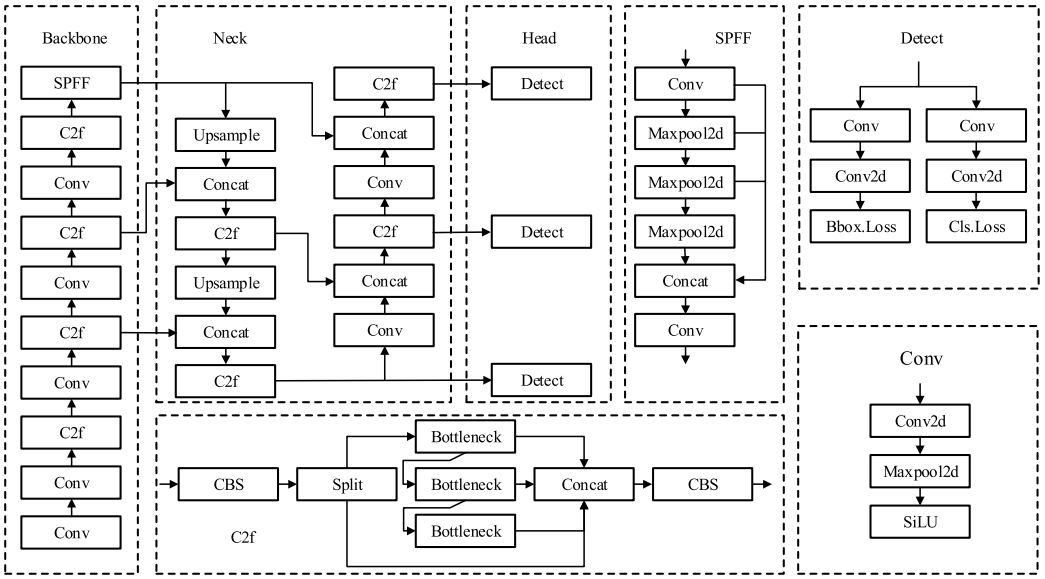

FIGURE 1. The YOLOv8n network structure.

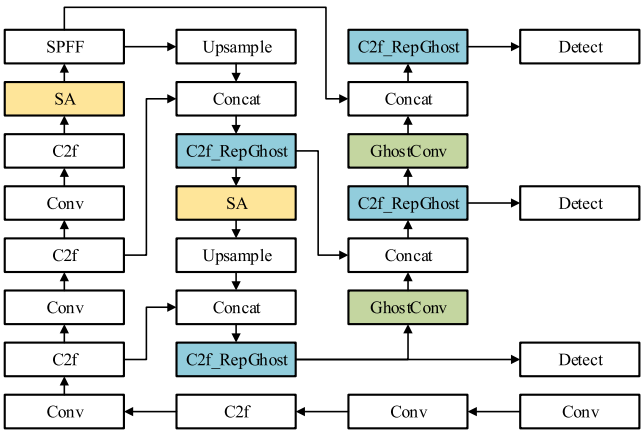

FIGURE 2. The proposed SHIP-YOLO network structure.

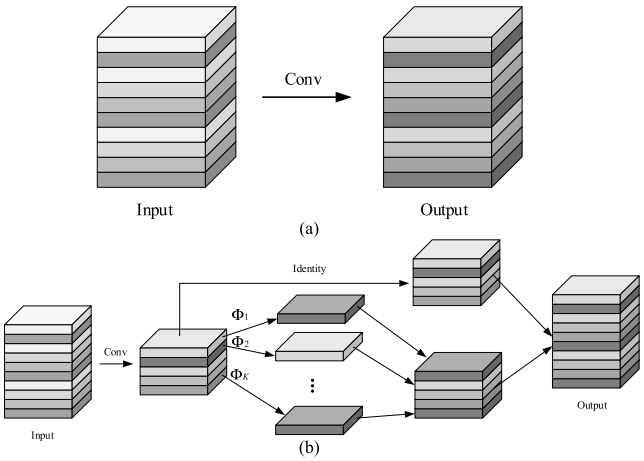

FIGURE 3. The process of Conv (a) and GhostConv (b).

linear operations, yielding additional feature maps. These two sets of feature maps were then concatenated to form a new output. This approach significantly reduces the parameters and computational workload while maintaining the model performance, paving the way for a more efficient and lightweight deployment of neural networks. Let "h", "w", and "c" denote the height, width, and channel number of input features, respectively. The height and width of output features are represented by "h" and "w", the number of convolutional kernels is denoted as "n", the kernel size is "k", the linear transformation kernel size is "d ", and the transformation count is "s". "r_s" and "r_c" are the ratios of floating-point operations (FLOPs) and parameters between Conv and GhostConv, where the formulas are as follows:

$$r_s = \frac{h \times w \times c \times H \times W \times n}{\frac{n}{s} \times H \times W \times k^2 \times c + (s-1) \times \frac{n}{s} \times H \times W \times d^2} = \frac{c \times k^2}{\frac{1}{s} \times c \times k^2 + \frac{(s-1)}{s} \times d^2} \approx s \tag{1}$$

$$r_c = \frac{n \times c \times k^2}{\frac{n}{s} \times c \times k^2 + (s-1) \times \frac{n}{s} \times d^2} \approx \frac{s \times c}{s + c - 1} \approx s \tag{2}$$

Equations (1) and (2) reveal that the ratio of FLOPs to parameters is influenced by the transformation count "s". Essentially, the more feature maps that are generated, the more effective the model acceleration. Therefore, introducing GhostConv into the model proved to be an efficient strategy to significantly reduce the FLOPs and parameters, ultimately boosting the operating speed and efficiency of the model.

2) C2f_RepGhost

Despite the enhanced accuracy of the YOLOv8 algorithm compared with its predecessors, the model's complexity and

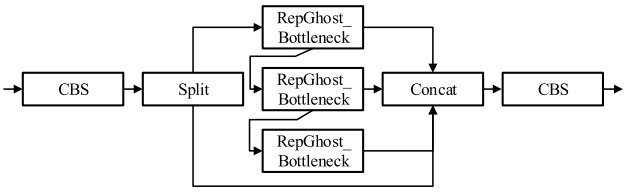

FIGURE 4. The proposed C2f_RepGhost module structure.

substantial parameter count pose challenges for practical deployment, particularly for devices with limited performance, such as edge-terminal devices. To address this, the RepGhost bottleneck structure replaces the original bottleneck structure in the C2f module, resulting in a reduction in parameters and overall model size. This modification overcomes the deployment challenges associated with the YOLOv8n network's demanding model parameters. A structural illustration of this process is shown in Figure 4.

