High-order Spatial Interactions Enhanced Lightweight Model for Optical Remote Sensing Image-based Small Ship Detection

Yin, Yifan; Cheng, Xu; Shi, Fan; Liu, Xiufeng; Huo, Huan; Chen, Shengyong

Published in:
IEEE Transactions on Geoscience and Remote Sensing

Link to article, DOI:
10.1109/TGRS.2023.3349168

Publication date:
2024

Document Version
Peer reviewed version

Citation (APA):
Yin, Y., Cheng, X., Shi, F., Liu, X., Huo, H., & Chen, S. (2024). High-order Spatial Interactions Enhanced Lightweight Model for Optical Remote Sensing Image-based Small Ship Detection. *IEEE Transactions on Geoscience and Remote Sensing*, *62*, Article 4201416. https://doi.org/10.1109/TGRS.2023.3349168

High-order Spatial Interactions Enhanced Lightweight Model for Optical Remote Sensing Image-based Small Ship Detection

Yifan Yin, Xu Cheng, *Member, IEEE,* Fan Shi, Xiufeng Liu, Huan Huo, and Shengyong Chen, *Senior Member, IEEE*

***Abstract*—Accurate and reliable optical remote sensing image-based small-ship detection is crucial for maritime surveillance systems, but existing methods often struggle with balancing detection performance and computational complexity. In this paper, we propose a novel lightweight framework called *HSI-ShipDetectionNet* that is based on high-order spatial interactions and is suitable for deployment on resource-limited platforms, such as satellites and unmanned aerial vehicles. HSI-ShipDetectionNet includes a prediction branch specifically for tiny ships and a lightweight hybrid attention block for reduced complexity. Additionally, the use of a high-order spatial interactions module improves advanced feature understanding and modeling ability. Our model is evaluated using the public Kaggle and FAIR1M marine ship detection datasets and compared with multiple state-of-the-art models including small object detection models, lightweight detection models, and ship detection models. The results show that HSI-ShipDetectionNet outperforms the other models in terms of detection performance while being lightweight and suitable for deployment on resource-limited platforms.**

***Index Terms*—Small ship detection, Optical remote sensing images, Convolutional neural networks, Spatial interaction, Lightweight model.**

I. Introduction

MONITORING the position and behavior of ships plays a critical role in maintaining marine traffic safety and supporting social and economic development. The use of optical remote sensing images provides valuable information for various applications such as fishery management, marine spatial planning, marine casualty investigation, and pollution treatment [1], [2]. However, when the altitude and angle of satellite photography vary, ship targets can have a large scale of variation, so there are a large number of small target ships in the images. The complex sea state can significantly impact the detection performance of small ships. Waves can cause variations in pixel values in the optical image due to the reflection of the sun and skylight off their slopes [3]. Additionally, satellites may encounter clouds or sunglint when observing the Earth, which can make it difficult to distinguish ships from the background, even for the naked eye [4]. Therefore, it is still difficult to accurately locate and recognize small ships from optical remote sensing images.

In the field of object detection, small objects can be defined in two ways. Chen et al. [5] propose a definition based on relative sizes, stating that small objects have a median relative area between 0.08% and 0.58% compared to other instances in the same category. Another way to define small objects is based on absolute sizes. In this approach, objects with occupied areas equal to or less than 32×32 pixels are considered small objects [6]. Over the past few decades, there has been a significant amount of research on small ship detection in optical remote sensing images. Traditional methods have mainly focused on feature design, including ship candidate extraction and ship identification [7]. Ship candidate extraction techniques such as statistical threshold segmentation [8], [9], visual saliency [10], and local feature descriptor [11] have been commonly used. In the identification stage, the support vector machine (SVM) [12] has been a frequently adopted method for ship classification. However, traditional methods may not be effective in complex conditions as the impact of variable weather factors on optical image imaging is uncontrollable. Additionally, these algorithms rely heavily on manual and expert experience for feature production and generation, resulting in poor generalization ability.

Recently, the use of convolutional neural networks (CNNs) has greatly improved the accuracy and efficiency of ship detection. However, the continuous downsampling characteristic of CNNs can still present challenges for detecting small ships in optical remote sensing images. One important way to improve the detection accuracy of small objects is to address the issue of multi-scale feature learning. Shallow layers of convolutional neural networks (CNNs) typically have higher resolutions and smaller receptive fields, which are more suitable for detecting small objects [13]. Several methods have been developed to make use of these shallow layers for small object detection, including the Single Shot MultiBox Detector (SSD) [14] and the top-down feature pyramid network (FPN) with lateral connections [15]. In addition to multi-scale feature learning, the use of contextual

Yifan Yin, Xu Cheng, Fan Shi, and Shengyong Chen are with the School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China, 300386.

Xiufeng Liu is with the Department of Technology, Management, and Economics, Technical University of Denmark, Produktionstorvet, Denmark, 2800.

Huan Huo is with the School of Computer Science, the University of Technology Sydney, Sydney, Australia, 9220.

Corresponding author: Xu Cheng and Fan Shi.

information can also be beneficial for improving object detection performance, particularly for small objects with insufficient pixels [13]. This is because specific objects often appear in specific environments, such as ships sailing in the sea. Context-based small object detection methods can be divided into two categories: local context modeling [16], [17] and global context modeling [18]–[20].

Despite the advancements made by CNN-based detection networks in improving the detection performance of small objects, several limitations persist. These limitations include:

- In this study, real-time performance is crucial for detecting ships in satellite images. To achieve this, the You Only Look Once (YOLO) [21]–[24]networks are employed as a one-stage algorithm, treating object detection as a regression problem. This approach significantly improves the detection speed. However, a direct application of YOLOv5 to the task is not suitable due to the large number of small ships present in satellite imagery. The limitation arises from the fact that YOLOv5 underutilizes shallow features, which are essential for detecting small objects. As the deep feature maps are fused to the neck of YOLOv5, excessive downsampling occurs, leading to the loss of valuable information about small objects.
- The use of CNN-based models for small object detection has been shown to be effective, but these models often have a high number of parameters and are complex. For example, the TPH-YOLOv5 detector [20] is well-known for its proficiency in small object detection, but it requires 60 million parameters. This complexity can lead to time delays when transmitting data from the platform to ground stations for processing [25]. To address this, it is necessary to migrate ship detection models from ground to space-borne platforms. However, hardware resources on such platforms are often limited, such as the NVIDIA Jetson TX2 which only has 8 GB of memory [26]. This makes it difficult to reduce model complexity while still maintaining accuracy in ship detection. Therefore, finding an optimal balance between model accuracy and complexity is an ongoing research challenge.
- As the depth of the network layers increases, the high-level features at the end of the backbone exhibit an abundance of combinatorial information. While these higher-level features carry richer semantic information, the location information they convey is ambiguous. This ambiguity can negatively impact the accuracy of small object detection, particularly for objects with insufficient pixels [27], making it challenging to accurately localize and regress small target ships. Additionally, the complexity of the background texture and harsh environmental conditions can weaken the ability of CNNs to extract features of ships, making it difficult to distinguish small ships from their background.

Given the limitations of existing methods in small ship detection and the need to balance detection performance with the limited storage space available on satellites, this paper proposes a novel lightweight ship detection framework based on high-order spatial interactions (HSI). The contributions of this study can be summarized as follows:

- This study proposes an enhanced ship detection network, HSI-ShipDetectionNet, which is designed to be more lightweight and effective for ship detection in optical remote sensing images. Furthermore, the proposed network demonstrates improved accuracy in the localization and identification of small ships.
- To make the detection model more accurate in detecting tiny ships, we add a predictive branch of tiny ships (P_{tiny}). To support this branch, we increase the number of layers in the neck of the detection frame, making the model more sensitive to tiny ships. Then, we design a lightweight hybrid attention block (LHAB) to replace the SE block in GhostNet, which is the backbone of the HSI-ShipDetectionNet, reducing the number of parameters, computations, and storage space required by the model. Finally, A high-order spatial interactions (HSI-Former) module is introduced at the tail of the backbone, extending the interaction between spatial elements to any order and strengthening the model's ability to understand and process advanced features in deep layers.
- We comprehensively evaluate the proposed ship detection framework using optical satellite remote sensing images. The performance of the proposed model is compared with that of state-of-the-art small object detection models, lightweight detection models, and ship detection models. The experimental results indicate that the proposed HSI-ShipDetectionNet demonstrates remarkable performance in detecting small ships under diverse sea conditions. Furthermore, the lightweight nature of the proposed model makes it highly suitable for deployment on resource-constrained satellite platforms.

The remainder of this article is organized as follows: Section II reviews related work on this topic. Section III outlines the framework of our discussed methodology. Section IV describes the experimental results and analysis, and Section V concludes the whole study.

II. Related Work

A. Methods for Small Ship Detection

Accurate and dependable detection of small ships is crucial for maritime surveillance systems. In recent years, there have been numerous efforts to improve the performance of small ship detection.

With the development of deep learning, the use of convolutional neural networks (CNNs) for ship detection has become mainstream. For example, Wu et al. [28] proposed a multi-scale detection strategy that uses a coarse-to-fine ship detection network (CF-SDN) with a feature pyramid network (FPN) to improve the resolution and semantic information of shallow and deep feature maps, respectively. Xie et al. [29] introduced an adaptive feature enhancement (AFE) module into FPN to adaptively reinforce the locations of deep ship features based on shallow features with rich spatial information. Wang et al. [30] developed a ship detection model based on YOLOX that incorporates a multi-scale convolution (MSC) for feature fusion and a feature transformer module (FTM) for context modeling. Jin et al. [31] input patches containing targets and surroundings into a CNN to improve small ship detection results. Tian et al. [4] proposed an image enhancement module base on generative adversarial network (GAN), and introduced the receptive field expansion module to improve the capability to extract features from target ships of different sizes.

Despite the remarkable detection performance demonstrated by existing ship detection models, these models are often characterized by large and complex network architectures, as evidenced by the substantial number of parameters and computational demands. This presents a significant challenge for resource-constrained applications, where the available hardware resources are limited. To overcome this limitation, we design a lightweight attention block and construct a lightweight ship detection framework, which reduces the number of parameters, computations, and storage space required by the model.

B. Methods for Lightweight CNNs

Lightweight design of CNNs is crucial for deploying models to resource-limited devices such as satellites, as it helps to reduce the number of parameters and computational requirements. A number of approaches have been proposed in the literature to achieve this goal, including SqueezeNet [32], which reduces the number of parameters by using 1×1 convolution kernels to decrease the size of the feature maps; the MobileNet series [33]–[35], which uses depthwise separable convolution to factorize standard convolution into a depthwise convolution and a pointwise convolution, reducing the number of parameters and computational requirements; ShuffleNet [36], [37], which replaces pointwise convolution with pointwise group convolution and performs channel shuffle to further reduce the number of parameters and address the disadvantages of group convolution; and GhostNet [38], which embraces abundant and redundant information through cheap operations as a cost-efficient way to improve network performance.

