remote sensing

MDPI

Article

HTC+ for SAR Ship Instance Segmentation

Tianwen Zhang and Xiaoling Zhang *

School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; twzhang@std.uestc.edu.cn
* Correspondence: xlzhang@uestc.edu.cn

Abstract: Existing instance segmentation models mostly pay less attention to the targeted characteristics of ships in synthetic aperture radar (SAR) images, which hinders further accuracy improvements, leading to poor segmentation performance in more complex SAR image scenes. To solve this problem, we propose a hybrid task cascade plus (HTC+) for better SAR ship instance segmentation. Aiming at the specific SAR ship task, seven techniques are proposed to ensure the excellent performance of HTC+ in more complex SAR image scenes, i.e., a multi-resolution feature extraction network (MRFEN), an enhanced feature pyramid net-work (EFPN), a semantic-guided anchor adaptive learning network (SGAALN), a context ROI extractor (CROIE), an enhanced mask interaction network (EMIN), a post-processing technique (PPT), and a hard sample mining training strategy (HSMTS). Results show that each of them offers an observable accuracy gain, and the instance segmentation performance in more complex SAR image scenes becomes better. On two public datasets SSDD and HRSID, HTC+ surpasses the other nine competitive models. It achieves 6.7% higher box AP and 5.0% higher mask AP than HTC on SSDD. These are 4.9% and 3.9% on HRSID.

Keywords: synthetic aperture radar; ship instance segmentation; HTC+; deep learning; convolutional neural network

Citation: Zhang, T.; Zhang, X. HTC+ for SAR Ship Instance Segmentation. *Remote Sens.* **2022**, *14*, 2395. https://doi.org/10.3390/rs14102395

Academic Editors: Zhihuo Xu, Jianping Wang and Yongwei Zhang

Received: 10 April 2022
Accepted: 13 May 2022
Published: 17 May 2022

1. Introduction

Ship surveillance has received widespread attention [1–4]. Synthetic aperture radar (SAR) is an active microwave sensor [5–7]. It works regardless of weather and light conditions, and is more suitable for ship monitoring than optical sensors [8]. Traditional methods [9,10] rely overly on hand-picked features, reducing model flexibility and migration. Now, more efforts are devoted to deep learning-based methods [11].

Most scholars use boxes to detect ships in the SAR community [12], but the instance segmentation at both box- and pixel-level has received less attention [13]. Moreover, Xu et al. [14] studied the dynamic detection of offshore wind turbines by spatial machine learning in SAR images, but offshore facilities and ships have different radar scattering characteristics. Some works [15–22] have studied SAR ship instance segmentation, but they mostly used models for generic objects directly, without considering the targeted characteristics of SAR ship objects, hindering further accuracy improvements and leading to poor segmentation performance in more complex SAR image scenes [23].

Thus, we propose HTC+ to explore better SAR ship instance segmentation. HTC is selected because it may be the best model [24]. For SAR ship mission, we enhance HTC using seven techniques for incremental performance to form HTC+. It is similar to the update from YOLOv3 [25] to YOLOv4 [26]. (1) A multi-resolution feature extraction network (MRFEN) is used to boost multi-scale feature description. (2) An enhanced feature pyramid network (EFPN) is designed to enhance better small ship search ability. (3) A semantic-guided anchor adaptive learning network (SGAALN) is proposed to optimize anchors. (4) A context ROI extractor (CROIE) is designed to boost background discrimination. (5) An enhanced mask interaction network (EMIN) is designed to boost multi-stage mask feature fusion. (6) A post-processing technique (PPT) using NMS [27] and Soft-NMS [28]

is used to reduce missed detections of densely moored ships. (7) A hard sample mining training strategy (HSMTS) [29] is used to deal with complex scenes and cases. Experiments are performed on two public datasets SSDD [13] and HRSID [15]. Results show that seven novelties contribute to improving accuracy; HTC+ offers the best accuracy over the other nine competitive models. The instance segmentation performance in more complex SAR image scenes becomes better. HTC+ offers 6.7% box AP and 5.0% mask AP increments on the vanilla HTC on SSDD; they are 4.9% and 3.9% on HRSID.