The reparameterization process of the RepGhost module is illustrated in Figure 5. This involves generating an equivalent convolution layer on the batch normalization (BN) branch and subsequently fusing the convolution layer with the BN layer. During training, the RepGhost module performs deep convolution on the input features, expanding the feature dimension and then BN to increase nonlinearity during training. This step can be fused during the inference. The feature maps generated by the two branches are then added while maintaining the same number of channels. Finally, a ReLU activation function is added to comply with the reparameterization rules for swift inference.

The structure of the RepGhost module during inference is straightforward, with the feature fusion process occurring in the weight space rather than in the feature space. In addition, by merging the parameters of the two branches, the RepGhost module achieves a structure optimized for rapid inference containing only regular convolution layers and ReLU activation functions, resulting in high hardware efficiency.

The RepGhost bottleneck structure is illustrated in Figure 6. It utilizes a 1×1 convolution layer and ReLU activation function to reduce the channel number of the input by half. After the first RepGhost module, maintaining the channel number, it passes through an SE attention layer and a 1×1 convolution to enhance the sensitivity of the model to the channel features, aligning the output channel number with the input. Finally, the feature undergoes a scale addition operation with the input feature map after RepGhost module, and the result is outputted. The RepGhost bottleneck includes only two branches during inference: saving memory resources and improving the inference speed.

B. LOSS FUNCTION

The YOLOv8 loss function primarily comprises two components: classification and regression losses. For classification, binary cross-entropy (BCE) loss is utilized, and for regression, distribution focal loss (DFL) [41] and CIoU are applied.

DFL aims to optimize the shape of the probability distribution $P(x)$ by explicitly encouraging high probabilities of values that are close to the target y. This is achieved by enlarging the probabilities of the two nearest values to y, denoted as y_i and y_{i+1}, where $y_i \leq y \leq y_{i+1}$. This encourages the network to focus on learning the probabilities of the values around the continuous locations of the target bounding boxes, as defined in Equation (3):

$$DFL\left(S_i, S_{i+1}\right) = -\left(y_{i+1} - y\right) \log\left(S_i\right) - \left(y - y_i\right) \log\left(S_i + 1\right) \tag{3}$$

where S_i and $S_{i=1}$ are the probabilities associated with the two nearest values to the target y.

The CIoU is defined as in Equation (4):

$$R_{CIoU} = \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \alpha v \tag{4}$$

b, b^{gt} represent the center points of the predicted and ground-truth bounding boxes, respectively. ρ denotes the Euclidean distance between the two points and c is the diagonal distance of the minimum enclosing rectangle that contains both boxes. α is the weight coefficient.

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{5}$$

IoU represents the intersection over union between the predicted box and ground-truth box. v measures the similarity in aspect ratios and is defined as in Equation (6):

$$v = \frac{4}{\pi^2}\left(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}\right)^2 \tag{6}$$

where w, h, w^{gt}, h^{gt} represent the width and height of the annotated and ground truth boxes, respectively. The complete definition of the CIoU is as follows:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \alpha v \tag{7}$$

However, the CIoU has some limitations. The aspect ratio v describes a relative value, leading to some ambiguity, and it does not consider the balance of difficulty in the samples. Owing to the presence of low-quality examples in the training data, geometric factors (distance and aspect ratio) intensify the penalty for these examples, and static FM cannot distinguish between high and low-quality annotated boxes, resulting in weaker model generalization performance [42].

Because SAR ship detection training data include a large number of images with low imaging quality, geometric measurements such as distance and aspect ratio exacerbate the penalty for low-quality images, thereby reducing the model's generalization performance.

Intervene in training without excessive interference and weaken the penalty for geometric measurements when anchor boxes overlap well with target boxes, thereby improving the generalization ability of the model, by using WIoU_v3 with a dynamic non-monotonic FM gradient gain allocation strategy. In the early stages of training, high-quality anchor boxes

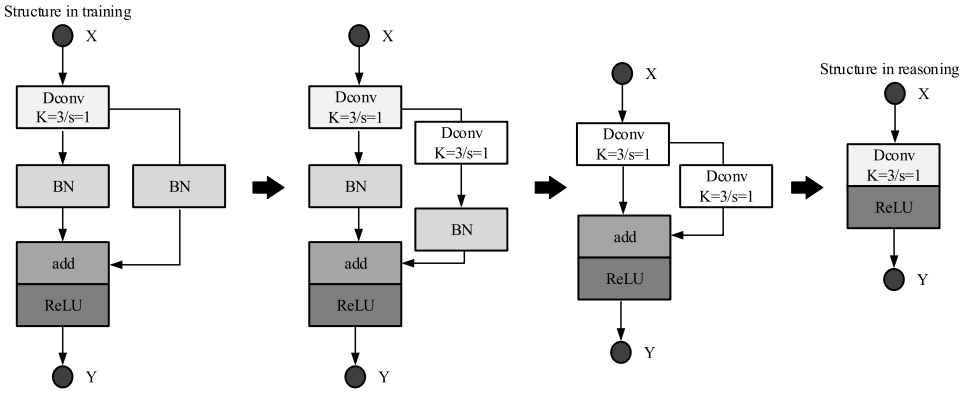

FIGURE 5. The reparameterization process of RepGhost module.