In the field of ship detection, it can be challenging to balance the performance and computational complexity of the model. To address this issue, Li et al. [39] optimized the backbone of YOLOv3 using dense connections and introduced spatial separation convolution to replace standard convolution in FPN, resulting in a significant reduction in parameters. Jiang et al. [40] developed YOLO-V4-light by reducing the number of convolutional layers in CSP-DarkNet53. Liu et al. [41] also improved upon YOLOv4 by substituting the original backbone with MobileNetv2, significantly reducing the complexity of the ship detection model. Zheng et al. [42] used BN scaling factor γ to compress the YOLOv5 network, achieving higher detection accuracy and shorter computational time compared to other object detection models.

C. Methods for Attention Mechanism

Attention mechanisms have become a key concept in the field of computer vision, with the ability to significantly improve the performance of networks [43]. Channel attention allows networks to model dependencies between the channels of their convolutional features, such as in the Squeeze-and-excitation (SE) network [44], which adaptively recalibrates channel-wise features using global information to selectively highlight important features. Wang et al. [45] further developed this concept with the efficient channel attention (ECA) module, which can be implemented using 1D convolution and has been shown to be more efficient and effective. Spatial attention, on the other hand, focuses on identifying specific positions in the image that should be emphasized, such as in CCNet [46], which captures full-image contextual information using criss-cross attention. The Convolutional Block Attention Module (CBAM) [47] combines channel and spatial attention, emphasizing important features in both dimensions.

The Transformer model, proposed by Vaswani et al. [48], has been a major milestone in the development of attention mechanisms, and its application to the field of computer vision is known as the Vision Transformer (ViT) [49]. The core idea of the Transformer is to use self-attention to dynamically generate weights that establish long-range dependencies. Self-attention achieves this through matrix multiplication between queries, keys, and values, allowing for the interaction of two spatial elements. However, it has been noted that the Transformer architecture is limited in its capability to model higher-order spatial interactions, which can potentially enhance the overall visual modeling performance [50]. In this work, we propose a novel lightweight ship detection framework for small ships that includes the following elements: an extension of FPN through the addition of a predictive branch for tiny ships, the use of the lightweight hybrid attention block (LHAB), and the introduction of the high-order spatial interactions (HSI-Former) module, resulting in more accurate and reliable ship detection in surveillance systems. Ablation and comparison experiments will be conducted to demonstrate the superior performance of our model.

III. Methodology

A. Overview

The proposed lightweight HSI-ShipDetectionNet for small ship detection, as depicted in Fig. 1, consists of five components: the Input, the Backbone, the HSI-Former module, the Neck and the Output. The input optical remote sensing images undergo processing in the backbone, which extracts the detailed features of the ship. To address the challenge of small ship detection, a predictive branch specifically designed for tiny ships is added to the shallow layer of the backbone, as discussed in detail in Section III-B. To further reduce the complexity of the model, the Ghost bottleneck in GhostNet has been improved with the implementation of a new Lightweight Hybrid Attention Block (LHAB), which replaces the SE block [44]. This results in a LHAB-Gbneck with a reduced number of parameters, computational effort, and occupied storage space, as explained in Section III-C. In addition, the HSI-Former module, which is designed to reinforce contextual learning and modeling capability of advanced features in deep layers, is introduced at the tail of the backbone. The function and implementation of the HSI-Former module are detailed in Section III-D. Finally, the neck layer fuses the features, and four separate output heads are employed to predict tiny, small, medium, and large ship targets, respectively.

B. The Predictive Branch of Tiny Ships

The problem of low detection accuracy for small ships in satellite imagery is a well-known issue. This is due to the continuous down-sampling of features by the convolutional layers in the backbone, which results in the loss of resolution and information for small ships. To address this issue, we propose adding a branch that predicts tiny ships in part 1 of the backbone, as shown in Fig. 1. This branch, named P$_{tiny}$, is specifically designed to extract features before the continuous downsampling process. Additionally, the number of layers in the PANet in the neck of the detection frame is increased to enhance the feature fusion effect for tiny ships. This structure gradually fuses shallow features with deep layers, ensuring that the feature maps of different sizes contain both semantic information and feature information of ships. This ultimately ensures the detection accuracy of ships with different scales, particularly for tiny ships.

With the addition of the new branch P$_{tiny}$ and the increased number of layers in PANet, the resulting output is augmented by an additional layer, bringing the total count of outputs to four. Correspondingly, we also add an additional set of anchors specifically tailored for tiny ships based on the original three groups of anchors of YOLOv5, resulting in a total of four groups of anchors. Instead of using the anchors generated by COCO dataset as in the original YOLOv5, we employ clustering to generate new anchors specifically for ship sizes in our dataset. This makes the regression of the anchors more accurate.

C. LHAB-GhostCNN

We select GhostNet as the backbone of our lightweight ship detector and further simplified it. We name this architecture as *LHAB-GhostCNN*.

1) Ghost Module: The Ghost module is a crucial element of the proposed LHAB-GhostCNN architecture for small ship detection. Its purpose is to maintain the same number of feature maps as a standard convolution while reducing the number of parameters and computational effort. Specifically, when the dimension of the input feature maps are C and the dimension of the output feature maps after standard convolution are D, the Ghost module can also produce feature maps in D dimension while minimizing the number of parameters and computations. The process can be defined as follows.

For the input feature $X \in \mathbb{R}^{H \times W \times C}$, the m intrinsic feature maps are first generated by a standard convolution, represented by the set Y_1:

$$Y_1 = Conv\left(X\right), \quad Y_1 \in \mathbb{R}^{H' \times W' \times m} \tag{1}$$

where $m \leq D$. To obtain the desired D-dimensional feature maps, each of the m intrinsic feature maps in Y_1 undergoes s cheap operations, implemented through depthwise convolution (DW-Conv), resulting in $m \times s$ ghost feature maps Y_2:

$$\begin{aligned} &Y_2 = \Phi\left(Y_1\right) : y_{ij} = DW\_Conv_{ij}\left(y_i\right), \\ &\forall i = 1,\cdots,m, \quad j = 1,\cdots,s \end{aligned} \tag{2}$$

where y_i represents the i-th intrinsic feature map in Y_1, and the j-th feature map y_{ij} is generated by the j-th linear operation $DW - Conv_{ij}$. As a result, these m intrinsic feature maps can eventually generate $m \times s$ feature maps, that is, $Y_2 \in \mathbb{R}^{H' \times W' \times ms}$. The final output of the Ghost module is the concatenation of Y_1 and Y_2:

$$Y_{out} = Y_1 \oplus Y_2 \tag{3}$$

By employing the Ghost module, D-dimensional feature maps can be obtained while maintaining the same number of feature maps as a standard convolution. Consequently, the output feature maps Y_{out} have a dimension of $m + ms = D$.

Analysis of complexities. We define r_F as the speed-up ratio of FLOPs of the Ghost module to FLOPs of the standard convolution:

$$\begin{aligned} r_F &= \frac{k \cdot k \cdot C \cdot m \cdot H' \cdot W' + d \cdot d \cdot m \cdot s \cdot H' \cdot W'}{k \cdot k \cdot C \cdot D \cdot H' \cdot W'} \\ &= \frac{C \cdot m + m \cdot s \cdot 9}{C \cdot m \cdot (1+s)} = \frac{C + s \cdot 9}{C \cdot (1+s)} \approx \frac{1}{1+s} \end{aligned} \tag{4}$$

where $k = 1$ is the standard convolution kernel size, while $d = 3$ is the kernel size of each linear operation, and $C \gg s$. Similarly, the compression ratio r_P of the parameters of the Ghost module to the parameters of the standard convolution is:

$$\begin{aligned} r_P &= \frac{k \cdot k \cdot C \cdot m + d \cdot d \cdot m \cdot s}{k \cdot k \cdot C \cdot D} \\ &= \frac{C \cdot m + m \cdot s \cdot 9}{C \cdot m \cdot (1+s)} = \frac{C + s \cdot 9}{C \cdot (1+s)} \approx \frac{1}{1+s} \end{aligned} \tag{5}$$

In this paper, we set the value of s to 1. As a result, the Ghost module can effectively reduce the number of parameters and the computational effort of the network by half.

2) LHAB-Gbneck: Similar to the basic residual block in ResNet [51], the Ghost bottleneck with LHAB (LHAB-Gbneck) integrates two Ghost modules and a shortcut, as shown in Fig. 2. The first Ghost module serves as an expansion layer to increase the number of channels, while the second Ghost module reduces the number of channels to match the shortcut connection. The shortcut is connected between the inputs and outputs of these two Ghost modules. When Stride=2, a depthwise convolution (DW-Conv) is added after the first Ghost module to reduce the size of

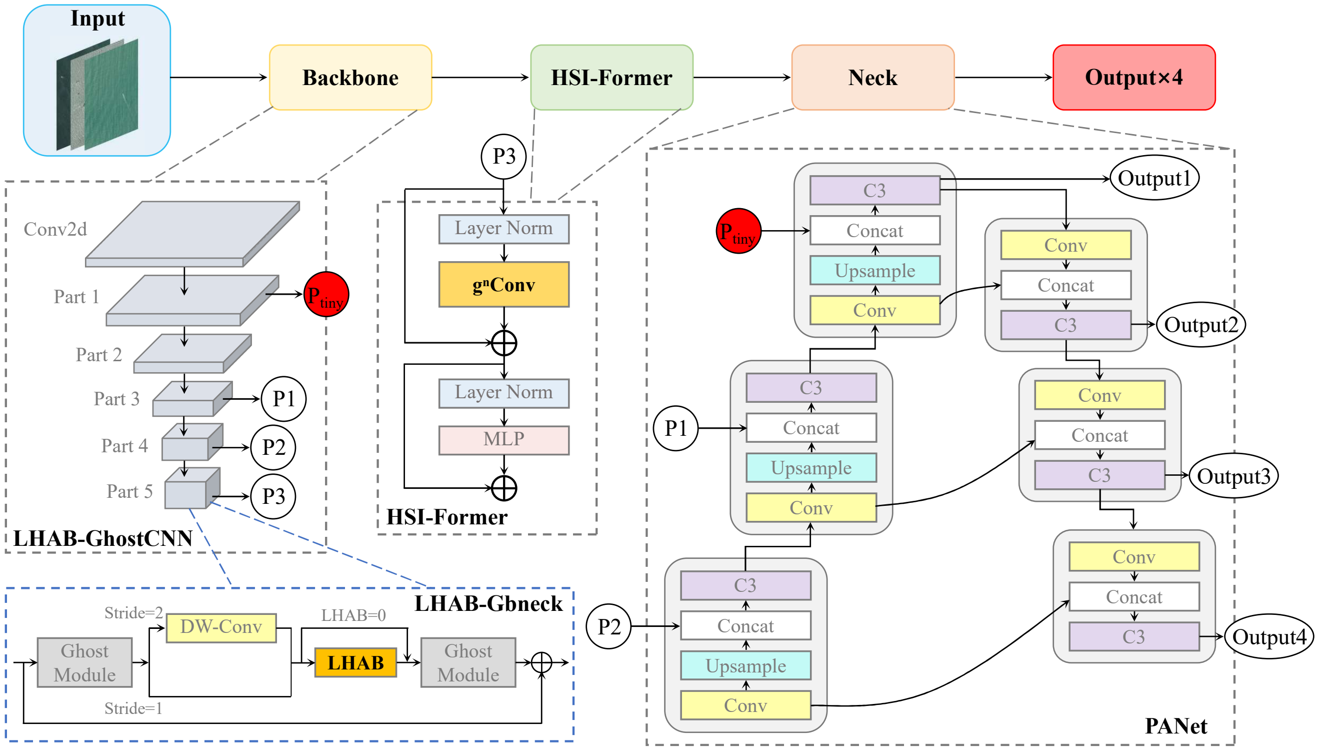


Fig. 1. Overview of the proposed HSI-ShipDetectionNet for small ship detection in optical remote sensing images. In the **Backbone**, a predictive branch is added to the shallow layer specifically for detecting tiny ships. The **Lightweight Hybrid Attention Block (LHAB)** in **LHAB-Gbneck** is designed, resulting in a significant reduction in the number of parameters, computational effort, and storage space required by the network. The **HSI-Former module** is added to the end of the Backbone to enhance the contextual learning and modeling of advanced features in the deep layers. The **Neck** layer then performs feature fusion, and four output heads are used to predict tiny, small, medium, and large ships respectively.

the feature maps by half, at this time the shortcut path goes through a downsampling layer to match the size of the feature maps. If LHAB=1, the Lightweight Hybrid Attention Block (LHAB) is selected. Compared with the SE attention [44] used in the original Ghost bottleneck, LHAB can further reduce the complexity of the network while enhancing the response of key features.

