Journal of *Marine Science and Engineering*

MDPI

Article

FSN-YOLO: Nearshore Vessel Detection via Fusing Receptive-Field Attention and Lightweight Network

Na Du [1], Qing Feng [2], Qichuang Liu [3], Hui Li [3,*] and Shikai Guo [3,4]

[1] Navigation College, Dalian Maritime University, Dalian 116026, China; duna@dlmu.edu.cn
[2] School of Maritime Economics and Management, Dalian Maritime University, Dalian 116026, China; qing2220211556@dlmu.edu.cn
[3] School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China; liuqichuang@dlmu.edu.cn (Q.L.); shikai.guo@dlmu.edu.cn (S.G.)
[4] Dalian Key Laboratory of Artificial Intelligence, Dalian 116024, China
* Correspondence: li_hui@dlmu.edu.cn; Tel.: +86-139-4090-6560

Abstract: Vessel detection is critical for ensuring maritime transportation and navigational safety, creating a pressing need for detection methodologies that are more efficient, precise, and intelligent in the maritime domain. Nonetheless, accurately detecting vessels across multiple scales remains challenging due to the diversity in vessel types and locations, similarities in ship hull shapes, and disturbances from complex environmental conditions. To address these issues, we introduce an innovative FSN-YOLO framework that utilizes enhanced YOLOv8 with multi-layer attention feature fusion. Specifically, FSN-YOLO employs the backbone structure of FasterNet, enriching feature representations through super-resolution processing with a lightweight Convolutional Neural Network (CNN), thereby achieving a balance between processing speed and model size without compromising accuracy. Furthermore, FSN-YOLO uses the Receptive-Field Attention (RFA) mechanism to adaptively fine-tune the feature responses between channels, significantly boosting the network's capacity to capture critical information and, in turn, improve the model's overall performance and enrich the discriminative feature representation of ships. Experimental validation on the Seaship7000 dataset showed that, compared to the baseline YOLOv8l approach, FSN-YOLO considerably increased accuracy, recall rates, and mAP@0.5:0.95 by absolute margins of 0.82%, 1.54%, and 1.56%, respectively, positioning it at the forefront of current state-of-the-art models.

Keywords: targeted detection of sea vessels; multi-scale vessel representation; FSN-YOLO; YOLOv8; receptive-field attention; integrated feature fusion

check for updates

Citation: Du, N.; Feng, Q.; Liu, Q.; Li, H.; Guo, S.. FSN-YOLO: Nearshore Vessel Detection via Fusing Receptive-Field Attention and Lightweight Network. *J. Mar. Sci. Eng.* **2024**, *12*, 871. https://doi.org/10.3390/jmse12060871

Academic Editor: Leszek Chybowski

Received: 13 April 2024
Revised: 14 May 2024
Accepted: 15 May 2024
Published: 24 May 2024

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

1. Introduction

The relentless pursuit of higher speeds and intelligent advancements embodies the inevitable direction of modern ship development. Accompanying the surge in the number of ships and the increased density of maritime traffic, incidents such as ship collisions and related maritime disasters are occurring with alarming frequency. Vessel detection occupies an indispensable central role in various sectors, including marine transportation, maritime surveillance, and port management. Timely, accurate, and rapid monitoring of vessels in the vicinity of coastlines can markedly diminish the incidence of maritime accidents, thereby elevating the safety of navigation and the effectiveness of port administration. In recent years, surveillance camera systems have been efficiently utilized in the marine transportation sector due to their cost-effectiveness, ease of installation, and real-time imaging capabilities. These systems provide timely acquisition of granular location and categorical data of marine vessels—an advantage over the more delayed remote image capturing by satellites—vastly improving intelligent dispatching operations at ports.

Nonetheless, influenced by factors such as waves and severe weather conditions, previous methods for detecting vessels within the ever-changing marine environment still

struggle to strike an optimal balance between model precision, speed, and parameterization. Moreover, the issue is exacerbated when vessels of varying sizes are present in a single scene or when occlusion among ships occurs. Current detection models often prioritize larger vessels, leading to missed or incorrect detections of smaller craft. Thus, cultivating an efficient and precise vessel detection method is of significance in terms of enhancing maritime safety and navigation efficiency.

Traditional vessel detection algorithms typically comprise the following three principal steps: generation of ship candidate regions, artificial extraction of features based on ship scale and shape, and classifier-based categorization and regression. The process of generating candidate regions frequently employs exhaustive search methods across the entire image, which are time-consuming and involve high spatial complexity. Manual feature extraction and classifier regression based on the dimensions and form of the ships are commonly hindered by low precision and limited generalizability. However, with the advancement of deep learning, methods based on convolutional neural networks (CNNs) have quickly achieved prominence in the field of object detection. CNN-based ship detection algorithms broadly fall into the following two categories: two-stage and one-stage methods. The two-stage detection process involves region generation to obtain a preselected box, followed by sample classification, regression via CNNs, and boundary positioning. Noteworthy among these methods are R-CNN [1], Fast R-CNN [2], and Faster R-CNN [3]. Wei et al. [4] achieved good results in ship precision and recognition rate based using a Faster R-CNN network, innovatively using an RPN to directly extract candidate regions and integrating it into the overall network, which improved detection speed and resolved the drawbacks of traditional RCNN inputting candidate regions into CNNs separately. Nonetheless, despite these advancements, two-stage algorithms like Faster R-CNN still fall short in meeting the demands of real-time detection due to their speed limitations. Single-stage object detection methods, on the other hand, leverage backbone feature extraction networks to directly localize and classify targets. Notable examples of detection methods include YOLO [5], SSD [6], and CenterNet [7]. While these approaches are fast, they tend to have higher rates of false positives and missed detections compared to two-stage detection methods.

Amidst the swift expansion of deep learning technology in recent years, image-based target detection algorithms have incrementally improved through the persistent endeavors of scholars worldwide. According to whether the model first generates candidate boxes for subsequent detection, detection models are divided into one-stage detectors and two-stage detectors. The YOLO-series models and SSD are classic single-stage detectors, while Faster R-CNN and Cascade R-CNN [8] are common two-stage detectors. In two-stage detectors, such as Faster-RCNN, the basic features are first extracted by backbone networks like VGG [9] and ResNet [10]. Then, a Region Proposal Network (RPN) [3] crafts Region-of-Interest (ROI) suggestions based on pre-established anchors, scales the proposed features to a constant dimensionality, and furthers them to classification and normalization. Finally, the detection results are obtained through a Non-Maximum Suppression (NMS) operation.

The design excellence in feature representation modules permits two-stage detectors to attain heightened levels of detection precision, hence their selection as preferred methods for early CNN-based object detection. However, for smaller targets, it is difficult to match the size of candidate regions with the size of the target, which often leads to the generation of too many or too few candidate regions, thereby reducing the accuracy of the detection results. Different resolutions of feature maps are employed in the Feature Pyramid Network (FPN) [11] to detect objects of assorted sizes. By allying each pixel on the feature maps with specific anchors, one-stage detectors are realized in YOLO and SSD models, which expedites detection velocity. Wang et al. [12] unveiled a feature fusion module predicated on SSD, condensing the model to bolster recognition finesse and swiftness for minuscule targets. Pang et al. [13] proposed the Libra R-CNN framework for target detection, which integrates three components, namely equilibrium sampling, equilibrium feature pyramids, and equilibrium L1 loss, which solve the problem of imbalance between the feature and

target levels. Lim et al. [14] proposed a target detection algorithm predicated on context and attention mechanisms, augmenting focus on the diminutive targets within acquisition imagery and integrating context information from object strata, which, in specific instances, betters the detection performance of small targets. Furthermore, Wang et al. [15] improved the original neck structure in YOLOv5, adopting a weighted bidirectional feature pyramid network from top to bottom and from bottom to top, enhancing the feature extraction capability and solving the problem of large target-scale changes in the dataset.

Zhu et al. [16] extended the YOLOv5 framework by substituting the conventional prediction head with a transformer-based prediction head, thus constructing the TPH-YOLOv5 model to achieve precise target localization in dense object scenarios. However, the incorporation of TPH introduced a substantial increase in parameters, which impacted the computational speed of the network. Aiming to accelerate the detection speeds of TPH-YOLOv5, Zhao et al. [17] engineered a cross-layer asymmetric transformer (CA-Trans) to replace the additional prediction head while retaining its knowledge. By harnessing a Sparse Local Attention (SLA) module, the asymmetric information between the supplemental head and other heads is captured effectively, enriching the feature representation of other heads, as detailed in "TPH-YOLOv5++". Wang et al. [18] proposed a drone target detection algorithm based on an enhanced version of YOLOv8 that embeds a Small Target Detection (STD) structure within the network, serving as a conduit between shallow and deep features to augment the collection of semantic information for small targets, thereby improving detection accuracy. Additionally, Shen et al. [19] introduced the Deformable Convolution C2f (DCN_C2f) module into the backbone network of YOLOv8, allowing for the adaptive adjustment of the network's receptive field. This enhancement overcomes the limitations inherent in the YOLO backbone network, such as restricted receptive fields due to fixed convolutional kernels and insufficient multi-scale feature learning capabilities, which result from a spatial and channel attention fusion mechanism that cannot adapt to the varying distribution of input data features.