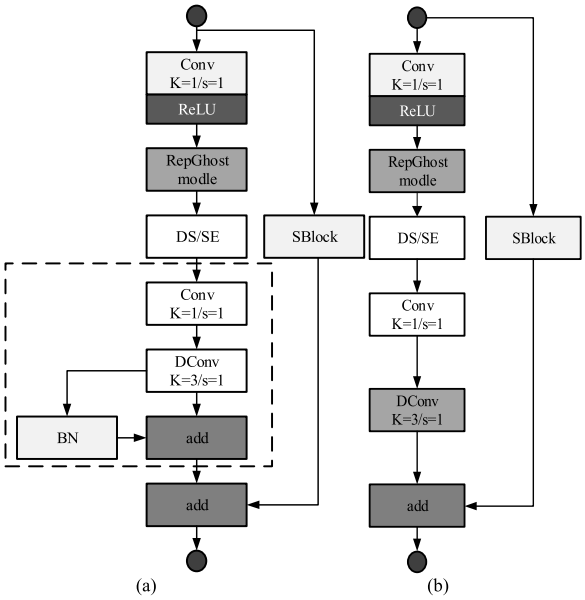

FIGURE 6. The structure of RepGhost bottleneck in training (a) and in reasoning (b).

were preserved to reduce the penalty for geometric factors and enhance the generalization ability of the model. In the later stages of training, WIoU_v3 allocates smaller gradient gains to low-quality anchor boxes to reduce harmful gradients and improve the localization performance of the model [42].

WIoU is defined as in Equation (8):

$$L_{WIoU\_v1} = R_{WIoU} \cdot L_{IoU} \tag{8}$$

$R_{WIoU} \in [1, e)$ significantly enlarges L_{IoU} of an ordinary mass anchor frame; $L_{IoU} \in [0, 1]$ significantly reduces the R_{WIoU} of high-quality anchor frames, and significantly reduces their focus on the distance between the center point when the anchor frame and target frame coincide well. To prevent R_{WIoU} from generating gradient impeding convergence, W_g and H_g are separated from the calculation diagram ($*$ indicates this operation). The R_{WIoU} is defined as in Equation (9):

$$R_{WIoU} = \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{\left(W_g^2 + H_g^2\right)}\right) \tag{9}$$

W_g and H_g are the width and height of the minimum bounding box, respectively. R_{WIoU} represents the normalized distance between the center points of the predicted and ground-truth bounding boxes.

To improve the model detection performance further, an outlier was defined to describe the quality of the anchor frame, as shown in Equation (10):

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

L_{IoU}^* denotes the monotonic focusing coefficient. During model training, the gradient gain L_{IoU}^* decreases with a decrease in L_{IoU}, where $\overline{L_{IoU}}$ is the moving average of the momentum m.

A small degree of outlier means that the anchor frame has high quality; therefore, a small gradient gain is allocated to it to make the boundary frame return to focus on the anchor frame with ordinary quality. Allocating a smaller gradient gain to the anchor frame with larger outliers effectively prevents low-quality image data from generating larger harmful gradients, which affects detection quality. Using β, a non-monotone focusing coefficient γ is constructed and multiplied by the WIoU_v1 to obtain WIoU_v3, the definition of which is shown in Equation (11):

$$L_{WIoU\_v3} = \gamma L_{WIoU\_v1}, \gamma = \frac{\beta}{\delta \alpha^{\beta - \delta}} \tag{11}$$

When $\beta = \delta$, $\gamma = 1$. When the degree of outlier of the anchor frame satisfies the fixed value of β, the anchor frame obtains the highest gradient gain. Because $\overline{L_{IoU}}$ is dynamic, the quality division standard of the anchor frame is also dynamic, which enables WIoU_v3 to create a gradient gain allocation strategy that best matches the current situation in the training process compared with WIoU_v1 and WIoU_V2.

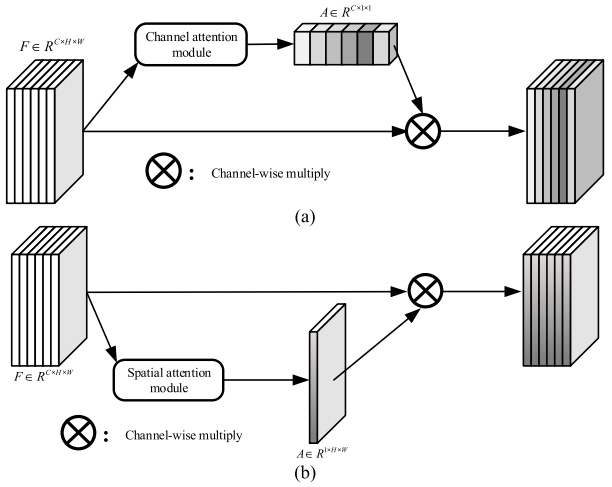

FIGURE 7. The process of channel attention (a) and spatial attention (b).

C. ATTENTION MECHANISM

The attention mechanism is instrumental in enabling the network to focus precisely on information relevant to the input, thus proving to be a vital component in enhancing the performance of deep neural networks. Attention mechanisms can be broadly categorized into two types: channel and spatial, as shown in Figure 7. While the channel attention mechanism emphasizes content, spatial attention focuses on regions relevant to the task. The combination of both can lead to improved performance, but inevitably results in an increase in the number of parameters and computational complexity. Shuffle Attention effectively addressed this challenge by adopting the channel shuffle concept.

First, the input features are divided into multiple sub-feature groups along the channel dimension to learn better feature representations and alleviate the difficulty of convergence in deep network training. Next, for each sub-feature, a shuffle unit is applied to simultaneously construct the channel and spatial attention, suppress potential noise and emphasize regions with correct semantic features, thereby generating correlation coefficients for the sub-features. Finally, all sub-features processed by the SA module are concatenated along the dimension, and the Channel Shuffle operation [43], [44] facilitates information exchange between different sub-feature groups, as illustrated in Figure 8.

For an input feature map$X$, with dimensions $c \times h \times w$, SA initially divides the feature map X into g groups along the channel dimension, resulting in grouped feature maps $X = \left[X_1, \cdots, X_k, \cdots, X_g\right], K \in [1, g]$. Where the channel number, height, and width of any sub-feature group are denoted as $\frac{c}{g}$, h, and w respectively. During training, each sub-feature X_k gradually acquires specific semantic information and obtains the corresponding weight coefficients through the SA module. Each sub-feature group obtained after grouping generates two branches, X_{k1} and X_{k2} at the beginning of the attention module. The channel number, height, and width of each branch are denoted as $\frac{c}{2g}$, h, and w respectively. They generated channel attention maps and spatial attention maps by utilizing channel and internal spatial relationships, respectively. The channel attention map emphasizes the semantic information extracted from the feature map to enhance useful feature channels for the current task and suppress less useful feature channels, while the spatial attention map emphasizes the spatial information extracted from the feature map, allowing the model to focus more on meaningful information.