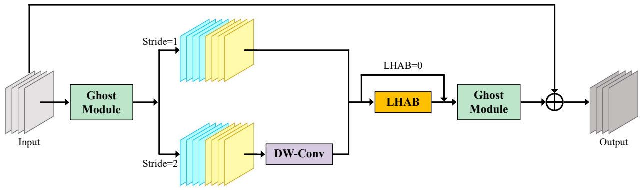


Fig. 2. LHAB-Gbneck. Stride=1 and Stride=2 go through different branches.

The SE block, a widely used channel attention mechanism, has limitations in ignoring spatial attention and adding complexity to the model. To balance the trade-off between model performance and complexity, we propose the Lightweight Hybrid Attention Block (LHAB), which is a lightweight and efficient attention block. LHAB consists of a channel attention block and a spatial attention block, enabling it to highlight significant information in both dimensions simultaneously.

Channel Attention Block. The Channel Attention Block (CAB) is a key component of the LHAB, which aims to capture interdependencies between channels. SENet [44] employed global average pooling to aggregate channel-wise statistics, but it overlooks the potential of max-pooling in inferring fine channel attention, as pointed out by Woo et al. [47]. Therefore, they proposed to use both average-pooling and max-pooling operations in tandem and generated the channel attention map using a shared network. In contrast, we believe that max-pooled features and average-pooled features each play distinct roles and therefore require dedicated parameters to store unique feature information. Therefore, we do not use shared parameters and instead employ two different one-dimensional convolutions for the max-pooled features and average-pooled features, respectively. This approach allows us to store different information and acquire cross-channel interactions without reducing the channel dimensionality. Furthermore, since we use one-dimensional convolution, the increase in the number of parameters is negligible even if no parameters are shared. The specific operation details are outlined below.

As shown in Fig. 3, we simultaneously apply max-pooling and average-pooling operations to the input feature map $\mathbf{U} \in \mathbb{R}^{\mathbf{H}\times\mathbf{W}\times\mathbf{C}}$, generating max-pooled features $\mathbf{U}_{\mathbf{C}}^{\mathbf{max}}$ and average-pooled features $\mathbf{U}_{\mathbf{C}}^{\mathbf{avg}}$, respectively. In contrast to SE [44], which used fully connected layers to achieve cross-channel interactions, we use two different one-dimensional convolutions ($\mathbf{C1D_k}$) of size k for $\mathbf{U_C^{max}}$ and $\mathbf{U_C^{avg}}$, respectively, to avoid the negative effects of channel dimensionality reduction and reduce model complexity. The kernel size k is defined as the coverage of k neighbors to participate in the interaction between channels, which is calculated using the equation from ECA-Net [45]:

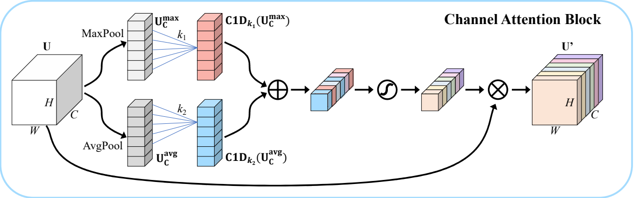


Fig. 3. Diagram of Channel Attention Block (CAB) of LHAB. Due to the max-pooling operations and average-pooling operations playing different roles in aggregating spatial dimension information, we design an adaptive channel attention block containing these two operations. The max-pooled and average-pooled features are passed through two separate one-dimensional convolutions, and then activated by the sigmoid function. The resulting vectors are then multiplied by the input feature map for adaptive feature refinement.

$$k = \psi\left(C\right) = \left|\frac{log_2\left(C\right)}{\gamma} + \frac{b}{\gamma}\right|_{odd} \tag{6}$$

where C is the number of channels and $|t|_{odd}$ represents the nearest odd number of t. γ and b are set to 2 and 1 respectively in this paper. Through the mapping ψ, kernel size k can be adaptively confirmed by the number of channels C.

Then we merge these two feature vectors $\mathbf{C1D_{k_1}}\left(\mathbf{U_C^{max}}\right)$ and $\mathbf{C1D_{k_2}}\left(\mathbf{U_C^{avg}}\right)$ using element-wise summation and pass the result through the sigmoid function. The final outcome is obtained by multiplying the original feature map $\mathbf{U}$ with the result of the sigmoid function to obtain $\mathbf{U'}$ for adaptive feature refinement. In a word, the CAB is summarized as:

$$\begin{aligned}\mathbf{U'} &= \sigma\left(\mathbf{C1D_{k_1}}\left(MP\left(\mathbf{U}\right)\right) \oplus \mathbf{C1D_{k_2}}\left(AP\left(\mathbf{U}\right)\right)\right) \otimes \mathbf{U}\\ &= \sigma\left(\mathbf{C1D_{k_1}}\left(\mathbf{U_C^{max}}\right) \oplus \mathbf{C1D_{k_2}}\left(\mathbf{U_C^{avg}}\right)\right) \otimes \mathbf{U}\end{aligned} \tag{7}$$

Where σ refers to sigmoid function. MP and AP refer to the max-pooling operation and average-pooling operation respectively.

Spatial Attention Block. To strengthen the inter-spatial relationship of features, we design a Spatial Attention Block (SAB). Similar to channel attention block, we first apply max-pooling and average-pooling operations along the channel axis to generate two 2D feature maps and then send them to two different two-dimensional convolution layers, which do not share parameters. We describe the detailed operation below.

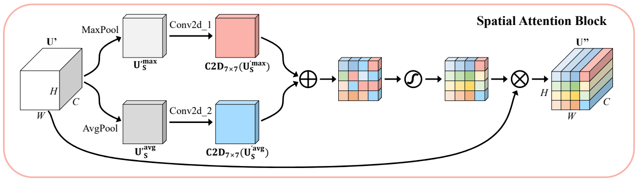


Fig. 4. Diagram of Spatial Attention Block (SAB) of LHAB. Due to the max-pooling operations and average-pooling operations playing different roles in aggregating channel dimension information, we design an adaptive spatial attention block containing these two operations. Then, the two 2D maps are passed through two different two-dimensional convolutions and further activated by the sigmoid function. Finally, the resulting vectors are multiplied by the input feature map for adaptive feature refinement.

As shown in Fig. 4, for the intermediate feature map $\mathbf{U'} \in \mathbb{R}^{\mathbf{H\times W\times C}}$ from the channel attention block, we aggregate channel information by max-pooling and average-pooling operations to obtain two new maps: $\mathbf{U'^{max}_S} \in \mathbb{R}^{\mathbf{H\times W\times 1}}$and $\mathbf{U'^{avg}_S} \in \mathbb{R}^{\mathbf{H\times W\times 1}}$. Those are then convolved by two different two-dimensional convolution layers ($\mathbf{C2D_{7\times7}}$), respectively. The kernel size of these two-dimensional convolutions is 7×7, which helps to generate larger receptive fields. Then, we merge these two feature maps $\mathbf{C2D_{7\times7}}\left(\mathbf{U'^{max}_S}\right)$ and $\mathbf{C2D_{7\times7}}\left(\mathbf{U'^{avg}_S}\right)$ using element-wise summation. The result is activated by the sigmoid function and finally $\mathbf{U'}$ multiply it to get the end map $\mathbf{U''}$. In a word, the SAB is summarized as:

$$\begin{aligned}\mathbf{U''} &= \sigma\left(\mathbf{C2D_{7\times7}}\left(MP\left(\mathbf{U'}\right)\right) \oplus \mathbf{C2D_{7\times7}}\left(AP\left(\mathbf{U'}\right)\right)\right) \otimes \mathbf{U'}\\ &= \sigma\left(\mathbf{C2D_{7\times7}}\left(\mathbf{U'^{max}_S}\right) \oplus \mathbf{C2D_{7\times7}}\left(\mathbf{U'^{avg}_S}\right)\right) \otimes \mathbf{U'}\end{aligned} \tag{8}$$

In conclusion, the LHAB module is composed of a CAB and a SAB, arranged sequentially with the CAB being in front of the SAB. The LHAB can make a significant reduction in the number of parameters, the computational effort, and the occupied storage space of the network, while still effectively capturing important information from the feature maps.

3) LHAB-GhostNet: The architecture of the proposed LHAB-GhostNet, which serves as the backbone for the HSI-ShipDetectionNet, is summarized in Table I. In this table, the parameters Exp and Out indicate the number of intermediate and output channels, respectively, and s represents the stride. The architecture of LHAB-GhostNet is based on GhostNet [38], with the G-bneck replaced by LHAB-Gbneck. The first layer of LHAB-GhostNet is a standard convolution operation, and the network is divided into 5 parts based on the input feature map sizes. The stride of the last LHAB-Gbneck in each part (except for part 5) is set to 2. Furthermore, LHAB is integrated into some LHAB-Gbnecks, as illustrated in Table I, to further simplify the backbone.

D. High-Order Spatial Interaction Mechanism

In recent years, the Transformer has gained popularity in vision applications and has challenged the dominance of CNNs by achieving excellent results. Scholars have started exploring the use of Transformer in the field of small object detection, as seen in recent studies such as [20] and [52]. The success of Transformer in vision tasks can be attributed

TABLE I
LHAB-GHOSTNET ARCHITECTURE.