In the complex and changeable marine environment, several researchers have made notable contributions to the detection of sea-going and nearshore vessels. Liu et al.[20] devised an Anchor-guided Attention Refinement Network (AARN) that effectuates accurate and swift detection of vessels on the high seas and in coastal regions. AARN alleviates the complexity of the ship's position and attitude, as well as issues related to background interference, by prominently featuring an Attention Feature Filtering Module (AFFM) and an Anchor-guided Alignment Detection Module (AADM). AFFM exploits attention supervision generated from high-level semantic features during the construction of a four-level feature pyramid to highlight information-rich target hints and suppress background distractions. In AADM, anchor-aligned features are used to definitively identify potential sea-going and nearshore vessels, which not only alleviates misalignment between precision anchors and pyramid features but also further heightens performance. Zhou et al. [21] advanced an improved real-time detection method based on YOLOv5 by incorporating a Collaborative Attention (CA) mechanism within the network architecture, granting the model enhanced precision in locating and recognizing the target areas. To better extract and fusevessel feature information, the original backbone network of YOLOv8 was replaced with a lightweight FasterNet architecture, and the neck module was optimized with an RFA mechanism in place of the original C2f module. This allows for more effective capture of ship features at various scales, enhancing the detection capability for small and distant vessels. Moreover, using an attention mechanism also improves the model's precision in locating and identifying targets in ship images with complex backgrounds or obstructions.

Building on current deep learning-based object detection frameworks, numerous advanced ship detection algorithms have been developed by researchers in recent years to tackle the challenges of maritime vessel detection. To address the intricacies of marine environments, Zwemer et al. [22] developed a real-time ship detection and tracking system using port surveillance cameras, training a Single Shot Detector (SSD) to detect targets by detecting ship scales and aspect ratio features. Hu et al. [23] proposed an enhanced

SSD detection algorithm by substituting the original VGG16 backbone with Resnet50 and incorporating the CBAM attention mechanism to reinforce high-level semantic information, which has improved ship detection accuracy to some extent. Shao et al. [24] introduced a saliency-aware CNN framework based on the YOLOv2 pipeline that utilizes CNNs to first predict the category and position of ships, followed by saliency detection, increasing the accuracy and robustness of ship detection under complex coastal surveillance conditions. Li et al. [25] suggested a novel object detection approach for images captured by Unmanned Surface Vessels (USVs) that fuses DenseNet with YOLOv3 to minimize feature loss, thereby enhancing the stability of ship detection in real marine environments. Chen et al. [26] proposed a complex-scene multi-scale ship detection model based on YOLOv7 that combines spatial pyramid pooling and a shuffle attention mechanism, allowing the model to focus on important information while ignoring irrelevant information, reducing the loss of ship features, thereby improving detection accuracy and the ability to detect multi-scale targets.

However, in both domestic and international research on maritime ship detection, these methods still face challenges when identifying a large number of multi-scale vessels against complex marine backgrounds. In practical applications, ship detection technology requires the identification and tracking of various targets at long distances and in multiple scenarios. Among them, ship targets often have small imaging and indistinct features and are easily confused with other targets, in addition to existing problems such as false detection, missed detection, and low accuracy.

As the YOLO network has continued to evolve, Ultralytics released the latest iteration, YOLOv8, in January 2023 as part of a new series of models. In comparison to its predecessors, particularly versions v5, v6, and v7, YOLOv8 represents significant improvements in both speed and accuracy. This advancement builds upon the YOLOv5 framework, incorporating a multitude of architectural enhancements, a novel backbone network, a state-of-the-art loss function, a cutting-edge anchor-free detection head, and other innovative features. Due to its advantages in detection speed, accuracy, and adaptability to various scales, YOLOv8 has also begun to be applied across a range of object detection tasks, with researchers making a series of contributions based on this framework. Chen et al. [27] employed an Auxiliary Learning Feature Fusion (ALFF) module composed of an LSTM and a convolutional block attention module as an auxiliary task on the YOLOv8 network to enhance head detection performance, aiding the model in more accurately perceiving targets.

Huang et al. [28] constructed a progressive feature pyramid architecture based on YOLOv8, accelerating the model's training speed and enhancing its feature extraction capabilities. Jin et al. [29] introduced the BiFormer attention module into the backbone network, improving the network's ability to represent features. Yang et al. [30] integrated deformable convolutions into the YOLOv8 framework to capture finer-grained spatial information and coordinated attention mechanisms to emphasize important features during the detection process. Li et al. [31] modified the YOLOv8 backbone network with an MHSA attention mechanism to enhance the network's capacity to extract diverse features. Yang et al. [32] combined a Dual-Path Attention Gate (DPAG) and a Feature Enhancement Module (FEM) in YOLOv8, increasing the model's detection precision in complex environments.

Due to the varying types and configurations of ships, there is considerable diversity in their dimensions and aspect ratios. Consequently, accurately discerning the position of vessels necessitates extensive extraction of multi-scale information from different nearshore ships. Moreover, due to the specificity and limitations of shooting visuals, nearshore ships in parallel waterways are prone to mutual occlusion, resulting in unstable recognition features and confusion. Maritime vessels, being distant, often appear smaller in images, with less discernible features. Furthermore, unavoidable natural and artificial surroundings, such as lighting, weather, and buildings, create severe background interference, posing significant obstacles to the accurate identification of ships. How to accurately detect and accentuate the inherent characteristics of ships remains an important challenge. Previous

deep learning-based ship detection methods still cannot fully meet the detection demands of real-world environments.

Therefore, we present a new ship detection framework based on an improved version of YOLOv8 for object detection in complex settings, which we designate as FSN-YOLO. The complete network architecture comprises a backbone network based on FasterNet, a neck incorporating the receptive-field attention (RFA) mechanism, and a head designed following the YOLOv8 framework.

More specifically, we start by reconstructing the backbone component of YOLOv8 with FasterNet's backbone network, utilizing a lightweight CNN for super-resolution processing of images to enrich feature representation. This approach seeks to balance speed and model size while delivering more accurate results. Additionally, we employ an RFA mechanism, which enhances the model's comprehension of complex scenes by emphasizing salient features and suppressing irrelevant noise, thereby improving accuracy. This is particularly critical for achieving fine-grained image recognition. We conducted extensive experiments on the public Seaship7000 dataset, comparing our approach with several domain-specific and general CNN-based detectors to examine the effectiveness of our method and its modules. The experimental outcomes attest to the superiority of this method in nearshore ship detection.

The contributions of this paper are summarized as follows:

(1) We introduce a ship detection methodology named FSN-YOLO, an enhanced ship detector based on YOLOv8. Utilizing FasterNet's efficient neural network, we reconstruct the backbone of YOLOv8, achieving a balance between model precision, speed, and parameter optimization.
(2) We employ the RFA mechanism to focus more on the distinctive features of nearshore ships following feature fusion, curbing the interference of background information and reducing feature redundancy.
(3) We conduct comprehensive experiments on the publicly available Seaship7000 ship detection dataset to assess the impacts of different improved modules in our model on ship detection performance. In addition, we demonstrate the effectiveness of our approach in detecting nearshore vessels compared with domain-specific and general CNN-based detection frameworks.

2. The Framework

2.1. Model Overview

In this section, we integrate a lightweight neural network architecture, FasterNet, with an RFA mechanism into YOLOv8 to develop a novel architecture named FSN-YOLO. The structure of the network consists of a backbone based on FasterNet, with the original neck replaced by an RFAConv module substituting the C2f block and retaining the original head of YOLOv8. The overall scheme of the proposed method is depicted in Figure 1.

The FSN-YOLO architecture comprises four main components, namely the input, backbone network, neck network, and detection head, as illustrated in Figure 1.

At the input stage, mosaic data augmentation techniques are utilized to increase the diversity of the dataset. Additionally, images can be adaptively scaled to a specified size.

As shown in Figure 1, the backbone network employs FasterNet as the backbone architecture, consisting of four hierarchical stages, each made up of several FasterNet blocks. This design not only speeds up the neural network's processing time but also ensures precision in visual tasks. Within the neck structure, introduce an RFAConv module to replace the original C2f block. The RFAConv module refines and expands the feature maps more dynamically, enhancing the model's understanding of the detection details and context. The detection head features an anchor-free design and incorporates an adaptive spatial feature fusion strategy to effectively filter out conflicting information, leading to a significant improvement in performance in target detection tasks.

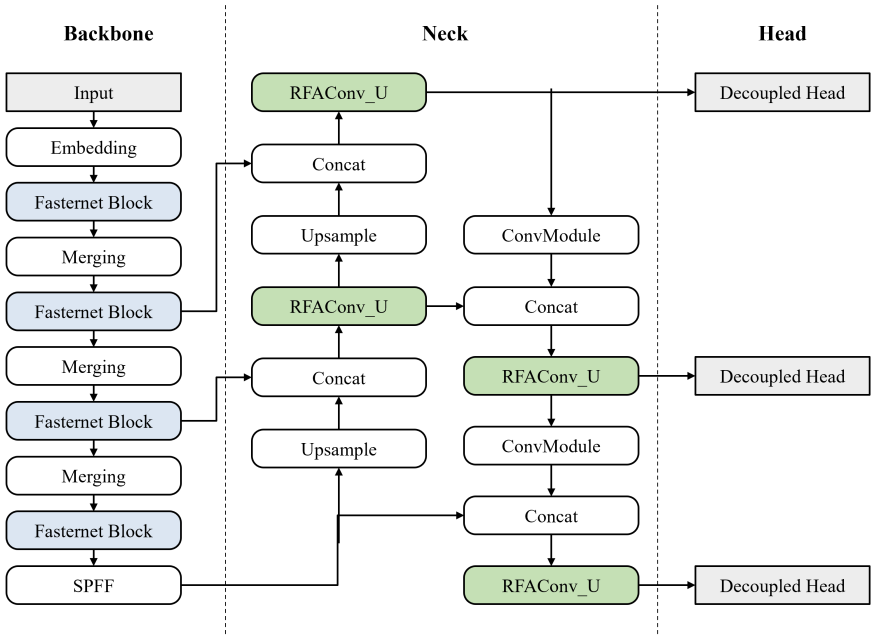


Figure 1. The overall framework of FSN-YOLO.