For branch X_{k1}, the channel attention mechanism was employed. Global Average Pooling (GAP) [45] was applied to produce channel-level data with a global receptive field, as shown in Equation (12).

$$S = Fgp\left(X_{k1}\right) = \frac{1}{h \times w} \sum_{i=1}^{h} \sum_{j=1}^{w} X_{k1}(i, j) \tag{12}$$

Then, an Fc linear transformation and activation are applied to s to make the channel-wise global data s more accurate. The generated weight mapping X'_{k1} for each channel is shown in Equation (13):

$$X'_{k1} = \sigma'\left(Fc\left(s\right)\right) \times X_{k1} = \sigma'\left(W_1 \times s + b_1\right) \times X_{k1} \tag{13}$$

where W_1 and b_1 represent the weight vector and bias vector of the linear transformation layer, respectively, and $\sigma'(\cdot)$ denotes the activation using the sigmoid function for each element of the input vector.

The spatial attention module places a greater emphasis on the location information of the target. Therefore, channel grouping normalization (Group Norm) [46] was applied to branch X'_{k2} to obtain spatial-level data $GN(X_{k2})$, and the same linear transformation and sigmoid function operations were used to obtain the final output, as shown in Equation (14):

$$X'_{k2} = \sigma'\left(W_2 \times GN\left(X_{k2}\right) + b_2\right) \times X_{k2} \tag{14}$$

where W_2 and b_2 represent the weight and bias vectors of the linear transformation layer, respectively.

Finally, branches X_{k1} and X_{k2} were superimposed to match the number of channels with the input. The Channel Shuffle operation is then applied to allow the flow of information across groups, effectively merging all sub-features while retaining inter-group information flow.

IV. EXPERIMENTAL DESIGN AND RESULT ANALYSIS

A. DATASET

SSDD and SAR-Ship-Dataset were used for both training and testing. SSDD consists of 1160 images containing 2456 ship instances. Given the limited dataset size, a random split may disrupt the consistency in the distribution between the training and test sets, leading to inconsistent training outcomes. To address this, the study followed SSDD's official recommendation, strictly defining images with the last digit of the file number being one or nine as the test set, with the rest used for training and validation. The split was conducted with a 7:1:2 ratio of the training, validation, and testing. The SAR-Ship-Dataset comprises 43819 images, spanning scenes such as ports, nearshore areas, islands, and the open sea,

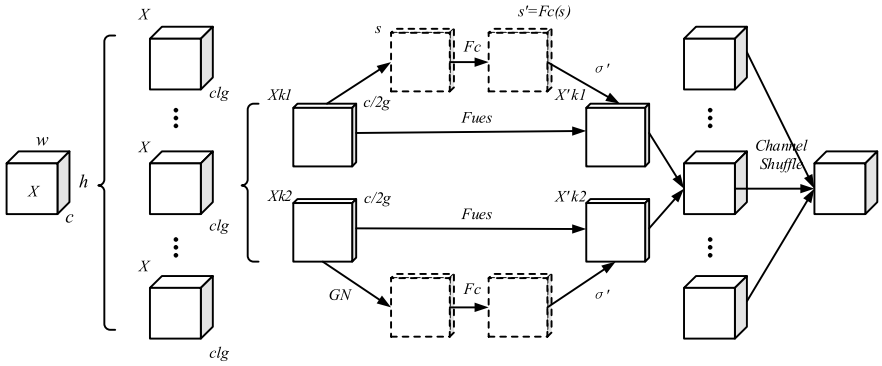

FIGURE 8. The detailed SA Module structure.

TABLE 1. The experimental environment used in this study.

Name	Configuration
Processor	15 vCPU AMD EPYC 7543 32-Core Processor
GPU	RTX 3090
RAM	80GB
Operating system	Ubuntu20.04
Platform	Pytorch2.0.0, Python3.8, Cuda11.8

TABLE 2. The experimental parameter settings used in this study.

batchsize	epch	lr0	lrf	momentum	weight_decay	imgsz	optimizer
64	300	0.01	0.01	0.937	0.0005	640	SGD

featuring various common ship types such as cruise ships, bulk carriers, large container ships, and fishing boats. The dataset was split into training, validation, and test sets in a 7:2:1 ratio. Considering the experimental equipment and time constraints, the smaller SSDD dataset was used for model training and testing, whereas the larger SAR-Ship-Dataset was used to assess the generalization of the algorithm.

B. EXPERIMENTAL ENVIRONMENT AND PARAMETER SETTINGS

Details of the experimental environment and parameter settings are presented in Tables 1 and 2, respectively.

C. EVALUATION METRICS

The performance evaluation metrics chosen for this experiment included precision, recall, mAP, and FLOPs. The formulations for each metric are as follows:

$$precision = \frac{T_P}{T_P + F_P} \tag{15}$$

$$recall = \frac{T_P}{T_P + F_N} \tag{16}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{n}\int_0^1 Precision(Recall)d(Recall) \tag{17}$$

where T_P is the number of true positive samples, F_P is the number of false positive samples, F_N is the number of false negative samples, and N is the number of detected categories.

D. IMPACT OF LIGHTWEIGHT MODULES ON ALGORITHM PERFORMANCE

To assess the influence of introducing lightweight modules at different positions within YOLOv8n on the algorithm performance, a comparative experiment was conducted, and the results are presented in Table 3. The "Backbone" scenario involves the replacement of all Conv or C2f modules in the YOLOv8n backbone part with lightweight GhostConv or C2f_RepGhost modules. For "Neck", all Conv or C2f modules in the YOLOv8n neck part are substituted with lightweight GhostConv or C2f_RepGhost modules. "All" refers to replacing all Conv or C2f modules throughout YOLOv8n with lightweight GhostConv or C2f_RepGhost modules. The findings indicate that with a reduction of 0.1M in parameters and 0.1G in FLOPs, replacing Conv in the neck part of YOLOv8 with GhostConv resulted in an improvement of 1.1% in precision and 0.2% in mAP50. Similarly, replacing the C2f modules in the YOLOv8n neck part with C2f_RepGhost modules, reducing parameters by 0.4M and FLOPs by 0.8G, yields a 0.3% increase in precision, a 0.5% increase in mAP50, and a 0.2% increase in recall. By simultaneously replacing Conv and C2f modules in the neck part of YOLOv8 with GhostConv and C2f_RepGhost modules, and further reducing parameters by 0.5M and FLOPs by 0.9G, the precision improved by 0.1%, mAP50 increased by 0.7%, and recall improved by 0.1%. These results highlight the effectiveness of the proposed lightweight optimization algorithm.