Part	Input size	Operator	Exp	Out	LHAB	S
—	$640^2 \times 3$	Conv2d	—	16	—	2
part 1	$320^2 \times 16$	LHAB-Gbneck	16	16	0	1
	$320^2 \times 16$	LHAB-Gbneck	48	24	0	2
part 2	$160^2 \times 24$	LHAB-Gbneck	72	24	0	1
	$160^2 \times 24$	LHAB-Gbneck	72	40	1	2
part 3	$80^2 \times 40$	LHAB-Gbneck	120	40	1	1
	$80^2 \times 40$	LHAB-Gbneck	240	80	0	2
part 4	$40^2 \times 80$	LHAB-Gbneck	184	80	0	1
	$40^2 \times 80$	LHAB-Gbneck	184	80	0	1
	$40^2 \times 80$	LHAB-Gbneck	184	80	0	1
	$40^2 \times 80$	LHAB-Gbneck	480	112	1	1
	$40^2 \times 112$	LHAB-Gbneck	672	112	1	1
	$40^2 \times 112$	LHAB-Gbneck	672	160	1	2
part 5	$20^2 \times 160$	LHAB-Gbneck	960	160	0	1
	$20^2 \times 160$	LHAB-Gbneck	960	160	1	1
	$20^2 \times 160$	LHAB-Gbneck	960	160	0	1
	$20^2 \times 160$	LHAB-Gbneck	960	160	1	1

to self-attention. Self-attention's ability to capture long-range dependencies allows the model to learn contextual information more effectively.

Despite its effectiveness, self-attention has some limitations that need to be addressed. For instance, its spatial interaction ability is limited to two orders by performing matrix multiplication between queries, keys, and values, while research by Rao et al. [50] has shown that higher-order spatial interactions can improve visual models' modeling ability. Moreover, self-attention introduces a quadratic complexity as it requires each token to attend to every other token. Lastly, self-attention lacks some of the inductive biases present in CNNs, which can make it difficult to generalize well with limited data. To overcome these limitations, we introduce the Iterative Gated Convolution (g^{n}Conv), a convolution-based architecture that replaces self-attention in our method. Specifically, we take g^{3}Conv as an example to illustrate its principle, as shown in Fig. 5.

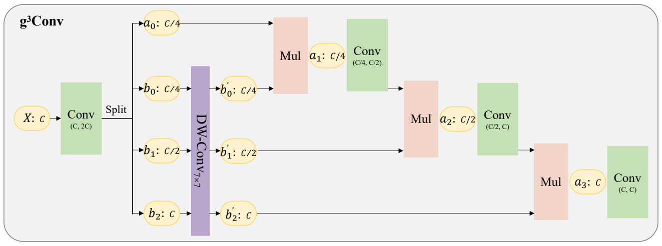


Fig. 5. **g^{3}Conv**. We take g^{3}Conv as an example to illustrate g^{n}Conv's principle. This module can extend the spatial interactions to three orders so that the correlation between features is gradually enhanced through the multiplication.

To process the input feature $X \in \mathbb{R}^{H\times W\times C}$, we first use a linear projection layer implemented as a 1×1 convolution operation to mix the channels. After this operation, the number of channels is doubled to obtain the intermediate feature $X' \in \mathbb{R}^{H\times W\times 2C}$. The formula for this process can be expressed as follows:

$$X' = Conv_{in}\left(X\right) \tag{9}$$

Then, the feature map X' is split along the channel dimension, which is expressed as follows:

$$\left[a_0, b_0, b_1, b_2\right] = Split\left(X'\right) \tag{10}$$

where the number of channels for a_0 is $\frac{C}{4}$, and the number of channels for b_0, b_1, and b_2 is $\frac{C}{4}$, $\frac{C}{2}$, and C, respectively. Then, the depthwise convolution (DW-Conv) is performed on b_0, b_1, and b_2, and the results are iteratively subjected to gated convolution operations with a_0, a_1, and a_2, respectively by:

$$\begin{aligned} a_1 &= h_0\left(a_0\right) \otimes DW\_Conv_0\left(b_0\right) \\ a_2 &= h_1\left(a_1\right) \otimes DW\_Conv_1\left(b_1\right) \\ a_3 &= h_2\left(a_2\right) \otimes DW\_Conv_2\left(b_2\right) \end{aligned} \tag{11}$$

where $\otimes$ is the multiplication of the elements in the matrix at the corresponding positions. The role of $\{h_i\}$ is to change the number of channels of a_i to match the number of channels of b_i. When $i = 0$, h_0 is an identity mapping; when i is 1 or 2, h_i doubles the channels of a_i. Finally, the a_3 received from the above steps is continued into a linear projection implemented as a 1×1 convolution operation to obtain the final result of g^{3}Conv:

$$y = Conv_{out}\left(a_3\right) \tag{12}$$

Based on the above analysis, g^{3}Conv can be generalized to the n-order spatial interaction, i.e. g^{n}Conv. For the input feature map $X \in \mathbb{R}^{H\times W\times C}$, the process is similar to g^{3}Conv, as follows:

$$X' = Conv_{in}\left(X\right) \in \mathbb{R}^{H\times W\times 2C} \tag{13}$$

$$\left[a_0^{H\times W\times C_0}, b_0^{H\times W\times C_0}, \cdots, b_{n-1}^{H\times W\times C_{n-1}}\right] = Split\left(X'\right) \tag{14}$$

Where,

$$C_0 + \sum_{0\leq i\leq n-1} C_i = 2C \tag{15}$$

$$C_i = \frac{C}{2^{n-i-1}}, 0 \leq i \leq n-1 \tag{16}$$

Equation (16) specifies how the channel dimensions are allocated in each order of the g^{n}Conv operation. This allocation is designed to reduce the number of channels used to compute lower orders, thereby avoiding a large computational overhead. After splitting the intermediate feature map X', the gated convolution continues iteratively:

$$a_{i+1} = h_i\left(a_i\right) \otimes DW\_Conv_i\left(b_i\right), i = 0, 1, \cdots, n-1 \tag{17}$$

where,

$$h_i\left(X\right)=\begin{cases} X, & i=0 \\ Conv\left(C_{i-1},C_i\right), & 1\le i\le n-1 \end{cases} \tag{18}$$

The final result for g^{n}Conv is acquired by equation (19), as follows:

$$y=Conv_{out}\left(a_n\right) \tag{19}$$

The proposed HSI-ShipDetectionNet model uses g^{n}Conv to replace the self-attention mechanism in the Transformer encoder to create the High-Order Spatial Interaction (HSI-Former) module. Traditionally, in vision tasks, the Transformer encoder is utilized independently without involving the decoder, so the HSI-Former proposed in this paper can be seen as a visual encoder. This is illustrated in Figure 1. The g^{n}Conv offers several advantages over self-attention, including its ability to extend spatial interactions to higher orders, resulting in improved feature correlation. Moreover, using a convolution-based architecture avoids the quadratic complexity of self-attention, while channel division reduces computational cost. In addition, convolutional operations introduce inductive biases that are helpful for ship detection tasks, such as translation equivariance and locality [49]. In the g^{n}Conv, the depthwise convolution utilizes large 7×7 convolution kernels to increase the receptive field. This improves context modeling and enhances the understanding of advanced semantics.

Analysis of complexities. We will calculate the FLOPs for g^{n}Conv in three parts: linear projection layers, DW-Conv operation and iterative gated convolutions.

- Linear projection layers: The FLOPs of two linear projection layers, Conv$_{in}$, and Conv$_{out}$, can be calculated as follows:

$$\begin{aligned} FLOPs(Conv_{in}) &= 2HWC^2 \\ FLOPs(Conv_{out}) &= HWC^2 \end{aligned} \tag{20}$$

- DW-Conv operation: We denote the kernel size of the DW-Conv as K. The DW-Conv is performed for all $\{b_i\}_{i=0}^{n-1}$, where $b_i\in\mathbb{R}^{H\times W\times C_i}$ and $C_i=\frac{C}{2^{n-i-1}}$. The FLOPs of DW-Conv operation can be calculated as follows:

$$\begin{aligned} FLOPs(DW\_Conv) &= HWK^2\sum_{i=0}^{n-1}\frac{C}{2^{n-i-1}} \\ &= 2HWCK^2(1-\frac{1}{2^n}) \end{aligned} \tag{21}$$

- Iterative gated convolutions (IGC): The FLOPs of iterative gated convolutions can be divided into two components: linear projections $\{h_i\}$ and element-wise multiplication.

$$\begin{aligned} FLOPs(IGC) &= HWC_0+\sum_{i=1}^{n-1}(HWC_{i-1}C_i+HWC_i) \\ &= HWC[2-\frac{1}{2^{n-1}}+\frac{2}{3}C(1-\frac{1}{4^{n-1}})] \end{aligned} \tag{22}$$

The total FLOPs are the sum of these three parts:

$$\begin{aligned} FLOPs(g^nConv) = & HWC[2K^2(1-\frac{1}{2^n})+2-\frac{1}{2^{n-1}}+ \\ & (\frac{11}{3}-\frac{2}{3\times 4^{n-1}})C] \end{aligned} \tag{23}$$

IV. Experiments

A. Experimental Setup

1) Parameter Settings: All experiments in this paper are conducted on a server equipped with NVIDIA Titan V100 GPUs, and the deep learning algorithms are implemented using PyTorch v1.9.0 and Python v3.8.0. During the training process, we set the batch size to 4 and use the SGD optimizer with momentum and weight decay of 0.937 and 5e-4, respectively, and an initial learning rate of 0.01. We stop training after 500 epochs. The other parameters are default values, empirically adopted as done in the original YOLOv5. For all baselines in this paper, rather than searching for the best hyperparameters in the hyperparameter space, we use the same training parameters as those in the corresponding models.

A summary of the settings of the main modules proposed in this paper is shown in Table II. Before being input into the network, the images are cropped to a size of 640 x 640. In the backbone, LHAB modules are applied on layers 5, 6, 11, 12, 13, 15, and 17, respectively, as explained in Section III-C. In the LHAB module, two parameters, γ and b, are set to specific values. γ is set to 2, while b is set to 1. These values are used in the ECA equation. For HSI-Former, the order of the g^{n}Conv operation is set to 3. Additionally, the DW-Conv operation is performed using kernels with a size of 7.

TABLE II
PARAMETER SETTINGS

Setting description	setting or value
Input size	640
Layers using LHAB in the backbone	5, 6, 11, 12, 13, 15, 17
γ in the ECA equation	2
b in the ECA equation	1
Number of the HSI-Former layer	1
The order n in g^{n}Conv	3
The kernel size of the DW-Conv in g^{n}Conv	7

2) Dataset: The dataset used in our experiments is sourced from the Kaggle competition for marine ship detection[1]. The dataset comprises 29GB of high-resolution optical remote sensing images, consisting of a total of 192,556 images in the training set and 15,606 images in the test set. Each image has a resolution of 768×768 pixels. To evaluate the effectiveness of our model in detecting small ships, we randomly select 1000 images from the dataset that contain small target ships and divide them into three subsets:

[1] https://www.kaggle.com/c/airbus-ship-detection

a training set, a validation set, and a test set, with a ratio of 7:2:1.

3) Evaluation Metrics: In order to provide a comprehensive evaluation of our proposed method, we consider not only the standard metrics of **Precision**, **Recall**, and the **mean Average Precision (mAP)**, but also the model size, the number of parameters, the calculated amount and time. These metrics are commonly used in the field of object detection and can provide a clear understanding of the performance of our model in comparison to other state-of-the-art methods.

These metrics are defined as follows:

$$Recall{=}\frac{TP}{TP+FN} \tag{24}$$

$$Precision{=}\frac{TP}{TP+FP} \tag{25}$$

$$mAP{=}\int_0^1 Precision\left(Recall\right)d(Recall) \tag{26}$$

where TP, FP, and FN represent true positive, false positive, and false negative, respectively, and Precision (Recall) refers to the Precision-Recall curve.