The contributions of our work are summarized as follows:

1. The vanilla HTC is designed for generic objects, with limited performance for SAR ships. Thus, we improve it to form its updated version HTC+ to achieve better SAR ship instance segmentation performance.
2. Aimed at the specific SAR ship task, we propose seven techniques (MRFEN, EFPN, SGAALN, CROIE, EMIN, PPT and HSMTS) to enable the excellent performance of HTC+. Here, the motivation and implementation of each technique are both related to the mission characteristics of SAR ships.
3. HTC+ offers a huge increase in accuracy building on HTC, which also surpasses the other nine competitive models.

The rest of this paper is arranged as follows. Section 2 reviews some related works. Section 3 introduces the methodology. Experiments are described in Section 4. Results are shown in Section 5. Ablation studies are made in Section 6. More discussions are introduced in Section 7. Finally, a summary of this paper is made in Section 8.

2. Related Works

In this section, we will review some commonly-used generic instance segmentation models in the computer vision community in Section 2.1. Afterwards, some existing SAR ship instance segmentation models will be introduced in Section 2.2.

2.1. Instance Segmentation

Mask R-CNN [30] is the most classic instance segmentation model, which designed a mask prediction branch on the basis of Faster R-CNN [31]. To measure mask quality, Mask Scoring R-CNN [32] added a scoring network to provide confidences of mask prediction. Cascade Mask R-CNN [33] designed a multi-stage detection and mask head with increasing intersection over union (IOU) thresholds to improve hypotheses quality and ease overfitting. PANet [34] added a bottom-top path aggregation to boost FPN's representation. To make full use of multi-scale features, Rossi et al. [35] proposed a novel region of interest (ROI) extraction layer (GROIE) for instance segmentation. YOLACT [36] is a real-time one-stage instance segmentation model, but its accuracy is poorer than two-stage ones. Furthermore, HTC [24] combined Cascade R-CNN and Mask R-CNN to leverage relationships between detection and segmentation, offering the state-of-the-art performance [37,38]. Therefore, we select it as our experimental baseline.

2.2. SAR Ship Instance Segmentation

Recently, many scholars in the SAR community have started to study SAR ship instance segmentation. Since the first public dataset called HRSID was released by Wei et al. [15] in 2020, various methods have emerged. Su et al. [16] proposed a high-quality network named HQ-ISNet for remote sensing object instance segmentation. They measured the model performance on SAR images; the results indicated the effectiveness of the proposed model. Yet, their model was just a mechanical borrowing from the computer vision community, without thought of appropriateness, hampering further performance improvements. Zhao et al. [17] proposed SA R-CNN which added attention mechanisms to boost accuracy, but the performance among complex scenes was still limited. Gao et al. [18] proposed an anchor-free model called CenterMask and a centroid-distance based loss to enhance benefits of ship feature learning, but such anchor-free models still cannot handle complex scenes

and cases [29]. HTC was applied to SAR ship instance segmentation by Zhang et al. [19], but such direct use led to limited accuracy for SAR ships.

In 2022, Fan et al. [20] designed an efficient instance segmentation paradigm (EISP) for interpreting SAR and optical images, which adopted transformers to extract features. Yet, this paradigm did not consider the targeted characteristics of SAR ships, with limited performance. Zhang et al. [21] designed a full-level context squeeze-and-excitation ROI extractor for SAR ship instance segmentation, but their method only considered extracting the optimized feature subset, and ignored improvements in other parts of the network, leading to limited ship segmentation performance in more complex SAR image scenes. Ke et al. [22] proposed a global context boundary-aware network to improve the positioning performance of the bounding box so as to achieve better segmentation effects, but they did not consider differences between segmentation tasks and detection tasks. Zhang et al. [23] improved Mask R-CNN further by using context information, and squeeze-and-excitation mechanism, but their network did not have adequate mask information interaction, leading to poor segmentation performance in more complex scenes.

In short, the above existing methods mostly used models for generic objects in the computer vision community directly. In other words, they did not consider the targeted characteristics of SAR ship objects, which hinders further accuracy improvements. Thus, we will research useful techniques in this paper to boost instance segmentation especially for SAR ships.