E. IMPACT OF IOU ON ALGORITHM PERFORMANCE

Using the YOLOv8n algorithm as the baseline model, where Conv and C2f modules in the neck part were replaced with GhostConv and C2f_RepGhost modules, respectively, different IoUs were evaluated for their impact on the algorithm performance. Table 4 presents the comparative models. The experimental results revealed that WIoU_v3 contributed to

TABLE 3. Performance Comparison of Lightweight Modules at Different Positions in the Algorithm.

Module	Algorithms	precision/%	recall/%	mAP50/%	parameters/M	FLOPs/G
Baseline	YOLOv8n	95.4	94.4	96.9	3	8.1
GhostConv	BackBone	95.6	93.8	97.1	2.8	7.6
	Neck	*96.5*	*94.4*	*97.1*	*2.9*	*8.0*
	All	93.4	94.9	96.9	2.7	7.5
C2f_RepGhost	BackBone	92.6	94	96.1	2.6	7.1
	Neck	*95.7*	*94.9*	*97.1*	*2.6*	*7.3*
	All	95.4	93.4	96.5	2.2	6.3
GhostConv+C2f_RepGhost	*Neck*	*95.5*	*95.1*	*97*	*2.5*	*7.2*

TABLE 4. Performance comparison of loss functions on algorithm impact.

Algorithms	Loss function	precision /%	recall/ %	mAP50 /%	parameters/M	FLOPs /G
GhostConv+C2f_RepGhost	CIoU	95.5	95.1	97	2.5	7.2
	DIoU [24]	95.5	95.6	97.6		
	EIoU [47]	96.4	94.2	96.7		
	GIoU [48]	95.6	92.5	96.4		
	SIoU [49]	95.6	94.2	96.5		
	MPDIoU [50]	96.6	94.2	97		
	WIoU_v1	95.2	94	96.7		
	WIoU_v2	97.2	93.6	96.8		
	WIoU_v3	*96.4*	*95.5*	*97.1*		

TABLE 5. Comparison of results of different improvement points added to the model.

GhostConv	C2f_RepGhost	WIoU	SA	precision/%	recall /%	mAP50 /%	parameters/M	FLOPs /G
				95.4	94.4	96.9	3	8.1
√				96.5	94.4	97.1	2.9	8.0
	√			95.7	94.9	97.1	2.6	7.3
		√		95.1	95.4	97	3	8.1
			√	96.1	93.4	96.8	3	8.1
√	√			95.5	95.1	97	2.5	7.2
√	√	√		96.4	95.5	97.1	2.5	7.2
√	√	√	√	97.1	94.5	97.1	2.5	7.2

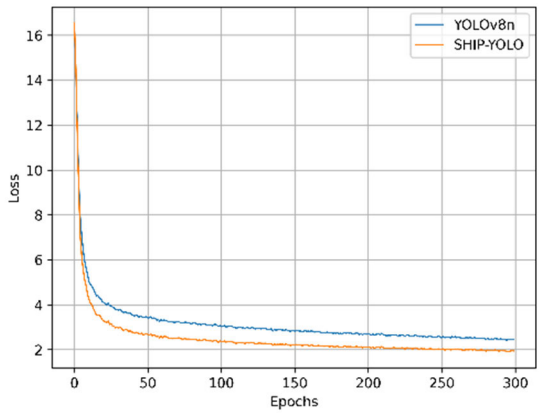

FIGURE 9. The loss curves of the proposed SHIP-YOLO and the original YOLOv8n model.

the most significant improvement in algorithm performance, with a 0.9% increase in precision, 0.4% increase in recall, and 0.1% increase in mAP50.

TABLE 6. The experimental results of SHIP-YOLO as compared to other models.

Algorithms	SSDD			SAR-Ship-Dataset			parameters/M	FLOPs/G
	precision/%	recall /%	mAP50 /%	precision/%	recall /%	mAP 50/%		
YOLOv3-tiny	94	90.4	95.3	94.2	94.2	96.8	12	18.9
YOLOv5n	95.1	94	96.6	92.2	92.9	96.2	2.5	7.1
YOLOv6n	96.9	94.1	97	91.3	93.2	95.8	4.2	11.8
YOLOv7-tiny	92.9	94.9	96.5	90.7	90.4	94.2	6	13
RT-DERT	89.6	87.7	93.3	-	-	-	32	103.4
YOLOv8n	95.4	94.4	96.9	92.8	92.6	96.4	3	8.1
CRAS-YOLO	97.3	95.5	98.7	-	-	-	10.3	19.7
Vessel-YOLO	-	-	97.9	-	-	96.8	4.8	9.5
SHIP-YOLO	*97.1*	*94.5*	*97.1*	*93.2*	*92.8*	*96.6*	*2.5*	*7.2*

F. ABLATION EXPERIMENT

To validate the effectiveness of the various improvement strategies proposed in this study, ablation experiments were designed to assess the impact of these strategies on the model detection performance under identical experimental conditions. The results presented in Table 5 reveal that following the incorporation of GhostConv and C2f_RepGhost modules into the YOLOv8n baseline algorithm, there is a reduction in both the number of parameters and computations, accompanied by improvements in precision, recall, and mAP50. Upon introducing WIoU on this foundation, while precision experiences a slight decline, recall shows a 1.1% improvement, indicating the contribution of WIoU to the precision of bounding box localization. Finally, the inclusion of SA modules substantially increased the precision of the algorithm. SHIP-YOLO compared with the YOLOv8n baseline model, demonstrated a reduction of 0.5G in parameters and 0.9G in FLOPs, yielding a 1.7% increase in precision, a 0.1% increase in recall, and a 0.2 increase in mAP50. These experimental results underscore the effectiveness of the lightweight optimization proposed in this study for the SAR ship detection model.