2.2. FasterNet

FasterNet is designed as a high-speed backbone network to overcome the limitations of traditional neural networks in terms of processing speed and efficiency. It innovatively incorporates a novel convolutional operator, PConv or partial convolution. PConv selectively performs conventional convolution on a subset of channels while keeping the remaining channels' features unchanged, thus reducing redundant computations and memory usage to enhance the efficiency of spatial feature extraction. FasterNet has been applied within the backbone networks of YOLOv5 and YOLOv8, aiming to accelerate the neural network's processing speed while maintaining a high level of accuracy. The primary differences between YOLOv5 and YOLOv8 in the backbone portion lie in the network structural modules and channel-number adjustments. For the network structure component, YOLOv8 employs a C2f structure to replace the C3 module in YOLOv5 to enrich gradient flow and enhance model performance and accuracy. The channel-number adjustment in YOLOv8 is tailored for different scales of models, ensuring that the model can select the appropriate number of channels based on different input sizes, thereby achieving more efficient feature extraction and computational efficiency.

As illustrated in the backbone section of Figure 1, FasterNet features four hierarchical levels, and each level is preceded by either an embedding layer (a regular 4×4 convolution with a stride of 4) or a merging layer (a regular 2×2 convolution with a stride of 2) for spatial downsampling and channel expansion. These levels are composed of several FasterNet blocks. As depicted in Figure 2, each FasterNet block contains one PConv followed by two 1×1 point-wise convolutions, forming an inverted residual structure. Within this structure, the intermediate layer has an expanded number of channels, and a shortcut is placed to facilitate the reuse of input features.

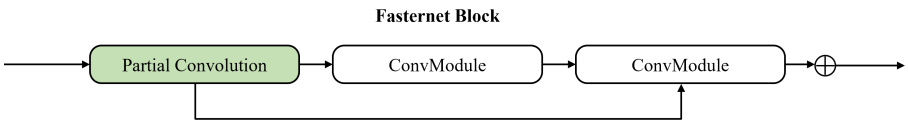


Figure 2. Structure of the FasterNet block.

As demonstrated in Figure 3, given input data of $I \in R^{c \times h \times w}$, a $k \times k$ conventional convolution is used to compute the output ($O \in \mathbb{R}^{c \times h \times w}$). The computational load and memory-access volume for completing one conventional convolution operation are calculated according to Equation (1) as follows:

$$q = h \times w \times k^2 \times c^2, n = h \times w \times 2c + k^2 \times c^2, \quad (1)$$

where q is the computational requirement for completing one convolution with a standard convolution; c is the number of channels involved in the convolution process; h and w are the width and height of the feature map, respectively; k is the size of the convolutional kernel; and n is the amount of memory access necessary to complete one convolution with a standard convolution.

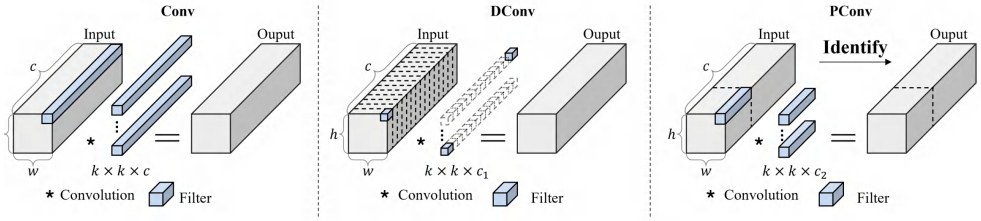


Figure 3. The differences between PConv, Conv, and DWConv.

In depthwise convolution, each filter operates independently across its respective channel, effectively reducing the number of floating-point operations (FLOPs). However, this approach tends to overlook inter-channel dependencies, which can lead to a significant decline in model accuracy. Therefore, it is not a straightforward replacement for standard convolution. In practice, the channel width is often increased in depthwise convolutions (DWConvs) to compensate for the loss of accuracy, but this results in higher memory access costs and potentially slows down overall computation speeds. When the number of channels is expanded from c to c_1 (where $c_1 > c$), the computational load and memory access volume for a single depthwise convolution operation are calculated according to Equations (2) as follows:

$$q_1 = h \times w \times k^2 \times c_1, n_1 = h \times w \times 2c_1 + k^2 \times c_1^2, \quad (2)$$

In the specified scenario, q_1 corresponds to the computational requirement for conducting one convolution with depthwise convolution, and n_1 denotes the amount of memory access required to complete one convolution. Additionally, c_1 indicates the number of channels involved in the convolution process.

Partial convolutions, designated as PConv, selectively conduct standard convolution operations on a subset of input channels for the purpose of extracting spatial features while simultaneously preserving the integrity of the remaining channels. The computational load and memory access profiles for PConv are calculated according to the Equations (3) as follows:

$$q_2 = h \times w \times k^2 \times (c_{2p})^2, n_2 = h \times w \times 2c_2 + k^2 \times c_2^2, \quad (3)$$

In this case, q_2 is defined as the computational quantity needed to perform one convolution in partial convolution (PConv), n_2 refers to the memory access required for one such convolution, and c_2 represents the channel count engaged in the convolution operation. In actual implementations, it is common to have a ratio of $r = c_2/c = 1/4$. This results in the computational expense for PConv being only 1/16th of that of standard convolution, and the required memory access is merely 1/4th. The residual channels, totaling $(c - c_2)$, are not involved in the computation, thus negating the need for memory access for those channels.

Upon comparing the computational demand and memory access requirements across three convolutional methods, it is evident that partial convolutions offer a lower floating-point operation count than standard convolutions and a reduced memory footprint as well as a higher count of floating-point operations per second (FLOPS) compared to depthwise convolutions. The FasterNet block, composed of partial convolutions (PConvs) and point-wise convolutions (PWConvs), is capable of efficiently extracting spatial features while minimizing redundant computation and memory access.

To efficiently leverage information from all channels, a point-wise convolution (PWConv) is typically appended following a PConv. As depicted in Figure 4, this architectural setup yields an effective receptive field on the input feature map akin to a "T-shaped" convolutional operation. This structure places heightened emphasis on the central region of the feature map compared to traditional convolutional approaches. While a T-shaped convolution could be implemented directly, decoupling it into PConv and PWConv is advantageous. This decomposition capitalizes on the redundancy among the filters, further diminishing the computational demand. For an identical input ($I \in \mathbb{R}^{c\times h\times w}$) and output ($O \in \mathbb{R}^{c\times h\times w}$), the floating-point operations (FLOPs) of the 'T'-shaped Conv are calculated according to the Equation (4) as follows:

$$h \times w \times (k^2 \times c_p \times c + c \times (c - c_p)), \tag{4}$$

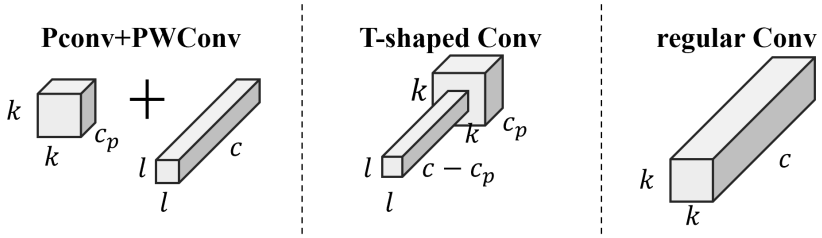


Figure 4. The effective receptive field on the input feature map, which collectively resembles a 'T'-shaped convolution.

2.3. RFA Attention Mechanism

The floating-point operations (FLOPs) resulting from the decoupling into PConv and PWConv are calculated according to the Equation (5) as follows:

$$h \times w \times (k^2 \times c_p^2 + c \times c_p), \tag{5}$$

Given that $c > c_p$ and $c - c_p > c_p$, decoupling into PConv and PWConv is beneficial for reducing the computational load.

The underlying concept of the neck multi-scale feature fusion network is to amalgamate feature maps extracted from distinct network layers to augment the performance of multi-scale target detection. However, the feature fusion layers in YOLOv8 still grapple with the issue of redundant information from disparate feature mappings and overlook unique prior knowledge intrinsic to the scene. This knowledge is instrumental, as some minute targets might be mistakenly detected without adequate reference to the distant background context, which varies for different targets. Moreover, the existing feature fusion layers often require substantial computational resources to integrate multi-scale features, leading to high model complexity and computational costs that can impede processing speed and fail to meet the real-time requirements for maritime environments.

To address these challenges, we implement an RFA mechanism within the feature fusion module of the YOLOv8 model. This method directs spatial attention to spatial features within the receptive field, both emphasizing the significance of different features within the field and addressing the limitations of traditional convolutional kernel parameter sharing. Through the receptive-field attention mechanism, the network can process features within the field with greater precision, as opposed to treating all features indiscriminately. This approach not only enhances the model's capacity to comprehend complex patterns but also significantly boosts network performance with minimal additional computational cost and parameter count.