3. Methodology

Aiming at the specific SAR ship task, we explore ways to enhance each component's performance on the basis of the vanilla HTC [24] to achieve the progressive improvements to the overall performance, resulting in the evolution from HTC to HTC+. Our research thinking is similar to the evolution from YOLOv3 [25] to YOLOv4 [26] where YOLOv4 proposed five key techniques and adopted some useful tricks to enhance YOLOv3 further.

Figure 1 depicts HTC+ architecture. MRFEN is a backbone network to extract multi-resolution ship features. EFPN is to improve multi-scale feature representation. SGAALN is to learn anchor location and shape used in the region proposal network (RPN) [31] that is responsible for producing proposals. Classifier (CLS) is used to identify foreground and background. Regressor (REG) predicts proposal positions. CROIE is to map proposals from RPN into MRFEN's feature maps to extract feature subsets [21] for the box-mask prediction head. EMIN predicts box and mask. PPT post-processes outputs. HSMTS works only in training selected hard samples to handle complex scenes and cases.

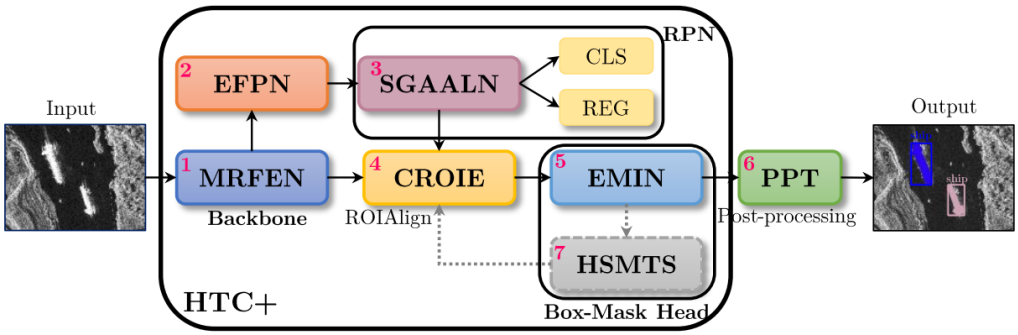

Figure 1. HTC+ architecture. (1) MRFEN denotes the multi-resolution feature extraction network; (2) EFPN denotes the enhanced feature pyramid network; (3) GAALN denotes the semantic-guided anchor adaptive learning network; (4) CROIE denotes the context regions of interest extractor; (5) EMIN denotes the enhanced mask interaction network; (6) PPT denotes the post-processing technique; (7) HSMTS denotes the hard sample mining training strategy. Seven novelties are marked by red numbers 1–7. The first five belong to the network architecture's improvements. The remaining two constitute extra tricks to boost performance further. Moreover, here, RPN denotes the region proposal network, CLS denotes classification, and REG denotes regression.

Moreover, for ease of reading, we summarize the following materials in Table 1. Next, we will introduce the seven components for incremental accuracy in detail in the following sub-sections Sections 3.1–3.7.

Table 1. Materials arrangement of the methodology.

Section ID	Sub-Section ID
Section 3.1. Multi-Resolution Feature Extraction Network (MRFEN)	Section 3.1.1. Multi-Resolution Feature Extraction (MRFF) Section 3.1.2. Multi-Scale Attention-Based Feature Fusion (MSAFF) Section 3.1.3. Atrous Spatial Pyramid Pooling (ASPP)
Section 3.2. Enhanced Feature Pyramid Network (EFPN)	Section 3.2.1. Content-Aware ReAssembly of Features (CARAFE) Section 3.2.2. Feature Balance (FB) Section 3.2.3. Feature Refinement (FR) Section 3.2.4. Feature Enhancement (FE)
Section 3.3. Semantic-Guided Anchor Adaptive Learning Network (SGAALN)	Section 3.3.1. Anchor Location Prediction (ALP) Section 3.3.2. Anchor Shape Prediction (ASP) Section 3.3.3. Feature Adaption (FA)
Section 3.4. Context Regions of Interest Extractor (CROIE)	Section 3.4.1. Concatenation Section 3.4.2. Channel Shuffle Section 3.4.3. Dimension Reduction Squeeze-and Excitation (DRSE)
Section 3.5. Enhanced Mask Interaction Network (EMIN)	Section 3.5.1. Global Feature Self-Attention (GFSA) Section 3.5.2. Adaptive Mask Feature Fusion (AMFF)
Section 3.6. Post-Processing Technique (PPT)	–
Section 3.7. Hard Sample Mining Training Strategy (HSMTS)	–