B. Comparison with State-of-the-art Methods

To evaluate the performance of our proposed method, we compare it with a total of three types of models: small object detection models, lightweight detection models, and ship detection models.

1) Comparison with Small Object Detection Models: To verify the superior performance of our proposed approach on small object detection, we compare HSI-ShipDetectionNet with two state-of-the-art small object detection models, as described below.

- **TPH-YOLOv5** [20]: This is a YOLOv5-based detector aimed at densely packed small objects. It incorporates advanced techniques such as Transformer blocks, CBAM, and other experienced tricks to improve performance.
- **SPH-YOLOv5** [52]: The original prediction heads of this detector are replaced with Swin Transformer Prediction Heads (SPHs), which can reduce the computational complexity considerably. In addition, Normalization-based Attention Modules (NAMs) are introduced to improve network detection performance.

TABLE III
COMPARISON OF DETECTION PERFORMANCE OF DIFFERENT SMALL OBJECT DETECTION MODELS

Models	Para	GFLOPs	R(%)	mAP(%)	Size(MB)	T(ms)
TPH-YOLOv5	60.35M	145.3	76.85	74.84	116.8	22.91
SPH-YOLOv5	27.81M	272.4	76.30	74.53	54.6	15.73
Ours	**4.15M**	**10.0**	76.85	74.35	**9.2**	**14.33**

As can be seen in Table III, which displays the number of parameters (Para) and recall (R), our proposed HSI-ShipDetectionNet has the smallest number of parameters and computational complexity, requiring only 9.2 MB of storage space. Additionally, the average inference time for each image in the proposed model is reported as 14.33ms. This indicates that the model can perform real-time detection, as it only takes approximately 1.4 seconds to complete the detection of 100 images from the testing set. Although TPH-YOLOv5 achieves a higher mAP value than ours by 0.49, it has 14.5 times more parameters and GFLOPs than our model. Similarly, the detection accuracy of SPH-YOLOv5 is comparable to that of HSI-ShipDetectionNet, but our model requires 85.1% fewer parameters and 96.3% less computational effort. While these two small object detectors have superior detection performance, they are built on deep and dense convolutional layers. In contrast, our proposed model is much lighter and achieves comparable detection accuracy and faster detection speed. Therefore, our method is better suited for real-time tasks and scenarios with limited resources.

2) Comparison with Lightweight Detection Models: To evaluate the performance of our model, we also compare HSI-ShipDetectionNet with the following eight lightweight detection models, described as follows.

- **MobileNetV3-Small** [35]: Based on MobileNetV2, MobileNetV3 added the SE block and improved the activation function using h-swish. The small version is targeted at low-resource use cases and therefore contains fewer bottleneck blocks.
- **PP-LCNet** [53]: This is a lightweight CPU network that utilizes the MKLDNN acceleration strategy. While the techniques used in the network are not novel and have been introduced in previous works, this model achieves a better balance between accuracy and speed through extensive experimentation.
- **ShuffleNetV2** [37]: Four policies were presented by the authors to reduce memory access costs (MAC), avoid network fragmentation, and reduce element-wise operations.
- **MobileNetV3-Large** [35]: Unlike MobileNetV3-Small, the large version is targeted at resource-intensive use cases and therefore contains more bottleneck blocks.
- **GhostNet** [38]: It has developed the Ghost module, which tends to accept abundant and redundant information in the feature maps through a cheap operation instead of discarding it.
- **Efficient-Lite0** [54]: The Efficient-Lite series is the on-device version of EfficientNet and consists of five versions, of which Efficient-Lite0 is the smallest.
- **YOLOv5s**: YOLOv5s is the smallest network in the YOLOv5 series in terms of depth and width.
- **YOLOv3-tiny** [23]: Compared to YOLOv3, YOLOv3-tiny has fewer feature layers and only two prediction branches, making it more suitable for high-speed detection tasks.

In order to ensure consistency in experimental conditions, we incorporated the aforementioned lightweight models (excluding YOLOv5s and YOLOv3-tiny) into the framework

TABLE IV
COMPARISON OF DETECTION PERFORMANCE OF DIFFERENT LIGHTWEIGHT DETECTION MODELS

Models	Para	GFLOPs	R(%)	mAP(%)	Size(MB)	T(ms)
MobileNetV3-S	3.54M	6.3	71.30	68.61	7.2	9.99
PP-LCNet	3.74M	8.1	72.59	70.87	7.4	8.93
ShuffleNetV2	3.78M	7.7	71.30	68.86	7.5	11.66
MobileNetV3-L	5.20M	10.3	72.59	70.52	10.2	12.33
GhostNet	5.20M	8.4	73.89	71.94	10.4	11.73
Efficient-Lite0	5.71M	11.5	71.85	70.08	11.2	10.18
YOLOv5s	7.05M	16.3	75.92	74.25	13.7	8.81
YOLOv3-tiny	8.67M	12.9	75.19	73.28	16.6	3.18
ours	4.15M	10.0	**76.85**	**74.35**	9.2	14.33

of YOLOv5 for the purpose of conducting target detection tasks.

As shown in Table IV, our proposed HSI-ShipDetectionNet achieves the highest recall and mAP. Compared to the second-best performing model, YOLOv5s, our model not only outperforms in terms of mAP but also has a significantly lower number of parameters, GFLOPs, and model size, at 41.1%, 38.7%, and 32.8% less respectively. Similarly, Our model outperforms YOLOv3-tiny with 2.2% higher recall and 1.5% higher mAP, despite having half the parameters and a simpler network structure. This is attributed to the fact that YOLOv3's two prediction branches result in fewer bounding boxes, thus weakening its detection performance. GhostNet and MobileNet-Large have a parameter count 1.05M higher than ours; however, their mAP are lower than ours by 2.41 and 3.83, respectively. On the other hand, ShuffleNetV2, PP-LCNet and MobileNetV3-Small are indeed lighter than our model, but their detection accuracy (mAP) is around 4 to 6 percentage points lower than that of HSI-SmallShipDetectionNet. These models prioritize lower model complexity over detection accuracy, whereas our HSI-ShipDetectionNet effectively balances both. Overall, HSI-ShipDetectionNet is more sensitive to the detection of small ships while maintaining a suitable level of model complexity.

In terms of inference time, our proposed model may not be as dominant compared to other lightweight models. It is worth noting that even though our model has a lower number of parameters and requires less computation than YOLOv5, the inference time of YOLOv5 is still lower. This observation can be attributed to the fact that our model utilizes a large number of depthwise convolutions (DW-Conv). Although DW-Conv has fewer parameters and GFLOPs compared to standard convolutions, they require more intermediate variables to be stored during the computation process. As a result, an amount of time is spent on reading and writing data, leading to slower inference time. Addressing this limitation is an area that can be explored in future work.

3) Comparison with Ship Detection Models: To further evaluate the performance of the proposed HSI-ShipDetectionNet in the field of ship detection, we compare it with two state-of-the-art ship detection models. These models are described as follows:

- **ShipDetectionNet** [2]: This is a lightweight ship detection network that utilizes an improved convolution unit to replace the standard convolution, resulting in a significant reduction in the number of parameters in the network.
- **Literature** [55]: This network proposes a new loss function, IEIOU_LOSS, and introduces the coordinate attention (CA) mechanism to achieve robust detection results for docked and dense ship targets.

TABLE V
COMPARISON OF DETECTION PERFORMANCE OF DIFFERENT SHIP DETECTION MODELS

Models	Para	GFLOPs	R(%)	mAP(%)	Size(MB)	T(ms)
Literature [55]	7.13M	16.4	73.89	71.79	14.0	14.39
ShipDetectionNet	6.05M	15.7	76.11	74.26	12.0	15.32
ours	**4.15M**	**10.0**	**76.85**	**74.35**	**9.2**	**14.33**

In the experiments illustrated in Table V, it can be seen that our proposed model outperforms all the other models in terms of all the evaluation metrics. HSI-ShipDetectionNet has almost 3.6% higher mAP than the network proposed in the literature [55]. Moreover, the number of parameters and GFLOPs of our model is 41.8% and 39.0% lower than that of the network in [55], respectively, indicating that our model consumes less storage space. Compared with ShipDetectionNet, our model has a reduction of 31.4% and 36.3% regarding parameters and GFLOPs, respectively, while achieving comparable detection accuracy. This is due to the new Lightweight Hybrid Attention Block (LHAB) proposed in our model, which replaces the SE attention mechanism used in ShipDetectionNet. In summary, HSI-ShipDetectionNet is more lightweight while having better detection accuracy and faster detection speed.

4) The Result Analysis and Visual Comparisons of Different Methods: We compare our model's detection performance to that of several models of similar complexity and magnitude. The resulting P-R curves are depicted in Figure 6. Notably, the red curve corresponds to our model and is situated prominently toward the upper-right corner of the plot. This result demonstrates the superior performance achieved by our model compared with other models of similar size.

To demonstrate the superior performance of our proposed method for detecting small targets, we present some inference results on the test set in Figure 7. It is evident from the results that HSI-ShipDetectionNet successfully locates and recognizes all small target ships that are missed by GhostNet and YOLOv5s. Although ShipDetectionNet also detects all small ships successfully, the confidence of its prediction box is not as high as that of HSI-ShipDetectionNet. In particular,

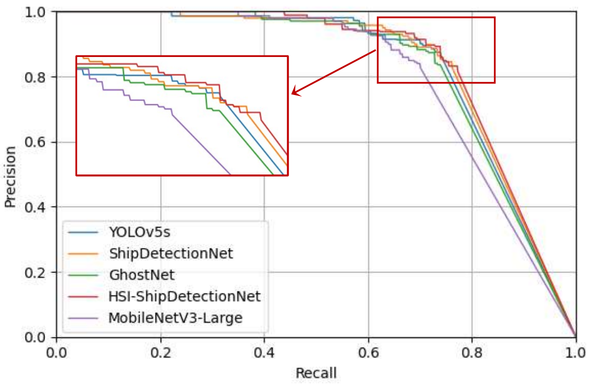


Fig. 6. Performance comparison on the testing set in terms of the P-R curves.

for some images where it is challenging to distinguish the ship from the background, as shown in row (a), HSI-ShipDetectionNet more accurately wraps the target ships. This is due to the fact that HSI-Former can better understand and model advanced features in deep layers, which improves the accuracy of location and regression for prediction boxes. Furthermore, our proposed model can detect small target ships at the edge of the images with relatively high confidence, as shown in rows (c) and (d). Additionally, our model exhibits excellent detection performance in the presence of bad weather conditions, such as cloud barriers shown in row (e).

To summarize, our proposed HSI-ShipDetectionNet demonstrates competency in detecting small ships in challenging sea conditions. This results in more precise and dependable prediction boxes on optical remote sensing images.

C. Ablation Experiments and Sensitivity Analysis

We evaluate the effectiveness of the proposed several modules by ablation analysis and sensitivity analysis.