Consequently, we adopt an RFA module to enhance the CNN-based feature fusion architecture, better capturing the range-finding environment of various objects within the scene and focusing on the maritime features of ships in images. This achieves effective multi-level feature fusion, substantially boosting the overall network performance through lightweight operations. During the feature fusion phase, we replace the C2f module with the

RFAConv module, which not only increases the efficiency of the model but also strengthens the network's ability to discern ships of varying scales and their performance in complex environments, especially in capturing targets with occlusions and irregular shapes.

The RFA attention mechanism, as illustrated in Figure 5, amalgamates spatial attention with convolutional operations to enhance the performance of the CNN.

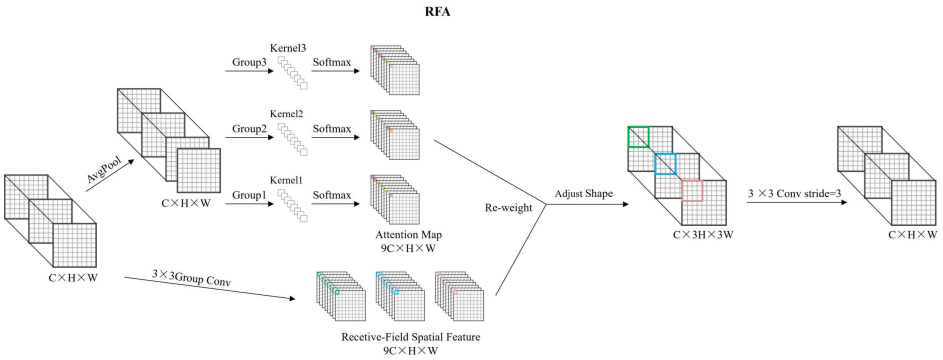


Figure 5. The RFA (receptive-field attention) mechanism module.

RFA can be regarded as an instantly integrable module that boosts the overall efficacy of CNNs with its specially designed RFAConv computation, serving as an alternative to standard convolution operations. The spatial attention mechanism addresses the issue of parameter sharing in traditional convolution models by focusing on receptive-field spatial features. To better comprehend the concept of receptive-field spatial features, we elucidate it specifically through Figure 6. Receptive-field spatial features are custom-tailored for convolution kernels and dynamically generated based on kernel size. Figure 6 illustrates a convolution kernel of size 3 × 3, where "spatial features" refer to the original feature map. The "receptive-field spatial features" are derived from the transformed feature map, which is composed of non-overlapping sliding windows. Each 3 × 3-sized window in the receptive-field spatial features represents a receptive-field block.

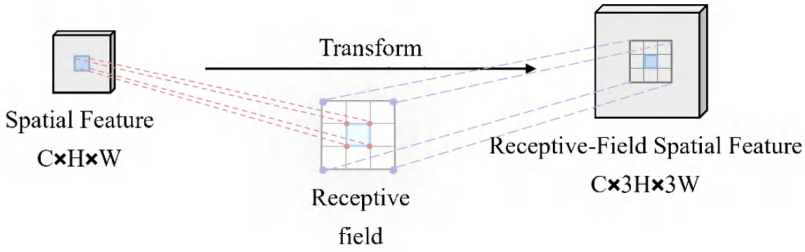


Figure 6. Transforming spatial features to obtain receptive-field spatial features.

In the RFAConv, group convolution is employed to rapidly extract spatial features from receptive fields, mapping the original features into a new feature space for unfolding, thereby achieving faster speed and higher efficiency compared to traditional unfolding methods. Additionally, the RFAConv improves network performance through the learning of attention maps by interacting with receptive-field feature information. It utilizes AvgPool to aggregate the global information of each receptive-field feature. Then, 1 × 1 group convolution operations are used for information exchange. Finally, softmax is applied to underscore the significance of each feature within the receptive-field characteristics, minimizing computational costs and the number of parameters. This enables the capture of more complex patterns and details, enhancing the model's ability to recognize ships and increasing detection speed. Generally, the computation within RFA can be calculated according to Equation (6) as follows:

$$F = Softmax\left(g^{(1\times1)}(AvgPool(X))\right) \times ReLU\left(Norm(g^{(k\times k)}(X))\right) = A_{rf} \times F_{rf}, \quad (6)$$

where $g^{1\times1}$ represents the group convolution of size i, k denotes the size of the convolution kernel, *Norm* signifies normalization, X stands for the input feature map, and F is the product of the attention map (A_{rf}) and transformed receptive-field spatial features (F_{rf}). Unlike CBAM and CA, RFA is capable of generating attention maps for each receptive-field feature. The performance of CNNs is constrained by the limitations of standard convolution operations, as they rely on shared parameters and are insensitive to positional variations in information. However, RFAConv effectively addresses this issue by emphasizing the importance of different features within the receptive-field block and giving precedence to receptive-field spatial features.

In the FSN-YOLOv8 model, the feature map input to the RFAConv module is denoted as X, with dimensions of $W \times H \times C$, where W is the width, H is the height, and C represents the number of channels. The input feature map (X) is divided into N groups, each containing C/N channels, resulting in N feature maps denoted as X_i, where $1, 2, \ldots, N$ and each has dimensions of $W \times H \times C \div N$. Group convolution is applied to each group to obtain N convolved feature maps.

For each convolved feature map (X_i'), the RFA mechanism is applied to enhance the inter-channel correlation. Each feature map (X_i') is element-wise multiplied by its corresponding attention weight (A_i) to produce the weighted feature map, which is denoted as X_i''.

$$X_i'' = X_i' \otimes A_i, \tag{7}$$

The weighted feature map (X_i'') is combined with the original input feature map (X) through channel-wise addition to yield the integrated feature map (Y_i).

$$Y_i = X + X_i'', \tag{8}$$

Aggregated feature maps (Y_i) are summed along the channel dimension to produce the final output feature map (Y).

$$Y = \sum_{i=1}^{N} Y_i, \tag{9}$$

Finally, the resulting output feature map (Y) is passed to the subsequent layer of the network, serving as the output of the RFAConv module.

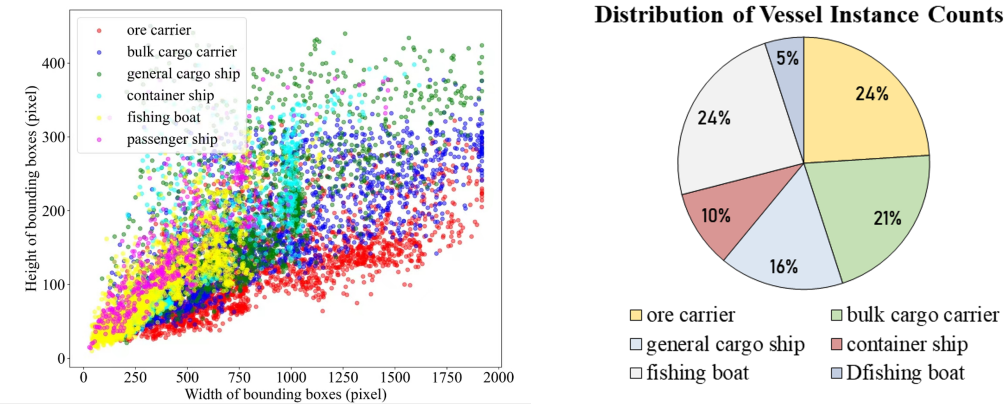
3. Experimental Results and Analysis

3.1. Dataset

In this study, we investigate the detection of nearshore vessels under complex conditions such as intensive traffic flow, diverse types of ships, adverse weather conditions, and instances where ship and shore elements are intermixed. The performance of the proposed FSN-YOLO model was evaluated on the publicly available Seaship dataset, which focuses on nearshore ship detection.

The dataset integrates 7000 high-resolution images, reaching up to 1920 × 1080 pixels, all captured by an array of visual surveillance cameras positioned in the vicinity of the Hengqin New Area in Zhuhai. The collection encompasses a diverse array of ship sizes and silhouettes, with images that chronicle the vessels' appearances across a spectrum of environmental conditions. This includes documentation under low-light scenarios, as well as a range of lighting conditions from uneven illumination and direct sunlight exposure to the nuanced lighting of twilight hours. Furthermore, the dataset comprehensively includes complex instances such as partial occlusions and situations where only segments of the ships are discernible, offering an extensive suite of test cases to evaluate the robustness of model performance within a multifaceted environmental context. The vessels are categorized into the following six classes: ore carriers (OCs), bulk cargo carriers (BCCs), general cargo ships (GCSs), container ships (CSs), fishing boats (FBs), and passenger ships (PSs). The number of instances for each category is 2199, 1952, 1505, 901, 2190, and 474, respectively.

Figure 7 illustrates the size and category distribution of inshore vessels within the Seaship7000 dataset. Among them, passenger ships and container ships have the fewest instances, followed by general cargo ships and bulk carriers, with ore ships and fishing vessels being the most prevalent.



(a) (b)

Figure 7. (**a**) Width and height of bounding boxes for vessels across different categories. (**b**) The instance distribution of different categories of ships.