3.1. Multi-Resolution Feature Extraction Network (MRFEN)

Existing approach. The raw HTC adopted the high-to-low resolution paradigm as the network deepens to extract features, as shown in Figure 2a, e.g., ResNet [39] and ResNeXt [40], i.e., the network depth is inversely proportional to the resolution. Still, this paradigm is not well-suited to SAR ship tasks, considering the two aspects below.

On the one hand, four stages in Figure 2a extract multi-scale features equally, i.e., the same number of conv blocks [e.g., 4 in Figure 2a]. Yet, the ship size distribution of existing datasets is uneven as in Figure 3a,b, i.e., small ships are far more than large ones. The main reason for this phenomenon is that SAR is a "bird-eye" remote sensing earth observation tool that is different from "person-eye" natural scene cameras. Thus, one should treat them differently. Otherwise, a huge performance imbalance between small ships and large ships will occur. We think that one should arrange heavy networks for small ship detection because they are more difficult to detect for fewer feature pixels; for contrast, one should use light networks for large ship detection because they are easier to detect due to their clearer features.

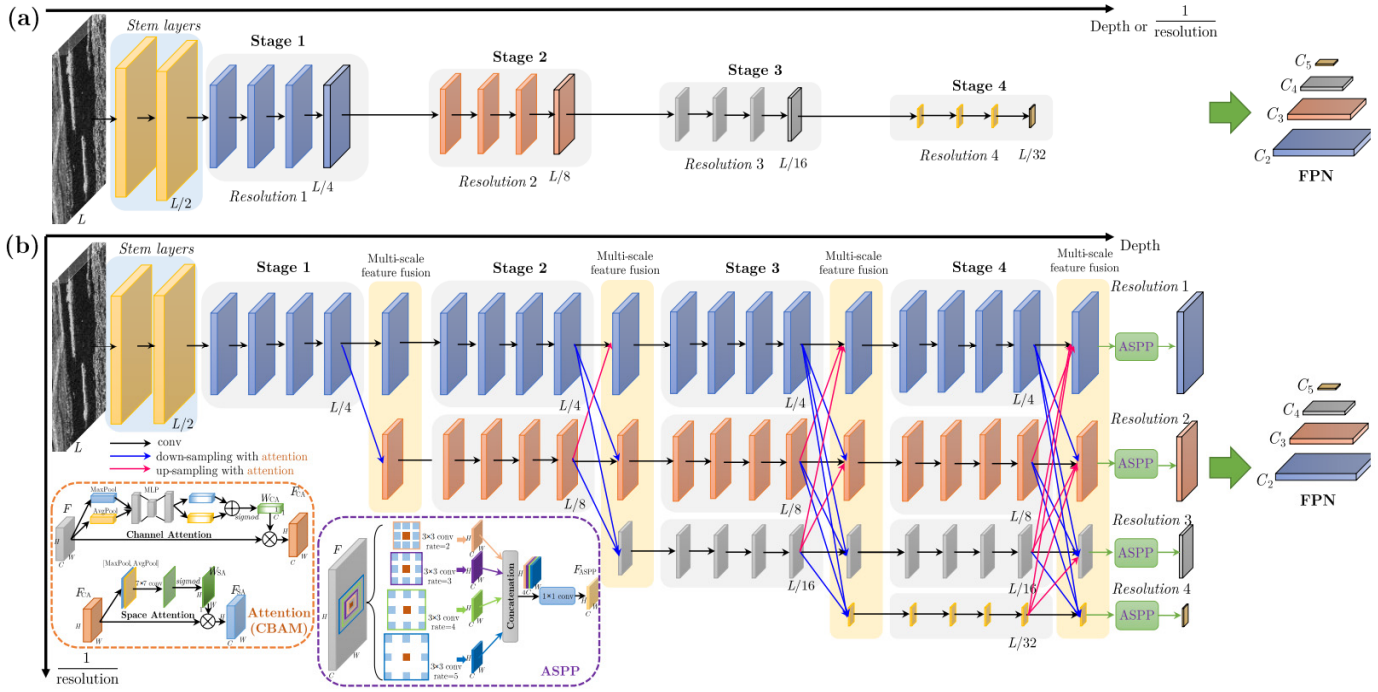

Figure 2. Different backbone networks for feature extraction. (**a**) Existing approach: the backbone network of HTC. (**b**) Proposed approach: the backbone network of HTC+ (MRFEN).

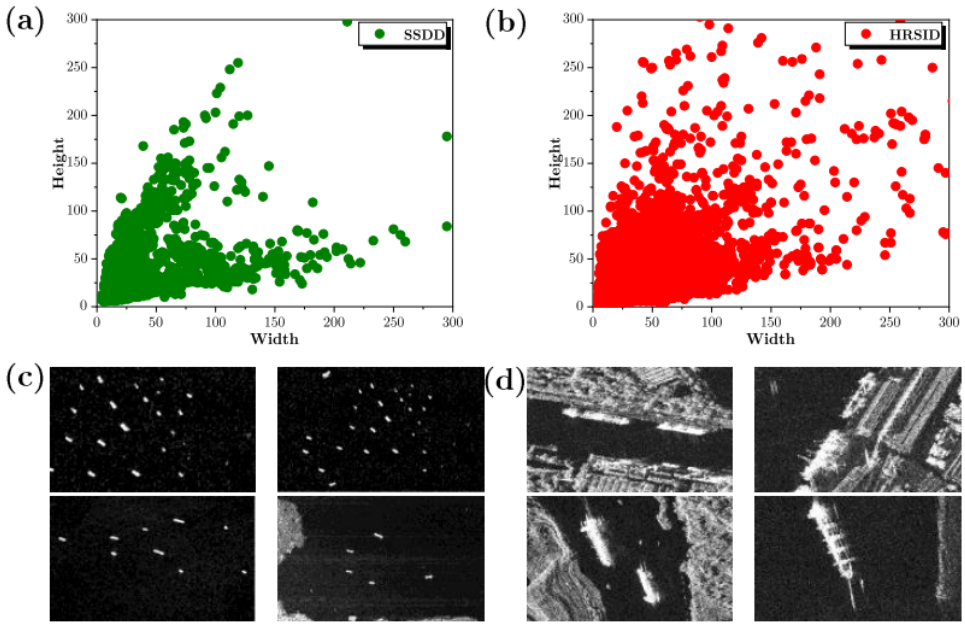