1) Choosing GhostNet as Our Baseline: To substantiate the superiority of GhostNet over other networks of comparable scale in the context of this study, the weights passing through GhostNet and MobileNetV3-Large are visualized as heat maps using the Grad-CAM [56]. Figure 8 in the paper illustrates these attention maps, providing insights into the model's focal points. MobileNetV3-Large tends to excessively emphasize background information, leading to a detrimental impact on detection performance. In contrast, GhostNet exhibits a more pronounced concentration on small ship targets while mitigating the undue emphasis on background elements. As a result, GhostNet has a higher mAP value, as shown in Table VI. We speculate the possible reasons for this as follows:

The authors of GhostNet found that some of the feature maps generated by the first residual group in ResNet-50 were very similar, indicating that there was abundant and redundant information in the feature maps. Rather than discarding these redundant feature maps, they chose to accept them in a cost-efficient way, which is the ”cheap operation”. Small ships occupy fewer pixel units, making the information about them extremely valuable. Removing redundant information to reduce the complexity of the network is not a good approach for small ship detection. However, GhostNet's approach of embracing redundant information in a cost-effective way is beneficial for small target detection. Therefore, we have selected GhostNet as the backbone of our lightweight ship detector and further simplified it.

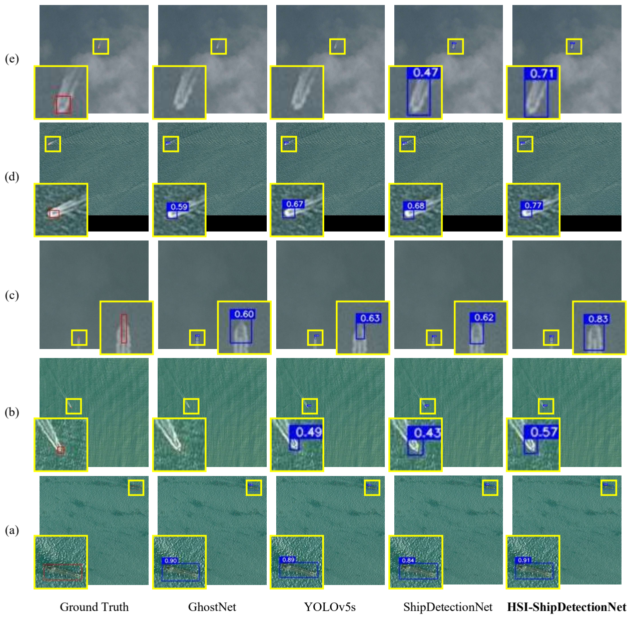


Fig. 7. To visualize the inference results from different detection methods on the test set, we display the outputs of the best-performing model for each method. The methods we comparing are GhostNet, YOLOv5s, ShipDetectionNet, and HSI-ShipDetectionNet.

TABLE VI
COMPARISON OF DETECTION PERFORMANCE BETWEEN MOBILENETV3-LARGE AND GHOSTNET

	mAP	Parameters
MobileNetV3-Large	70.52	5.20M
GhostNet	71.94	5.20M

2) Ablation of the Predictive Branch of Tiny Ships: To study the influence of the predictive branch of tiny ships (P_{tiny}) on detection performance, we first conduct experiments on the detection framework with GhostNet as the backbone. We obtain results for GhostNet on the original detection framework (with only three predictive branches), and then add P_{tiny} on top of it. The results in Table VII show that the introduction of P_{tiny} significantly improves mAP by 1.07. This indicates that adding the P_{tiny} branch can improve the network's detection accuracy.

Moreover, to demonstrate the effectiveness of P_{tiny} in HSI-ShipDetectionNet, we conduct experiments by removing it, and the corresponding results are presented in Table VIII. Notably, the introduction of P_{tiny} leads to a

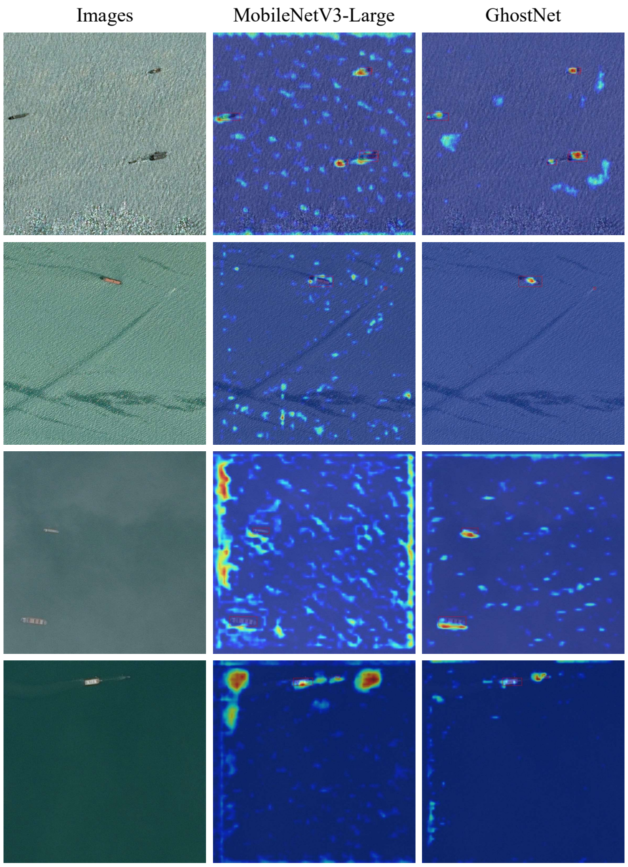


Fig. 8. Visualization of heat maps for MobileNetV3-Large and GhostNet.

TABLE VII
ADDING THE PREDICTIVE BRANCH OF TINY SHIPS TO THE BASELINE MODEL GHOSTNET

	mAP	Parameters
GhostNet	71.94	5.20M
GhostNet w/ P$_{tiny}$	**73.01**	5.34M

significant increase of 1.06 in the model's detection accuracy (mAP), while requiring only a minimal increase of 0.14M parameters. This demonstrates the impact of P$_{tiny}$ on enhancing the model's performance without substantially increasing its complexity.

TABLE VIII
ABLATION OF THE PREDICTIVE BRANCH OF TINY SHIPS

	mAP	Parameters
w/o P$_{tiny}$	73.29	4.01M
ours	**74.35**	4.15M

3) Ablation and Sensitivity Analysis of the High-Order Spatial Interaction Mechanism: Expanding on the detection framework described in the previous part, which already includes the P$_{tiny}$ branch, we now examine the effects of integrating the High-Order Spatial Interaction (HSI-Former) module on detection performance. Table IX presents the results of this analysis, where L denotes the number of HSI-Former layers and n refers to the order of g^{n}Conv.

To investigate the impact of the order on model performance, we conduct experiments with varying n from 1 to 4, where the number of HSI-Former layers is fixed at 1. Our findings indicate that the model performs best when the order is 3, with the mAP value 1.66 higher than that without the HSI-Former module. Conversely, the worst performance is observed when the order is 1, as 1-order spatial interactions are equivalent to plain convolution [50], thus contributing little to model performance. Furthermore, 2-order spatial interactions show a slight improvement in the modeling ability by 0.26, while 4-order spatial interactions yield an improvement of only 0.37 compared to the model without the HSI-Former module. This result suggests that it is not that the higher the order of spatial interaction is, the greater the positive impact on the network will be. Further, we also try the effect of 3-order when the HSI-Former layers are 2. It is interesting to see that in the case where layers are 2 when the order of spatial interaction is 3, the model performance is slightly lower than when the HSI-Former layer is 1. This indicates that too many HSI-Former modules may burden the network.

On the other hand, as the HSI-Former is specifically designed based on the analysis of the Transformer encoder, we conduct a test to evaluate the impact of the Transformer block on the overall network performance. As shown in Table IX, we observe that the size of the model with the Transformer module is comparable to that of the model with HSI-Former(L=1, n=3). However, the mAP value decreases by 1.04, indicating that 3-order spatial interactions have more potential for learning and modeling context when compared to 2-order spatial interactions. This finding strongly suggests that the HSI-Former architecture with higher order spatial interactions has superior performance in capturing and modeling context for the given task.

TABLE IX
ABLATION AND SENSITIVITY ANALYSIS OF THE HIGH-ORDER SPATIAL INTERACTION MECHANISM

	mAP	Parameters
GhostNet w/ P$_{tiny}$	73.01	5.339M
w/ HSI-Former(L=1, n=1)	72.70	5.631M
w/ HSI-Former(L=1, n=2)	73.27	5.647M
w/ HSI-Former(L=1, n=3)	**74.67**	5.653M
w/ HSI-Former(L=1, n=4)	73.38	5.655M
w/ HSI-Former(L=2, n=3)	73.92	5.967M
w/ Transformer(L=1)	73.63	5.493M

In addition, we conduct experiments to determine the most appropriate size of the convolution kernel in depthwise convolution (DW-Conv) within g^{n}Conv. We test kernel sizes of 3, 5, 7, and 9, measuring the corresponding mAP values, as presented in Figure 9. The results indicate that our model achieves the highest mAP value of 74.35 when using 7 × 7 convolutional kernels. This suggests that larger 7 × 7 kernels applied to the deep layers of the network are more effective in understanding advanced features. One of our initial hypotheses is that a larger kernel would improve context modeling and, consequently, detection accuracy. However, the experimental results show that 9 × 9 kernels do not lead to the desired effect. This implies that although large kernels can facilitate context modeling, there seems to be an upper limit based on the specific task. Theoretically, a smaller kernel corresponds to a smaller receptive field and is more suitable for detecting small targets. However, using 3 × 3 kernels in DW-Conv does not yield the desired results. We attribute this to the fact that the g^{n}Conv module, located at the back of the backbone network, is primarily responsible for understanding high-level features. As we have already extracted feature information of the small targets in advance by increasing P$_{tiny}$ in the shallow layers of the backbone, the size of the convolution kernel in the DW-Conv might not correlate well with small targets. This analysis also suggests that smaller kernels are not as effective as larger ones in understanding deeper features.

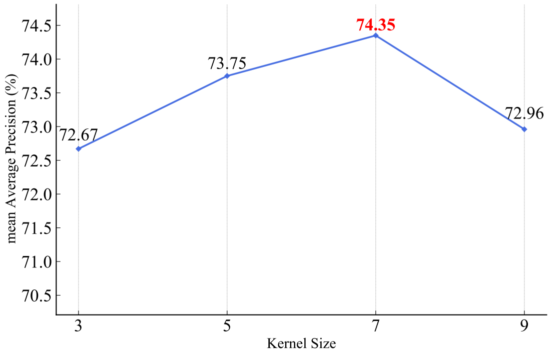


Fig. 9. The effect of convolution kernel size in DW-Conv within g^{n}Conv on mean Average Precision (mAP) values.

4) *Ablation of the Lightweight Hybrid Attention Block:*

To simplify the network further, we design the Lightweight Hybrid Attention Block (LHAB). Our design thinking for the Channel Attention Block (CAB) in LHAB is demonstrated through ablation experiments, and the results are presented in Table X. Here, ECA denotes the original ECA module, where only an average-pooling operation is employed. ECA$_{update1}$ implies that both the max-pooling operation and average-pooling operation are utilized in the ECA module, and the parameters of both operations are shared. Referring to Table X, the inclusion of both operations in a network enhances the mAP compared to the average-pooling operation alone. And then, ECA$_{update2}$ indicates that the parameters of these two operations are not shared, which is our proposed CAB. Since max-pooled and average-pooled features have distinct convolutions, the mAP value is increased by another 0.51 and the optimal results are achieved for both Precision and Recall values. Although this increases the network's parameter count, the use of one-dimensional convolutions for feature extraction means that only 31 (4149741-4149711 = 31) parameters are added, which is insignificant.