In the experiment, as depicted in Figure 8, we randomly divided the dataset into training, validation, and test sets in an 8:1:1 ratio. The training set comprised 5600 ship images, while the test and validation sets contained 700 ship images each. All results were evaluated on the test set. The images were resized to 640 × 640 to fit the input size required by the method.

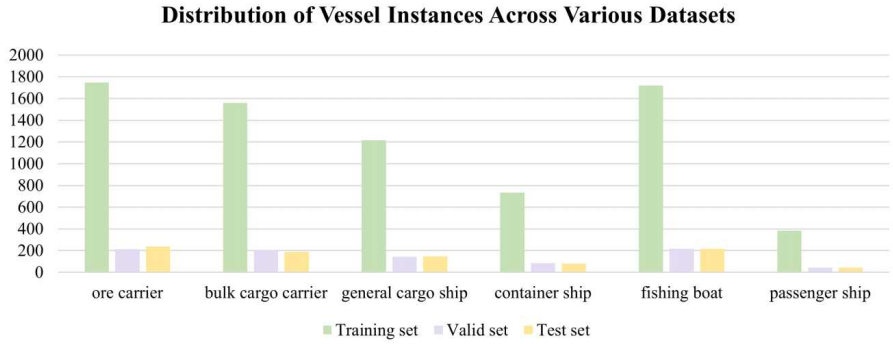


Figure 8. The number of different categories of ships in the training, validation, and test sets.

3.2. Implementation Details

The experimental environment was based on Python 3.8.18 and implemented using the PyTorch 1.11.0 framework in a Windows 11 setting. All experiments were conducted on an Ubuntu 23.04 server equipped with an NVIDIA A6000 GPU (ASUS Corporation, Shanghai, China), a 10-core 4.8 GHz CPU (with Turbo Boost) (Intel Corporation, Dalian, China), and 192 GB of RAM (Kingston Technology Company, Inc., Shenzhen, China). In the preprocessing of images, we set the height and width to 640 × 640, with a batch size of 16. During training, observed that the model tended to stabilize around epoch 150. To conserve computational resources, we adjusted the number of epochs for the training of all models to 300. We employed SGD as the optimizer, with an initial learning rate of 0.01, decreasing to 0.001 towards the end of training; a momentum of 0.937; and weight decay set at 0.0005. To accelerate model convergence, we disabled mosaic data augmentation during the final 10 epochs of each training session .

In YOLOv8, there are various variants with different network architectures. Considering the requirements of precision and speed for ship detection, we implemented the

proposed method based on YOLOv8l, using the YOLOv8l trained on the Seaship7000 dataset as a representative baseline model.

3.3. Evaluation Metrics

In this paper, we employ precision (P), recall (R), and mean Average Precision (mAP) to assess the performance of the detection models, and we use the number of parameters and inference time to evaluate the efficiency of the models. The specific definitions are as follows:

(1) Precision and Recall

Precision (P) represents the accuracy rate, which is the ratio of the number of objects correctly detected to the total number of objects detected by the algorithm. The higher the precision value, the more accurate the detection results obtained by the algorithm. Recall (R) represents the recall rate, which is the ratio of the number of objects detected by the algorithm to the number of actual objects that exist. The higher the recall value, the less likely the algorithm is to miss real detection objects. The specific formulas for P and R are presented as follows:

$$P = TP/(TP + FP), \tag{10}$$

$$R = TP/(FP + FN), \tag{11}$$

where TP refers to correctly classified positive samples, TN refers to correctly classified negative samples, FP refers to incorrectly classified negative samples, and FN refers to incorrectly classified positive samples.

(2) Mean Average Precision

Mean average precision (mAP) is a metric that measures the accuracy of an algorithm in detecting objects across different classes. It takes into account the overall network performance indicators of precision (P) and recall (R) and is the most commonly used evaluation metric in object detection. A larger mAP value indicates better detection accuracy. mAP@0.50, which is the mean average precision at an IoU threshold of 0.50, is an indicator used in evaluating the performance of object detection models. It reflects the overall accuracy of the model in detection tasks. mAP@0.5:0.95 represents the average mAP calculated at different IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05, providing a more comprehensive evaluation of model performance.

Therefore, employing both mAP@0.5 and mAP@0.5:0.95 metrics provides differing degrees of performance assessment for the model, with higher mAP@0.5 and mAP@0.5:0.95 scores indicating better detection performance. The calculation formula is

$$AP = \int_0^1 P(R)\, dR, \tag{12}$$

$$mAP = \frac{1}{n} \sum_{i=1}^{n} \mathrm{AP}_i, \tag{13}$$

where AP_i represents the AP value for category i, and n is the number of categories.

(3) Parameters

The number of parameters refers to the total count of parameters that need to be trained during model training, which describes the complexity of the entire model and is used to measure the size of the model. Generally speaking, the fewer the network parameters, the easier it is to embed the model into devices.

(4) Inference time

Inference time is the amount of time required to test an image, reflecting the processing speed of the model in practical applications. The longer the inference time, the slower the network speed.

3.4. Ablation Study

We conducted a series of ablation studies to evaluate the contribution of each module within the FSN-YOLO framework and examined the impact of various techniques utilized in each method on detection performance through extensive experimentation. We implemented

the proposed FSN-YOLO model and comparative models based on the "l" version of YOLOv8 (YOLOv8l), ensuring they utilized identical hyperparameters during training. We used YOLOv8l, devoid of any design modules, as a baseline for our ablation experiments, assessing the influence of various improved modules on FSN-YOLO through the Seaship dataset. In addition, we tested four incomplete FSN-YOLO architectures by sequentially removing each module. YOLOv8+FasterNet denotes the replacement of the original YOLOv8 backbone structure with the FasterNet module; YOLOv8+RFA indicates that on the basis of the original YOLOv8, the C2f module in the neck is replaced by the RFA module.

Table 1 illustrates the outcomes of various ablation experiments conducted on the FSN-YOLO model, with ✓ indicating the activation and **x** representing the deactivation of specific modules within the compared methods. The results indicate that incorporating the FasterNet and RFA modules into the original YOLOv8 network improved experimental outcomes to varying degrees. Notably, the introduction of the FasterNet module led to a 0.93% relative increase in recall, highlighting the enhancement in the model's detection scope. Moreover, there was a significant rise of 2.28% in the stringent mAP@0.50:0.95 evaluation metric. This indicates that YOLOv8, which adopts the FasterNet backbone network, is capable of capturing more detailed features of ships compared to the original YOLOv8. It reduces the interference from background information, thereby enabling the model to extract appropriate discriminative features and reduce the occurrence of false detections and missed detections. Concurrently, a significant parameter reduction was observed along with a decrease of 1.4 ms in inference time, thus shortening the model's response time and enhancing its suitability for real-time applications that require rapid decision-making. Although a slight decrease in precision was noted, it is anticipated that subsequent refinements will compensate for this.

Table 1. Results of various ablation experiments for the FSN-YOLO network on the Seaship7000 testing set.

Model	FasterNet	RFA	P	R	mAP@0.5	mAP@0.5:0.95	Parameters (M)	Inference Time (ms)
YOLOv8	x	x	0.981	0.971	0.99	0.832	43.63	4.4
YOLOv8+ FasterNet	✓	x	0.975	0.98	0.992	0.851	43.12	3.0
YOLOv8+ RFA	x	✓	0.982	0.985	0.992	0.851	35.77	7.4
FSN-YOLO (Ours)	✓	✓	0.989	0.986	0.993	0.845	48.56	7.4

Following the incorporation of the RFA module, marked enhancements in accuracy, recall, mAP@0.5, and mAP@0.5:0.95 were observed relative to the YOLOv8 baselines. Specifically, there was a 1.44% uplift in recall and a 2.28% improvement in mAP@0.5:0.95. Consequently, the step-wise addition of modular components to the network is justified. Each module within the FSN-YOLO framework contributes to the refinement of detection performance, advancing the model's capabilities to measurable degrees.

Table 1 further reveals that our proposed FSN-YOLO model achieved the best performance in precision, recall, and mAP@0.5, with values of 0.989, 0.986, and 0.993, respectively, showcasing its accurate target recognition capabilities. The most salient enhancement was noted in recall, where, compared to YOLOv8, performance improved from 0.971 to 0.986—a relative increase of 1.54%. Additionally, both the model's mAP@0.5:0.95 and precision also exhibited substantial increases of 1.56% and 0.82%, respectively, further validating the substantial strides made in bolstering the model's recognition abilities. Although this led to a slight increase in the size of the model parameters, with overall inference time rising from 4.4 ms to 7.4 ms, the network maintained its real-time capability throughout processing. Thus, it is evident that our proposed FSN-YOLO model is effectively applicable to the task of vessel detection.

To provide a more intuitive analysis of the impact of our proposed FSN-YOLO model on ship detection, we visualized the performance curves of the YOLOv8, YOLOv8+FasterNet, YOLOv8+RFA, and FSN-YOLO models across various metrics during the training process. The specific content is shown in Figure 9.