Figure 9 shows the training loss curves for YOLOv8n and SHIP-YOLO on SSDD. It can be observed that SHIP-YOLO and YOLOv8n exhibit almost identical training loss decreases at the start of training. However, with the

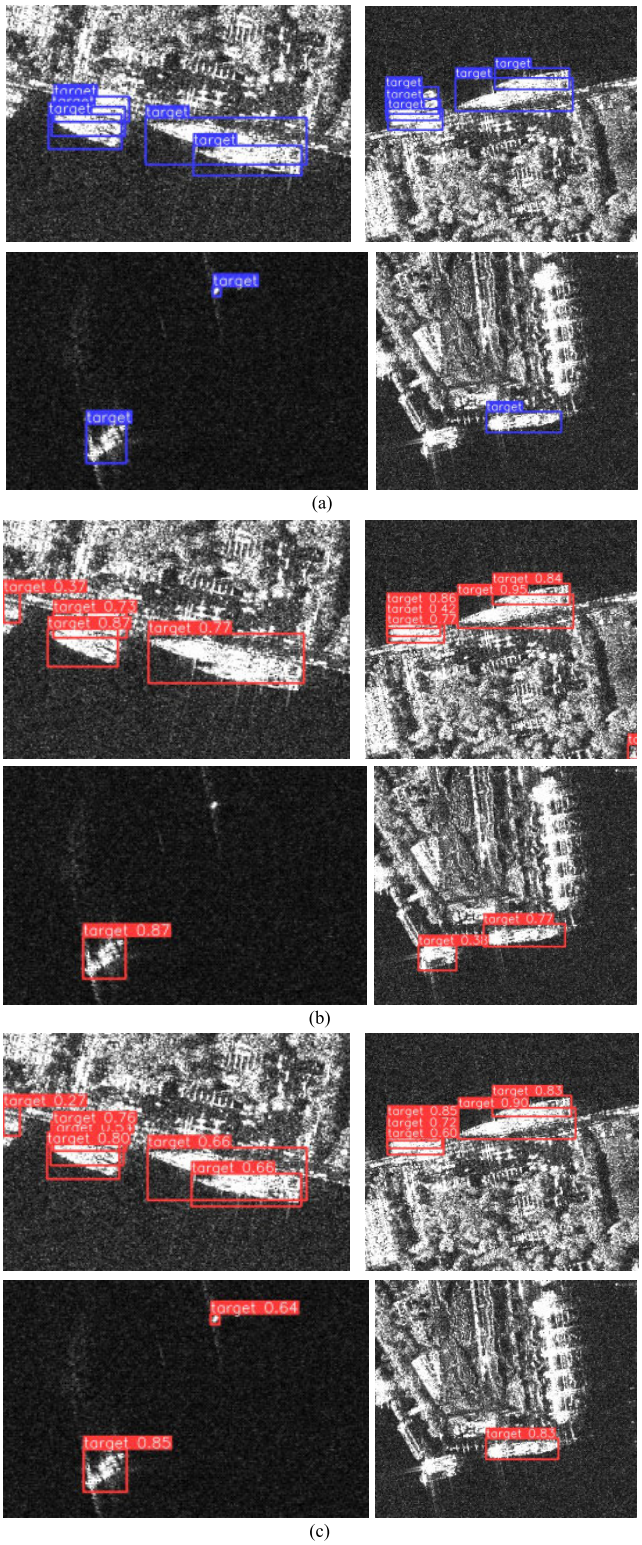

FIGURE 10. Ship detection results on SSDD: (a) is the tag visualization results of SSDD, (b) is the ship detection results based on YOLOv8n model, and (c) is the ship detection result based on the proposed SHIP-YOLO model.

progression of training, after 10 epochs, the training loss of SHIP-YOLO decreased at a faster rate than YOLOv8n. In summary, the proposed SHIP-YOLO model can more

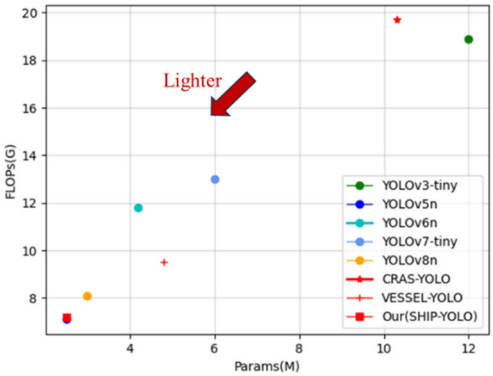

FIGURE 11. The comparison of FLOPs and parameters between the proposed SHIP-YOLO and other models.

effectively reduce loss and hastening model convergence. Figure 10 shows the ship detection results on SSDD, where a is the visualization result of the dataset label, b is the detection result of the original YOLOv8n, and c is the detection result of the proposed SHIP-YOLO model. YOLOv8n mistakenly identified some targets as ships and multiple ships parked side-by-side as the same ship. However, small ships were not detected in this study. The improved model proposed in this study avoids these limitations.

G. COMPARATIVE EXPERIMENT

To further verify the effectiveness of the proposed model, a comparative experiment was conducted against the lightest algorithms in the YOLO series, including YOLOv3-tiny, YOLOv5n, YOLOv6n, YOLOv7-tiny, YOLOv8n, and Baidu's Paddle team's RT-DERT [51], on the SSDD and SAR-Ship-Dataset. As indicated in Table 6, compared with the other algorithms, the proposed SHIP-YOLO achieved nearly the highest precision, recall, and mAP50, with nearly the lowest parameters and FLOPs. A comparison of the FLOPs and parameters of these models are shown in Figure 11. In comparison to YOLOv5n with similar parameters and FLOPs, SHIP-YOLO demonstrates a 2% and 1% increase in precision, 0.5% and 0.4% increase in recall, and 0.5% and 0.4% increase in mAP50 on the two datasets. Owing to experimental constraints, the RT-DERT algorithm has not been experimentally validated on the SAR-Ship-Dataset. However, based on the experimental results on SSDD, the parameters and FLOPs of the RT-DERT algorithm are much larger than those of the SHIP-YOLO model, and all detection metrics are inferior to SHIP-YOLO. Meanwhile, the SHIP-YOLO model maintains a detection performance similar to that of the CRAS-YOLO [52] and Vessel-YOLO [53] models, which are both SAR ship detection models. However, our proposed model is lightweight. Compared to all the aforementioned algorithms, the proposed SHIP-YOLO ship detection model not only demonstrates outstanding performance in detection but also maintains lower parameters and computations, showing significant overall performance and proving the feasibility and effectiveness of the proposed improvement algorithm.