Building upon CAB, we introduce an independent Spatial Attention Block (SAB), which does not share parameters, to create LHAB. In Table XI, compared with no LHAB, the LHAB enhances the mAP by 1.59, but does not introduce a huge number of parameters. The LHAB and CBAM share certain similarities as they both incorporate channel attention and spatial attention mechanisms. However, LHAB stands out as a more lightweight option compared to CBAM, resulting in a reduction of 0.37 (4.52 - 4.15 = 0.37) million parameters. This reduction is primarily achieved through the utilization of one-dimensional convolution in LHAB. Additionally, the performance evaluation indicates that LHAB outperforms CBAM with a 0.68 higher mAP value. On the other hand, compared with SE attention, LHAB reduces the parameter count by 1.50 (5.65-4.15 = 1.50) million presented in Table IX. These experimental results demonstrate the superiority of the proposed LHAB.

TABLE X
ABLATION OF THE CHANNEL ATTENTION BLOCK (CAB)

	mAP	Precision(%)	Recall(%)	Parameters
w/ ECA	73.55	79.47	75.93	4149711
w/ ECA$_{update1}$	73.68	78.84	76.48	4149711
w/ ECA$_{update2}$ **(CAB)**	**74.19**	**79.99**	**76.85**	4149742

TABLE XI
ABLATION OF THE LIGHTWEIGHT HYBRID ATTENTION BLOCK (LHAB)

	mAP	Precision(%)	Recall(%)	Parameters
w/o LHAB	72.76	79.34	75.37	4149680
w/ CBAM	73.67	78.53	76.48	4524718
w/ CAB+SAB **(LHAB)**	**74.35**	**80.43**	**76.85**	4150428

In addition, our analysis extends to the examination of the kernel size in LHAB and its impact on network performance, particularly concerning the fifth part of the backbone, where the feature map is set at 20 × 20. As depicted in Figure 10, the evaluation of various kernel sizes applied to part 5 reveals that the 7 × 7 kernel not only attains the highest mAP but also facilitates the swiftest convergence of the model, which reaches the optimal detection accuracy after only 412 epoches. These findings affirm the superiority of the 7 × 7 kernel. Even when applied to smaller feature maps, it consistently maintains excellent performance. Our analysis suggests the following reasons for its superiority:

With the emergence of Vision Transformers (ViTs), CNNs face challenges in various visual tasks. The efficacy of ViTs is attributed to their multi-head self-attention (MHSA) mechanism, enabling the modeling of long-range dependencies and facilitating information gathering from expansive regions. This prompts the question of whether employing large kernels in traditional CNNs could foster more diverse interactions between spatial locations, akin to the capabilities of ViTs. Indeed, several studies have extensively explored the use of large kernels. For instance, literature [57] has extended kernel sizes to 31×31, demonstrating that employing a few large kernels, as opposed to numerous small ones, can enhance the effectiveness of CNNs, particularly in downstream tasks. Additionally, literature [58] introduced sparsity to further expand kernel sizes beyond 51×51, resulting in improved performance. Consequently, the incorporation of large-size kernels plays a pivotal role in global modeling.

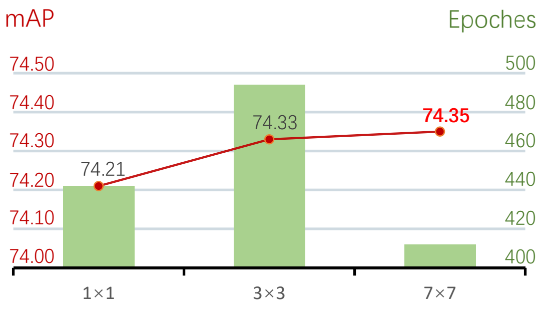


Fig. 10. The impact of varying convolutional kernel sizes on the network's performance within the part 5 of the backbone.

D. Tests on the FAIR1M Dataset

To evaluate the performance of our proposed model, we conduct a comparative study using four models of the same size level on a new dataset called FAIR1M [59]. By conducting experiments on the FAIR1M dataset of ship-containing images, we aim to assess how well our proposed model performs, particularly on dense ships in remote sensing images. The images in FAIR1M vary in size, ranging from 1000×1000 to 10000×10000 pixels. The dataset comprises 5 categories and 37 subcategories. For our specific task, we focus on the category of ships and select 1000 ship-containing images to create our dataset. We randomly divide this dataset into training, validation, and test sets in a ratio of 7:2:1. During the training process, each network has a batch size of 4, and we train them for 500 epochs while keeping the default values of the hyperparameters of the original network.

The experimental results are illustrated in Figure 11. It is evident that the presence of a significant number of small and dense ships in the FAIR1M dataset poses challenges, leading to a decrease in overall detection accuracy compared to the Kaggle dataset, regardless of the model used. However, our proposed model stands out significantly, achieving the highest detection accuracy while requiring the lowest number of parameters. Its mean Average Precision (mAP) value is even on par with YOLOv5, which has the highest number of parameters among the compared models. This outcome highlights the remarkable trade-off achieved by our model between accuracy and complexity.

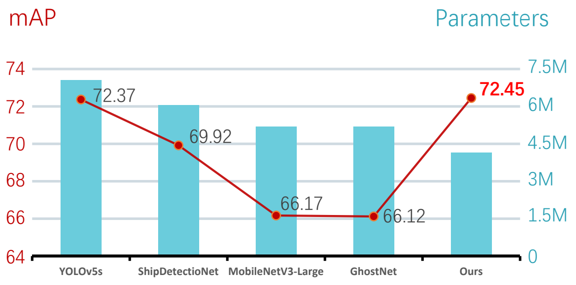


Fig. 11. The detection results on the FAIR1M dataset of ship-containing images. Our proposed model achieves the highest detection accuracy while requiring the lowest number of parameters.

Figure 12 visualizes the detection results of HSI-ShipDetectionNet on dense ships using the FAIR1M dataset. The red box represents the ground truth, while the lower image shows the prediction box generated by our model, indicated in yellow. The results demonstrate that our model successfully localizes most of the ships, even those of smaller sizes, as exemplified by the two smaller ships at the bottom in Figure 12 (b). However, we acknowledge that there is still room for improvement in our network. In particular, for cases where two ships are situated side by side, as shown in Figure 12 (a), our network encounters a missed detection. In Figure 12 (b), there are also misdetections, such as identifying the upper-right dock as a ship. We attribute this situation to the scarcity of dense dataset instances, making it challenging for the model to fit these occurrences during training.

Figure 13 portrays the divergent focus of attention between our model and the baseline, GhostNet, through heat maps. In line (a), GhostNet's undue emphasis on the inconsequential background is conspicuous. Contrastingly, in line (b), our model astutely attends to the small ships at the lower part, exhibiting a perceptible advantage. Likewise, in line (c), it can be observed that our model focuses more adequately on the smaller ships. It can be seen that in more complex scenarios, our model demonstrates a stronger ability to focus on small targets than the baseline GhostNet.

In summary, our model excels in achieving the highest detection accuracy while maintaining a more lightweight architecture compared to models of the same size. With further study to address the challenges presented by dense objects, we believe our model has the potential to further enhance its performance and robustness in remote sensing target detection applications.

V. Conclusion

In this paper, we present a novel ship detection framework called HSI-ShipDetectionNet, which aims to address the challenges of accurate and efficient detection of small ships on resource-limited platforms. To make full use of the

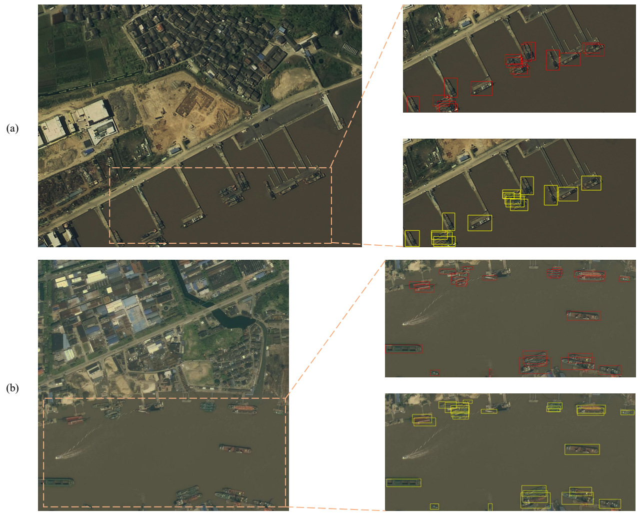


Fig. 12. Visualisation of the detection of the dense ships. The red box represents the ground truth, while the lower image shows the prediction box generated by our model, indicated in yellow.

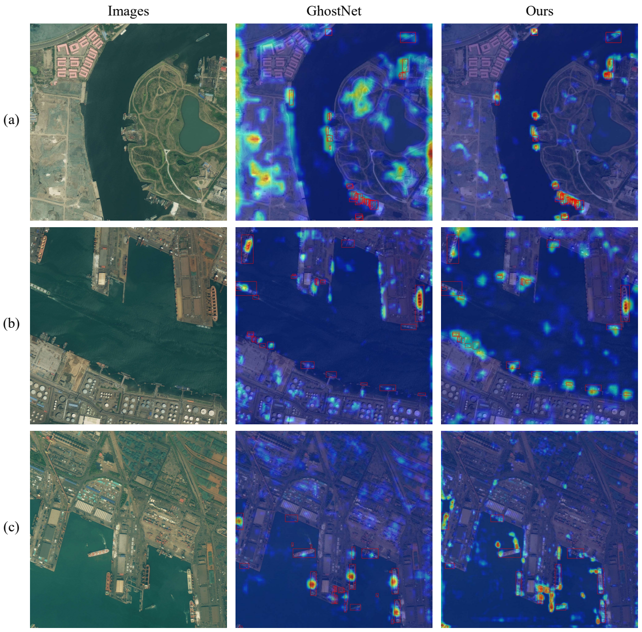


Fig. 13. Visualization of heat maps for our model and the baseline, GhostNet.

shallow features of the network, we introduce a predictive branch specifically designed for tiny ships. To reduce the network's complexity without compromising detection performance, we propose a Lightweight Hybrid Attention Block (LHAB). To further enhance the network's ability to capture advanced features in deep layers, we introduce the high-order spatial interaction (HSI-Former) module. We conduct comprehensive evaluations of the proposed model, including comparison experiments and ablation studies. The results demonstrate the superiority and effectiveness of our proposed model.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China Under Grant 62020106004, Grant 62272342, and Grant 62306212.

REFERENCES

[1] K. Eldhuset, "An automatic ship and ship wake detection system for spaceborne sar images in coastal regions," *IEEE transactions on Geoscience and Remote Sensing*, vol. 34, no. 4, pp. 1010–1019, 1996.

[2] Y. Yin, X. Cheng, F. Shi, M. Zhao, G. Li, and S. Chen, "An enhanced lightweight convolutional neural network for ship detection in maritime surveillance system," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 5811–5825, 2022.