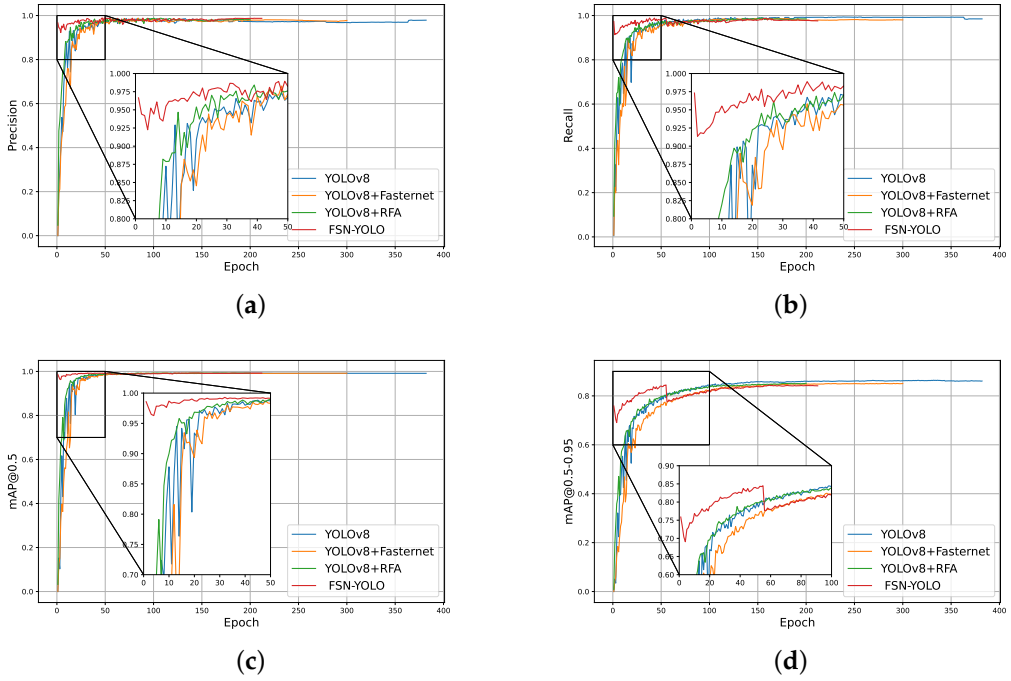


Figure 9. Training processes of different networks during ablation experiments. (**a**) The precision curve, (**b**) the recall curve, (**c**) the mAP@0.5 curve, and (**d**) the mAP@0.5:0:95 curve.

It can be discerned from Figure 9 that our proposed FSN-YOLO (represented in red) outperformed YOLOv8 and its comparable methods across all evaluation metrics. Consequently, FSN-YOLO exhibited a significantly superior ability to identify various multi-scale ships in complex maritime scenes when compared to its counterparts.

Figure 10 presents a comparison of the precision–recall (*PR*) curves between YOLOv8 and our proposed FSN-YOLO model. As seen in Figure 10, our model performed well across the majority of ship categories. With the optimization goal of enhancing ship detection precision while maintaining real-time network performance, our FSN-YOLO model achieved an average precision of 0.993 across all categories, making it superior to the traditional YOLOv8 in overall performance. The PR curves in Figure 10b, approaching the ideal state (near the point (1,1)), denote extremely high standards of precision and recall, further validating the model's effectiveness. Notably, the average precision for the fishing boat category increased from 0.974 with YOLOv8 to 0.987 with FSN-YOLO, corresponding to a growth of 1.33%.

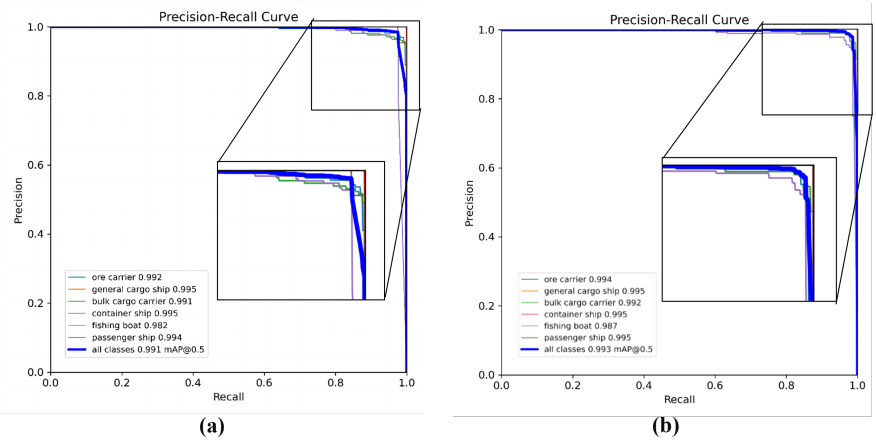


Figure 10. Comparative chart of PR results at an IOU threshold of 0.5. (**a**) *PR* of YOLOv; (**b**) *PR* of our proposed network.

Figure 11 demonstrates the specific detection effects of our proposed FSN-YOLO model in comparison with baseline methods on the test set. The baselines were selected as YOLOv8, along with the three comparably best-performing methods, namely YOLOv8+FasterNet and YOLOv8+RFA. This comparison includes the concrete detection results of different models on the test set. We selected six different kinds of images (Figure 11a–f) corresponding sequentially to the following categories: OC, BCC, GCS, CS, FB, and PS.

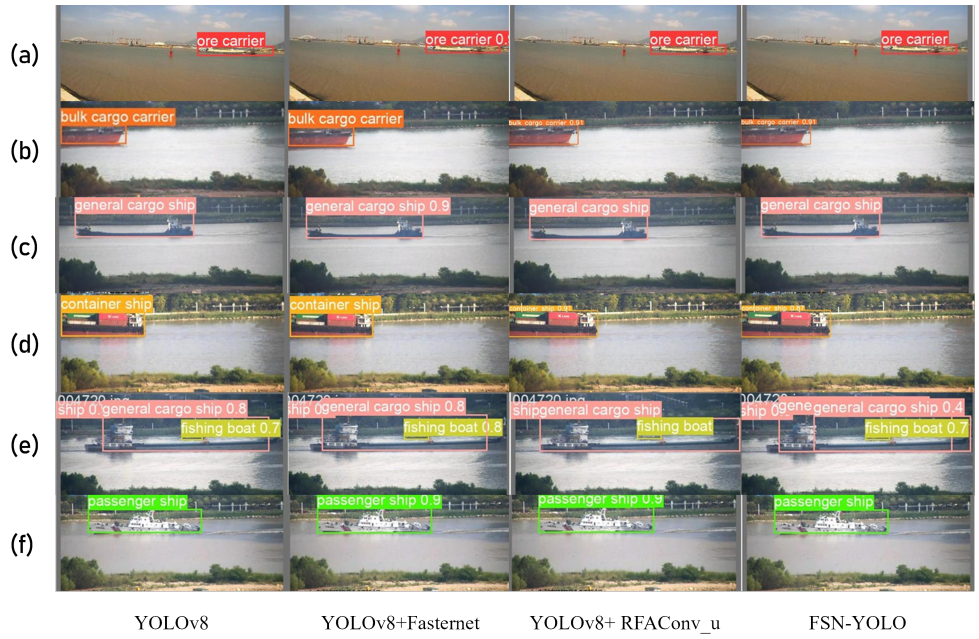


Figure 11. (**a**) OC detection results of different model, (**b**) BCC detection results of different model, (**c**) GCS detection results of different model, (**d**) CS detection results of different model, (**e**) The detection results of multi-scale ships, including FB, on different models,and (**f**) PS detection results of different model.

As shown in the fifth row of Figure 11, utilizing YOLOv8 for detection of distant ships, a GCS and an FB were detected from the image. However, when utilizing the FSN-YOLO method to detect distant ships, two GCS and one FB were detected. Compared to the original YOLOv8, FSN-YOLO's detection is more comprehensive, reducing missed detections and false detections and improving the detection performance of small targets. It performs well in both target positioning and classification.

3.5. Comparison with State-of-the-Art Methods

3.5.1. Comparison with Generic Detection Methods

When our model is juxtaposed with the current leading CNN-based detection models, it demonstrates superlative performance. To ensure a fair and equitable comparison, a range of models including YOLOv6, YOLOv7, YOLOv8, TPH-YOLOv5, and TPH-YOLOv5++ was selected for evaluation. All models were assessed based on the 'Large' (L) size version benchmark.

To guarantee the consistency of the results, all methods under comparison were trained on the same dataset, 'Seaship7000'. The comparative results shown in Table 2 reveal the performance comparison between our proposed method and conventional CNN-based methods. In the case of models designed for the 'l' size, both TPH-YOLOv5 and TPH-YOLOv5++ represent enhancements based on the original YOLOv5 architecture. Compared to other baseline models, there are, indeed, improvements in precision, recall, and mAP0.5. Specifically, TPH-YOLOv5++ achieves a precision of 0.977, a recall of 0.967, and an mAP@0.5 of 0.987. However, in the case of similar parameter quantity and inference time, these models still fall short compared to our proposed 'L' size model, Ours-l. Our

method achieves a precision of 0.989 and a recall rate of 0.986. Our model also achieves a high mAP@0.5 of 0.993 and an mAP@0.5–0.95 of 0.845, outperforming all other detection methods mentioned above. Notably, our Ours-l model improves precision by 1.12% and recall by 1.02% compared to the other models.

Table 2. Comparison with current CNN-based general methods.