V. CONCLUSION

This study introduced a lightweight SAR ship detection model, named SHIP-YOLO, based on the YOLOv8n algorithm. Innovative designs, including GhostConv, RepGhost, WIoU, and SA modules were incorporated to enhance both the detection accuracy and algorithmic lighting. A key advantage of this study, compared with existing research, is the significant achievement of algorithmic lighting while maintaining high accuracy. Specifically, this study integrated RepGhost structure into C2f module, replacing Conv in the neck network with GhostConv, and integrating SA modules into the backbone and neck networks to enhance the perception and feature representation capabilities of the model.

Experimental validation demonstrated the outstanding performance of the SHIP-YOLO model in SAR ship detection tasks. Compared with the baseline YOLOv8n algorithm, on SSDD and SAR-Ship-Dataset, the precision improved by 1.7% and 0.4%, recall improved by 0.1% and 0.2%, and mAP50 improved by 0.2%. Meanwhile, parameter count reduced from 3M to 2.5M, and FLOPs reduced from 8.1G to 7.2G, achieving lightweight model improvements. However, there are some potential areas for improvement in this study. First, the generalization ability of the model in specific scenarios needs further enhancement, possibly requiring more data augmentation techniques or domain adaptation methods to improve generalization performance. Second, inference efficiency in GPU environments still needs optimization, especially concerning the computational intensity of ghost structures, which may necessitate further research into optimizing GPU inference speed.

Future research directions may include optimizing network structures to further improve performance and efficiency, introducing small object detection layers to enhance feature extraction capabilities, and exploring more lightweight model methods, such as lightweight backbone networks, knowledge distillation, and network pruning, to further improve model performance and scalability.

REFERENCES

[1] B. O. Steenson, "Detection performance of a mean-level threshold," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-4, no. 4, pp. 529–534, Jul. 1968, doi: 10.1109/TAES.1968.5409020.

[2] L. M. Novak, M. C. Burl, and W. W. Irving, "Optimal polarimetric processing for enhanced target detection," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, no. 1, pp. 234–244, Jan. 1993, doi: 10.1109/7.249129.

[3] H. Lim, D. Chae, J. H. Yoo, and K.-I. Kwon, "Template matching-based target recognition algorithm development and verification using SAR images," *J. Korea Inst. Mil. Sci. Technol.*, vol. 17, no. 3, pp. 364–377, Jun. 2014, doi: 10.9766/kimst.2014.17.3.364.

[4] E. Grosso and R. Guida, "A new automatic ship wake detection for Sentinel-1 imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Waikoloa, HI, USA, Sep. 2020, pp. 1259–1262, doi: 10.1109/IGARSS39084.2020.9324604.

[5] Y. He, H. He, Y. Xu, Y. Wang, and J. Su, "Marine target detection based on improved wavelet transform," *Syst. Eng. Electr.*, vol. 42, no. 1, pp. 83–89, Aug. 2020, doi: 10.3969/j.issn.1001-506X.2020.01.12.

[6] J. Li, C. Qu, and J. Shao, "Ship detection in SAR images based on an improved faster R-CNN," in *Proc. SAR Big Data Era, Models, Methods Appl. (BIGSARDATA)*, Beijing, China, 2017, pp. 1–6, doi: 10.1109/BIGSARDATA.2017.8124934.

[7] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, "A SAR dataset of ship detection for deep learning under complex backgrounds," *Remote Sens.*, vol. 11, no. 7, p. 765, Mar. 2019, doi: 10.3390/rs11070765.

[8] S. Xian, W. Zhirui, and S. Yuanrui, "AIR-SARShip-1.0: High-resolution SAR ship detection dataset," *J. Radars*, vol. 8, no. 6, pp. 852–862, 2019, doi: 10.12000/jr19097.

[9] S. Lei, D. Lu, X. Qiu, and C. Ding, "SRSDD-v1.0: A high-resolution SAR rotation ship detection dataset," *Remote Sens.*, vol. 13, no. 24, p. 5104, Dec. 2021, doi: 10.3390/rs13245104.

[10] 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.*, Columbus, OH, USA, Jun. 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[12] 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.*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Computer Vision—ECCV*, vol. 9905, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, Sep. 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.

[15] Y. Zhang, X. Liu, and P. Zhou, "Ship detection technology in SAR images based on improved faster R-CNN," *Radio Eng.*, vol. 52, no. 12, pp. 2280–2287, Oct. 2022, doi: 10.3969/j.issn.1003-3106.2022.12.023.

[16] Y. Zhang, Y. Jia, and Y. Chen, "Improved RTMDet for SAR ship detection," *Comput. Eng. Appl.*, Sep. 2023. [Online]. Available: https://link.cnki.net/urlid/11.2127.TP.20230920.1326.052, doi: 10.3778/j.issn.1002-8331.2307-0175.

[17] T. Zhang and X. Zhang, "High-speed ship detection in SAR images based on a grid convolutional neural network," *Remote Sens.*, vol. 11, no. 10, p. 1206, May 2019, doi: 10.3390/rs11101206.

[18] T. Zhang, X. Zhang, J. Shi, and S. Wei, "Depthwise separable convolution neural network for high-speed SAR ship detection," *Remote Sens.*, vol. 11, no. 21, p. 2483, Oct. 2019, doi: 10.3390/rs11212483.

[19] X. Xu, X. Zhang, Z. Shao, J. Shi, S. Wei, T. Zhang, and T. Zeng, "A group-wise feature enhancement-and-fusion network with dual-polarization feature enrichment for SAR ship detection," *Remote Sens.*, vol. 14, no. 20, p. 5276, Oct. 2022, doi: 10.3390/rs14205276.

[20] X. Xu, X. Zhang, T. Zhang, Z. Yang, J. Shi, and X. Zhan, "Shadow-background-noise 3D spatial decomposition using sparse low-rank Gaussian properties for video-SAR moving target shadow enhancement," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2022.3223514.