[3] U. Kanjir, H. Greidanus, and K. Oštir, "Vessel detection and classification from spaceborne optical images: A literature survey," *Remote sensing of environment*, vol. 207, pp. 1–26, 2018.

[4] L. Tian, Y. Cao, B. He, Y. Zhang, C. He, and D. Li, "Image enhancement driven by object characteristics and dense feature reuse network for ship target detection in remote sensing imagery," *Remote Sensing*, vol. 13, no. 7, p. 1327, 2021.

[5] C. Chen, M.-Y. Liu, O. Tuzel, and J. Xiao, "R-cnn for small object detection," in *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13*. Springer, 2017, pp. 214–230.

[6] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[7] L. Bo, X. Xiaoyang, W. Xingxing, and T. Wenting, "Ship detection and classification from optical remote sensing images: A survey," *Chinese Journal of Aeronautics*, vol. 34, no. 3, pp. 145–163, 2021.

[8] K. H. Pegler, D. J. Coleman, Y. Zhang, and R. P. Pelot, "The potential for using very high spatial resolution imagery for marine search and rescue surveillance," *GeoCarto International*, vol. 18, no. 3, pp. 35–39, 2003.

[9] U. Kanjir, A. Marsetič, P. Pehani, and K. Oštir, "An automatic procedure for small vessel detection from very-high resolution optical imagery," *Proc. 5th GEOBIA*, pp. 1–4, 2014.

[10] F. Xu and J.-h. Liu, "Ship detection and extraction using visual saliency and histogram of oriented gradient," *Optoelectronics Letters*, vol. 12, no. 6, pp. 473–477, 2016.

[11] M. U. Selvi and S. S. Kumar, "A novel approach for ship recognition using shape and texture," *International Journal of Advanced Information Technology (IJAIT) Vol*, vol. 1, 2011.

[12] Y. Xia, S. Wan, and L. Yue, "A novel algorithm for ship detection based on dynamic fusion model of multi-feature and support vector machine," in *2011 Sixth International Conference on Image and Graphics*. IEEE, 2011, pp. 521–526.

[13] K. Tong, Y. Wu, and F. Zhou, "Recent advances in small object detection based on deep learning: A review," *Image and Vision Computing*, vol. 97, p. 103910, 2020.

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[15] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[16] S. Zagoruyko, A. Lerer, T.-Y. Lin, P. O. Pinheiro, S. Gross, S. Chintala, and P. Dollár, "A multipath network for object detection," *arXiv preprint arXiv:1604.02135*, 2016.

[17] L. Guan, Y. Wu, and J. Zhao, "Scan: Semantic context aware network for accurate small object detection," *International Journal of Computational Intelligence Systems*, vol. 11, no. 1, pp. 951–961, 2018.

[18] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy *et al.*, "Deepid-net: Deformable deep convolutional neural networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2403–2412.

[19] Y. Zhu, R. Urtasun, R. Salakhutdinov, and S. Fidler, "segdeepm: Exploiting segmentation and context in deep neural networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4703–4711.

[20] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2778–2788.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[22] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[23] ——, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[24] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[25] X. Xu, X. Zhang, and T. Zhang, "Lite-yolov5: A lightweight deep learning detector for on-board ship detection in large-scene sentinel-1 sar images," *Remote Sensing*, vol. 14, no. 4, p. 1018, 2022.

[26] P. Xu, Q. Li, B. Zhang, F. Wu, K. Zhao, X. Du, C. Yang, and R. Zhong, "On-board real-time ship detection in hisea-1 sar images based on cfar and lightweight deep learning," *Remote Sensing*, vol. 13, no. 10, p. 1995, 2021.

[27] C. Hu, C. Chen, C. He, H. Pei, and J. Zhang, "Sar detection for small target ship based on deep convolutional neural network," *Journal of Chinese inertial technology*, vol. 27, no. 3, pp. 397–406, 2019.

[28] Y. Wu, W. Ma, M. Gong, Z. Bai, W. Zhao, Q. Guo, X. Chen, and Q. Miao, "A coarse-to-fine network for ship detection in optical remote sensing images," *Remote Sensing*, vol. 12, no. 2, p. 246, 2020.

[29] X. Xie, L. Li, Z. An, G. Lu, and Z. Zhou, "Small ship detection based on hybrid anchor structure and feature super-resolution," *Remote Sensing*, vol. 14, no. 15, p. 3530, 2022.

[30] S. Wang, S. Gao, L. Zhou, R. Liu, H. Zhang, J. Liu, Y. Jia, and J. Qian, "Yolo-sd: Small ship detection in sar images by multi-scale convolution and feature transformer module," *Remote Sensing*, vol. 14, no. 20, p. 5268, 2022.

[31] K. Jin, Y. Chen, B. Xu, J. Yin, X. Wang, and J. Yang, "A patch-to-pixel convolutional neural network for small ship detection with polsar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 9, pp. 6623–6638, 2020.

[32] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[33] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[34] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[35] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[36] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.

[37] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 116–131.

[38] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: More features from cheap operations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1580–1589.

[39] Z. Li, L. Zhao, X. Han, M. Pan, and F.-J. Hwang, "Lightweight ship detection methods based on yolov3 and densenet," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–10, 2020.

[40] J. Jiang, X. Fu, R. Qin, X. Wang, and Z. Ma, "High-speed lightweight ship detection algorithm based on yolo-v4 for three-channels rgb sar image," *Remote Sensing*, vol. 13, no. 10, p. 1909, 2021.

[41] S. Liu, W. Kong, X. Chen, M. Xu, M. Yasir, L. Zhao, and J. Li, "Multi-scale ship detection algorithm based on a lightweight neural network for spaceborne sar images," *Remote Sensing*, vol. 14, no. 5, p. 1149, 2022.

[42] J.-C. Zheng, S.-D. Sun, and S.-J. Zhao, "Fast ship detection based on lightweight yolov5 network," *IET Image Processing*, vol. 16, no. 6, pp. 1585–1593, 2022.

[43] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.

[44] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[45] Q. Wang, B. Wu, P. Zhu, P. Li, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[46] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 603–612.

[47] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[49] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[50] Y. Rao, W. Zhao, Y. Tang, J. Zhou, S.-N. Lim, and J. Lu, "Hornet: Efficient high-order spatial interactions with recursive gated convolutions," *arXiv preprint arXiv:2207.14284*, 2022.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[52] H. Gong, T. Mu, Q. Li, H. Dai, C. Li, Z. He, W. Wang, F. Han, A. Tuniyazi, H. Li *et al.*, "Swin-transformer-enabled yolov5 with attention mechanism for small object detection on satellite images," *Remote Sensing*, vol. 14, no. 12, p. 2861, 2022.

[53] C. Cui, T. Gao, S. Wei, Y. Du, R. Guo, S. Dong, B. Lu, Y. Zhou, X. Lv, Q. Liu *et al.*, "Pp-lcnet: A lightweight cpu convolutional neural network," *arXiv preprint arXiv:2109.15099*, 2021.

[54] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[55] X. Tan and Penghui, "Improve yolov5's ship target detection in sar image," *Computer Engineering and Applications*, vol. 58, no. 4, pp. 247–254, 2022.

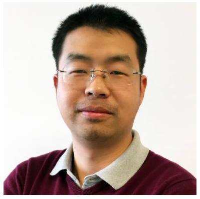
[56] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[57] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31x31: Revisiting large kernel design in cnns," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 963–11 975.

[58] S. Liu, T. Chen, X. Chen, X. Chen, Q. Xiao, B. Wu, T. Kärkkäinen, M. Pechenizkiy, D. Mocanu, and Z. Wang, "More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity," *arXiv preprint arXiv:2207.03620*, 2022.

[59] X. Sun, P. Wang, Z. Yan, F. Xu, R. Wang, W. Diao, J. Chen, J. Li, Y. Feng, T. Xu *et al.*, "Fair1m: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 184, pp. 116–130, 2022.

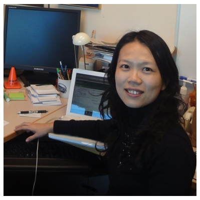
Yifan Yin received B.S. and M.S. degrees in Computer Science and Technology from Tianjin University of Technology, Tianjin, China. She's going to study for a PhD at the University of Technology Sydney. Her current research interests include artificial intelligence, deep learning, computer vision, and object detection.



Xu Cheng (Member, IEEE) received his Ph.D. degree in Engineering from the Department of Ocean Operations and Civil Engineering, Intelligent Systems Laboratory, Norwegian University of Science and Technology (NTNU), Ålesund, Norway, in June 2020. From June 2020 to March 2022, he worked as a Postdoctoral fellow, and researcher at the Department of Manufacturing and Civil Engineering, Gjøvik, Norway. From April 2022 to June 2023, he worked at Smart Innovation Norway as a permanent researcher. Now, he is a full professor at the Tianjin University of Technology. He has applied for and coordinated more than 5 projects supported by the Norwegian Research Council (NFR), the EU, and industry. He has published more than 60 papers as first and co-author in his research interests, including data analysis and artificial intelligence in maritime operations, time series analysis, and predictive maintenance of wind turbines.

Fan Shi (Member, IEEE) is a Professor at the School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China. Dr. Shi received his Ph.D. degree from Nankai University, Tianjin, China, in 2012. From June 2018 to August 2019, he was a research scholar in West Virginia University. His research interests include machine vision, pattern recognition and optics.

Xiufeng Liu received the Ph.D. degree in computer science from Aalborg University, Denmark, in 2012. He was a post-doctoral researcher at the University of Waterloo and a research scientist at IBM, Canada, from 2013 to 2014. He is currently a senior researcher at the Department of Technology, Management and Economics at the Technical University of Denmark. His research interests include smart meter data analysis, data warehousing, energy informatics, and big data.



Huan Huo received the B.Eng and Ph.D. degrees in Computer Science and Technology from Northeastern University, China in 2002 and 2007, respectively. Dr. Huan HUO taught at the Department of Computer Information System, the University of the Fraser Valley in Canada, and did collaborative research in the University of Waterloo as a visiting scholar for one year. Since 2018, she has been a senior lecturer in the School of Computer Science at the University of Technology Sydney, Australia. Her research interests include data stream management technology, advanced data analysis, and data-driven cybersecurity.

Shengyong Chen (Senior Member, IEEE) is a full professor at Tianjin University of Technology and the director of the Engineering Research Center of Learning-Based Intelligent System (Ministry of Education). He has been conducting research on vision sensors for robotics for more than 20 years. He obtained the Ph.D. degree in computer vision from City University of Hong Kong. From 2006 to 2007, he received a fellowship from the Alexander von Humboldt Foundation of Germany and worked at University of Hamburg, Germany. From 2008 to 2012, he worked as a visiting professor at Imperial College London and University of Cambridge, U.K. He has published over 300 scientific papers in international journals and conferences, including 80 papers in IEEE Transactions. He also published 10+ books in the past years and applied 100+ patents. He received the National Outstanding Youth Foundation Award of NSFC. He organized about 20 international conferences and serves as associate editors of 3 international journals, e.g. IEEE Transactions on Cybernetics.