Method	P	R	mAP@0.5	mAP@0.5:0.95	Parameters (M)	Inference Time (ms)
YOLOv6l	0.978	0.976	0.989	0.827	110.87	33.6
YOLOv7	0.98	0.98	0.993	0.816	36.51	12.3
YOLOv8l	0.981	0.971	0.99	0.832	43.63	4.4
TPH-YOLOv5	0.967	0.969	0.986	0.781	45.40	33.2
TPH-YOLOv5++	0.977	0.967	0.987	0.801	41.52	19.2
Ours-l	0.989	0.986	0.993	0.845	48.56	7.4

Table 3 presents a comparison of our proposed method against other CNN-based detection methods in terms of mAP values for different types of ships across varying IOU thresholds from 0.5 to 0.95. Table 3 demonstrates the mAP values of our proposed method and other CNN-based detection methods for different types of vessels under different IOU thresholds from 0.5 to 0.95. Among the various types of vessels, our proposed method achieves more competitive average precision compared to other methods. Specifically, for GCS, CS, and FB, our method achieves improvements of 1.40%, 2.44%, and 1.90%, respectively over the best results achieved by other detection methods. Particularly for container ships, despite the small actual number, the improvement effect is most evident.

Table 3. Comparison of mAP0.5:0.95 for different types of ships with current CNN-based general methods.

	Ore Carrier	Bulk Cargo Carrier	General Cargo Ship	Container Ship	Fishing Boat	Passenger Ship
YOLOv6l	0.831	0.826	0.86	0.86	0.78	0.808
YOLOv7	0.802	0.85	0.831	0.851	0.755	0.807
YOLOv8l	0.802	0.826	0.856	0.86	0.791	0.858
TPH-YOLOv5	0.787	0.792	0.821	0.811	0.705	0.772
TPH-YOLOv5++	0.797	0.818	0.838	0.839	0.747	0.768
Ours-l	0.814	0.833	0.872	0.881	0.806	0.864

Figure 12 displays precision–recall curves for all CNN-based comparison methods at an IoU threshold of 0.5. It is clear to see that as the recall increases, the curve of our proposed method (e.g., the red curve) remains higher than those of the others. This demonstrates that our methodology attains superior precision and enhanced performance efficacy for object detection tasks applying a 0.5 IOU threshold criterion. Specifically, our model excels in accurately recognizing and localizing target objects with increased precision as the recall rate incrementally rises. When juxtaposed with other techniques, especially in challenging detection scenarios, our model manifests a pronounced competitive edge.

Moreover, to observe the detection performance variability among different ship types more comprehensively, the precision–recall curves for all comparison methods detecting various ship categories at an IoU threshold of 0.5 are illustrated in Figure 13.

We can observe from Figure 13 that our model's curve exceeds those of other models as the recall rate increases for most ship categories, implying that our model obtains competitive accuracy in the detection of a wide range of vessel classes. Especially in categories such as BCC (bulk cargo carriers), FB (fishing boats), and PS (passenger ships), consistent accuracy improvements are achieved. Particularly in the category of passenger ships, where the number of instances is the least, our Ours-L model also achieves the best detection effect. These results indicate that our model not only performs exceptionally well overall but also achieves outstanding performance in various specialized fields.

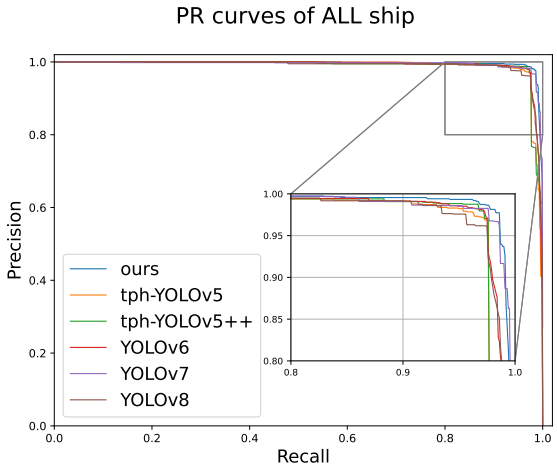


Figure 12. Precision–recall curves of CNN-based comparison methods on the Seaship7000 dataset at an IoU threshold of 0.5.

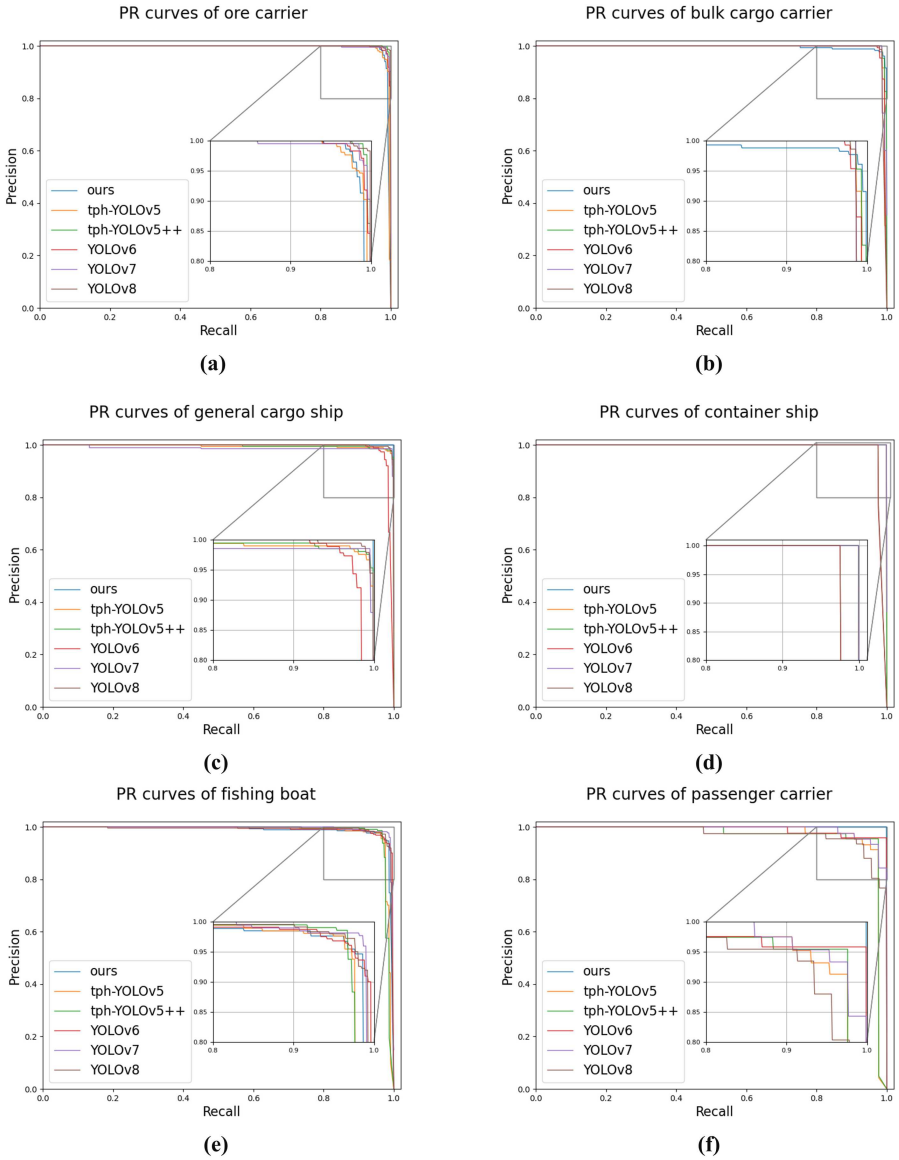


Figure 13. Precision–recall curves for different types of vessels detected by CNN-based methods at an IoU threshold of 0.5.(**a**) PR curves of OC , (**b**) PR curves of BCC, (**c**) PR curves of GCS, (**d**) PR curves of CS, (**e**) PR curves of FB,and (**f**) PR curves of PS.

3.5.2. Domain-Specific Comparison Methods

To more clearly validate the effectiveness of our proposed model in ship detection, we compared our framework with several domain-specific methods for ship detection. These include the Anchor-Guided Attention Refinement Network (AARN) for ship detection proposed by Liu et al. [20], the multi-scale weighted fusion-based ship detection model proposed by Zhou et al. [21], and the Incremental Learning-based ship detection approach, IL-YOLOv5, proposed by Liu et al. [33].

For a fair comparison, we employ the commonly used metrics of mAP@0.5 and mAP@50:95 to evaluate the performance of the models. Additionally, we incorporate an FPS metric, with the definition of FPS being consistent with that of Liu et al. [20], as shown in Equation (14).

$$FPS = \frac{1}{t_{\text{img}}}, \tag{14}$$

where t_{img} represents the time required for a method to process an image.

Table 4 shows the comparative detection results of our proposed method with several domain-specific methods for nearshore ship detection. As can be observed from Table 4, except for the mAP@50 for FB, our method achieved the highest scores in mAP@50 and mAP@50:95 among these specific domain methods. For instance, our method improved by 0.40%, 0.80%, 1.43%, and 0.40% over the optimal baseline model, IL-YOLOv5, in detecting OC, GCS, CS, and PS, respectively, on mAP@50. For BCC, our method is on par with the best baseline, while there was a slight decrease in 0.50% on FB, but overall, there was a significant increase in mAP@50, with an average increase of 6.62%. With respect to the mAP@50:95 metric, our proposed method displayed improvements over the best baseline model 0f 7.25%, 5.20%, 4.93%, 5.63%, 7.47%, and 8.89% for OC, BCC, GCS, CS, FB, and PS, respectively.