[21] Q. Huang, G. Jin, X. Xiong, L. Wang, and J. Li, "Lightweight SAR target detection based on channel pruning and knowledge distillation," *Acta Geod. Cartogr. Sin.*, Nov. 2023. [Online]. Available: https://link.cnki.net/urlid/11.2089.P.20231109.1649.004, doi: 11.2089,p.20231109.1649.004.

[22] X. He, J. Wu, Y. Yu, X. Gao, and J. Wei, "Improvement of SAR ship target detection algorithm in complex background," *Comput. Technol. Dev.*, vol. 33, no. 11, pp. 41–49, Nov. 2023, doi: 10.3969/j.issn.1673-629X.2023.11.007.

[23] S. Zhang, M. Li, Y. Chen, and Z. Zhang, "Ship target detection algorithm in SAR images based on improved YOLOv7," *Electr. Opt. Contr.*, Nov. 2023. [Online]. Available: https://link.cnki.net/urlid/41.1227.TN.20231109.1001.002

[24] X. Ren, Y. Bai, G. Liu, and P. Zhang, "YOLO-lite: An efficient lightweight network for SAR ship detection," *Remote Sens.*, vol. 15, no. 15, p. 3771, Jul. 2023, doi: 10.3390/rs15153771.

[25] Y. Guo, S. Chen, R. Zhan, W. Wang, and J. Zhang, "LMSD-YOLO: A lightweight YOLO algorithm for multi-scale SAR ship detection," *Remote Sens.*, vol. 14, no. 19, p. 4801, Sep. 2022, doi: 10.3390/rs14194801.

[26] 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, doi: 10.3390/rs16030486.

[27] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.
[28] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, Jun. 2020, pp. 1577–1586.
[29] C. Chen, Z. Guo, H. Zeng, P. Xiong, and J. Dong, "RepGhost: A hardware-efficient ghost module via re-parameterization," 2022, *arXiv:2211.06088*.
[30] Z. Tong, Y. Chen, Z. Xu, and R. Yu, "Wise-IoU: Bounding box regression loss with dynamic focusing mechanism," 2023, *arXiv:2301.10051*.
[31] Q.-L. Zhang and Y.-B. Yang, "SA-net: Shuffle attention for deep convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Toronto, ON, Canada, Jun. 2021, pp. 2235–2239, doi: 10.1109/ICASSP39728.2021.9414568.
[32] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.
[33] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 12993–13000, doi: 10.1609/aaai.v34i07.6999.
[34] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Honolulu, HI, USA, 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.
[35] L. Yang, X. Dong, S. Huang, Y. Wang, and S. Gao, "Research on the defect detection algorithm of lightweight insulator based on YOLOV5," *Commun. Inf. Technol.*, vol. 5, pp. 13–18, Sep. 2023.
[36] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Vancouver, BC, Canada, Jun. 2023, pp. 7464–7475.
[37] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*.
[38] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.
[39] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[40] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[41] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," in *Proc. 34th Int. Conf. Neural Inform. Process. Syst.*, Vancouver, BC, Canada, Dec. 2020, pp. 21002–21012.
[42] X.-B. Liu, X.-Z. Yang, Y. Chen, and S.-T. Zhao, "Object detection method based on CIoU improved bounding box loss function," *Chin. J. Liquid Crystals Displays*, vol. 38, no. 5, pp. 656–665, 2023, doi: 10.37188/cjlcd.2022-0282.
[43] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 6848–6856.
[44] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis.*, Munich, Germany, 2018, pp. 122–138.
[45] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*.
[46] Y. Wu and K. He, "Group normalization," in *Proc. Eur. Conf. Comput. Vis.*, Munich, Germany, Sep. 2018, pp. 3–19.
[47] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, "Focal and efficient IOU loss for accurate bounding box regression," *Neurocomputing*, vol. 506, pp. 146–157, Sep. 2022, doi: 10.1016/j.neucom.2022.07.042.
[48] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Jun. 2019, pp. 658–666, doi: 10.1109/CVPR.2019.00075.
[49] Z. Gevorgyan, "SIoU loss: More powerful learning for bounding box regression," 2022, *arXiv:2205.12740*.
[50] M. Siliang and X. Yong, "MPDIoU: A loss for efficient and accurate bounding box regression," 2023, *arXiv:2307.07662*.
[51] W. Lv, Y. Zhao, S. Xu, J. Wei, G. Wang, C. Cui, Y. Du, Q. Dang, and Y. Liu, "DETRs beat YOLOs on real-time object detection," 2023, *arXiv:2304.08069*.
[52] 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.
[53] F. Ning, L. Zhao, L. Zheng, G. Liang, Y. Xi, and Z. He, "A ship object detection algorithm for SAR images," *Electr. Opt. Contr.*, Nov. 2023. [Online]. Available: https://link.cnki.net/urlid/41.1227.TN.20231116.1544.005

YONGHANG LUO received the B.S. degree from Xijing University, in 2022. He is currently pursuing the M.S. degree with the School of Computer and Information Science, Chongqing Normal University. His research interests include computer vision, including object detection, object classification, and natural language processing.

MING LI is currently pursuing the Ph.D. degree in applied economics with Xi'an Jiaotong University. Additionally, he is also a Professor and a Master Supervisor with the School of Computer and Information Science, Chongqing Normal University. His research interests include smart agriculture, cities, and e-commerce. Furthermore, he is also a member of the E-Commerce Education Guidance Committee of the Ministry of Education.

GUIHAO WEN received the B.S. degree from Chongqing University of Education, in 2022. He is currently pursuing the M.S. degree with the School of Computer and Information Science, Chongqing Normal University. His research interest includes object detection in computer vision. For example, the application of YOLO algorithm in crop pest and disease detection.

YUNFEI TAN received the B.S. degree from Chongqing College of Humanities, Science & Technology, in 2022. He is currently pursuing the M.S. degree with the School of Computer and Information Science, Chongqing Normal University. His research interest includes the detection and classification of hyperspectral satellite remote sensing images.

CHAOSHAN SHI received the B.S. degree from Sichuan Agricultural University, in 2022. He is currently pursuing the M.S. degree with the School of Computer and Information Science, Chongqing Normal University. His research interests include deep learning in natural language processing and temporal data prediction.