Although Liu et al. [20] enhanced the salience of ship targets and suppressed background interference through an attention mechanism, the detection performance remained suboptimal due to the limited feature extraction and recognition capabilities of their backbone network. Moreover, Zhou et al. and Liu et al. [21,33] introduced the Channel Attention (CA) mechanism into the YOLOv5 framework, enabling more accurate localization and identification of target regions. In contrast, our approach is based on the YOLOv8 structure, and we replaced the backbone with FasterNet. Compared to the original backbone in YOLOv8, FasterNet facilitates a richer flow of information, thus preserving global and local features of ships to the greatest extent possible. Additionally, we replaced the C2f module in the neck structure with the RFAConv attention mechanism, which dynamically selects convolutional kernels of varying receptive-field sizes during feature fusion. This ability to capture discriminating features of multi-scale ships, coupled with addressing the diversity of vessel classes and locations, significantly improves the accuracy of target detection.

Table 4. Ship detection performance of domain-specific methods.

Model	Metrics	ALL	OC	BCC	GCS	CS	FB	PS	Parameters (M)	FPS
AARN	mAP@0.5	0.947	0.948	0.947	0.958	0.980	0.927	0.923	35.82	45
	mAP@0.5:0.95	0.702	0.677	0.708	0.718	0.786	0.659	0.666		
YOLOv5ship	mAP@0.5	0.976	0.984	0.963	0.975	0.983	0.972	0.98	40.3	60
	mAP@0.5:0.95	0.71	0.644	0.678	0.741	0.794	0.656	0.744		
IL-YOLOv5	mAP@0.5	0.989	0.99	0.992	0.987	0.981	0.992	0.991	29.8	94
	mAP@0.5:0.95	0.79	0.759	0.792	0.831	0.834	0.75	0.777		
Ours(all-l)	mAP@0.5	0.993	0.994	0.992	0.995	0.995	0.987	0.995	48.56	134.83
	mAP@0.5:0.95	0.845	0.814	0.833	0.872	0.881	0.806	0.864		
Impro	mAP@0.5	0.40%	0.40%	0.00%	0.80%	1.43%	−0.5%	0.40%		
	mAP@0.5:0.95	7.00%	7.25%	5.20%	4.93%	5.63%	7.47%	8.89%		

Moreover, our model achieved outstanding performance in the FPS index, with a score of 134.83, demonstrating its strong competitiveness in processing speed. This is attributed to our targeted hardware acceleration for the model, which fully leverages the high-performance computing capabilities of modern GPUs and TPUs, and meticulous tuning, including adjusting batch sizes, optimizing I/O operations, and reducing network communication. These detailed improvements all contributed to the enhancement of the FPS. In practical applications, this means that users can experience smooth and latency-free image processing, making it more suitable for multi-ship target detection in complex scenarios.

4. Conclusions and Future Work

Based on YOLOv8, we developed a new FSN-YOLO model for nearshore vessels detection. The experimental studies show that the target detection capability of YOLOv8 was significantly enhanced by replacing the lightweight FasterNet backbone network and introducing an RFA mechanism to replace the original C2f module in the neck. A series of experimental evaluations through ablation studies and comparative trials showed that the accuracy of the FSN-YOLO model can reach 0.989% and mAP@0.5 can reach 0.993, demonstrating its good detection performance. Moreover, this model can adapt to dynamic and complex marine environments, such as changes in weather and sea surface fluctuations. It achieves superior performance in multi-scale target detection, able to identify small-scale or distant ships that other methods fail to detect, thus enhancing the comprehensiveness of detection. This model has significant practical significance for the development of automated ship monitoring systems, especially in areas such as port management, maritime surveillance, and maritime traffic safety.

Author Contributions: Conceptualization, H.L.; Methodology, S.G.; Validation, Q.L.; Formal analysis, N.D.; Model development and Writing—original draft, Q.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. 52171292), the Fundamental Research Funds for the Central Universities (No. 3132019355), and the Dalian Outstanding Young Talents Program (No. 2022RJ05).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

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

References

1. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
2. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
3. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [CrossRef] [PubMed]
4. Wei, S.; Chen, H.; Zhu, X.; Zhang, H. Ship detection in remote sensing image based on faster R-CNN with dilated convolution. In Proceedings of the 2020 39th Chinese Control Conference (CCC), Shenyang, China, 27–29 July 2020; pp. 7148–7153.
5. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
6. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
7. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6569–6578.

8. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
9. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
10. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
11. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
12. Wang, X.; Li, K.; Shi, B.; Li, L.; Lin, H.; Wang, X.; Yang, J. Single shot multibox detector object detection based on attention mechanism and feature fusion. *J. Electron. Imaging* **2023**, *32*, 023032. [CrossRef]
13. Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; Lin, D. Libra r-cnn: Towards balanced learning for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 821–830.
14. Lim, J.S.; Astrid, M.; Yoon, H.J.; Lee, S.I. Small object detection using context and attention. In Proceedings of the 2021 International Conference on Artificial intelligence in information and Communication (ICAIIC), Jeju, Republic of Korea, 20–23 April 2021; pp. 181–186.
15. Wang, J.; Pan, Q.; Lu, D.; Zhang, Y. An Efficient Ship-Detection Algorithm Based on the Improved YOLOv5. *Electronics* **2023**, *12*, 3600. [CrossRef]
16. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. 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, Montreal, QC, Canada, 11–17 October 2021; pp. 2778–2788.
17. Zhao, Q.; Liu, B.; Lyu, S.; Wang, C.; Zhang, H. TPH-YOLOv5++: Boosting Object Detection on Drone-Captured Scenarios with Cross-Layer Asymmetric Transformer. *Remote Sens.* **2023**, *15*, 1687. [CrossRef]
18. Wang, F.; Wang, H.; Qin, Z.; Tang, J. UAV target detection algorithm based on improved YOLOv8. *IEEE Access* **2023**, *11*, 116534–125137. [CrossRef]
19. Shen, L.; Lang, B.; Song, Z. DS-YOLOv8-Based Object Detection Method for Remote Sensing Images. *IEEE Access* **2023**, *11*, 125122–125137. [CrossRef]
20. Liu, D.; Zhang, Y.; Zhao, Y.; Shi, Z.; Zhang, J.; Zhang, Y.; Ling, F.; Zhang, Y. AARN: Anchor-guided attention refinement network for inshore ship detection. *IET Image Process.* **2023**, *17*, 2225–2237. [CrossRef]
21. Zhou, W.; Peng, Y. Ship detection based on multi-scale weighted fusion. *Displays* **2023**, *78*, 102448. [CrossRef]
22. Zwemer, M.H.; Wijnhoven, R.G.; de With, P.H. Ship Detection in Harbour Surveillance based on Large-Scale Data and CNNs. In Proceedings of the VISIGRAPP (5: VISAPP), Madeira, Portugal, 27–29 January 2018; pp. 153–160.
23. Hu, C.; Zhu, Z.; Yu, Z. Ship Identification Based on Improved SSD. In Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering, Xiamen, China, 21–23 October 2022; pp. 476–482.
24. Shao, Z.; Wang, L.; Wang, Z.; Du, W.; Wu, W. Saliency-aware convolution neural network for ship detection in surveillance video. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 781–794. [CrossRef]
25. Li, Y.; Guo, J.; Guo, X.; Liu, K.; Zhao, W.; Luo, Y.; Wang, Z. A novel target detection method of the unmanned surface vehicle under all-weather conditions with an improved YOLOV3. *Sensors* **2020**, *20*, 4885. [CrossRef] [PubMed]
26. Chen, Z.; Liu, C.; Filaretov, V.F.; Yukhimets, D.A. Multi-scale ship detection algorithm based on YOLOv7 for complex scene SAR images. *Remote Sens.* **2023**, *15*, 2071. [CrossRef]
27. Chen, J.; Wang, G.; Liu, W.; Zhong, X.; Tian, Y.; Wu, Z. Perception reinforcement using auxiliary learning feature fusion: A modified yolov8 for head detection. *arXiv* **2023**, arXiv:2310.09492.
28. Huang, M.; Cai, Z. Steel surface defect detection based on improved YOLOv8. In Proceedings of the International Conference on Algorithms, High Performance Computing, and Artificial Intelligence (AHPCAI 2023), Yinchuan, China, 18–19 August 2023; Volume 12941, pp. 1356–1360.
29. Jin, Y.; Cai, L.; Cheng, K.; Wang, X.; Luo, C.; Jiao, S. PCB bare board defect detection based on improved YOLOv5s. In Proceedings of the 2023 CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS), Yibin, China, 22–24 September 2023; pp. 1–6.
30. Yang, W.; Wu, J.; Zhang, J.; Gao, K.; Du, R.; Wu, Z.; Firkat, E.; Li, D. Deformable convolution and coordinate attention for fast cattle detection. *Comput. Electron. Agric.* **2023**, *211*, 108006. [CrossRef]
31. Li, P.; Zheng, J.; Li, P.; Long, H.; Li, M.; Gao, L. Tomato maturity detection and counting model based on MHSA-YOLOv8. *Sensors* **2023**, *23*, 6701. [CrossRef] [PubMed]
32. Yang, G.; Wang, J.; Nie, Z.; Yang, H.; Yu, S. A lightweight YOLOv8 tomato detection algorithm combining feature enhancement and attention. *Agronomy* **2023**, *13*, 1824. [CrossRef]
33. Liu, W.; Chen, Y. IL-YOLOv5: A Ship Detection Method Based on Incremental Learning. In Proceedings of the International Conference on Intelligent Computing, Zhengzhou, China, 10–13 August 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 588–600.

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