Figure 3. Multi-scale SAR ships. (**a**) Ship size uneven distribution in SSDD. (**b**) Ship size uneven distribution in HRSID. (**c**) Small ships. (**d**) Large ships.

On the other hand, the network backend in Figure 2a is lacking in rich high-resolution representations, i.e., the spatial position information is lost to some degree. This is not conducive to handling position-sensitive vision problems [37], e.g., SAR ship instance segmentation. Thus, one should maintain high-resolution position representations totally across the whole conv process. Moreover, we think that the strong coupling between the depth and resolution directions potentially limits feature description capacity. Signs [41,42] have indicated that decoupling them would help to improve the performance of pixel-sensitive tasks.

Proposed approach. Given the above, we designed MRFEN to extract more precise position representations and richer semantic representations. Moreover, although the multi-resolution approach offers limited accuracy gains for small ships, it can improve the instance segmentation performance of very large ships when the high-resolution mode is used. Figure 2b shows its architecture. MRFEN has three design concepts, i.e., (1) multi-resolution feature extraction (MRFF) in Section 3.1.1, (2) multi-scale attention-based feature fusion (MSAFF) in Section 3.1.2, and (3) atrous spatial pyramid pooling (ASPP) in Section 3.1.3.

3.1.1. Multi-Resolution Feature Extraction (MRFF)

We retain high-resolution representations across the entire system, including, the uppermost resolution-1 branch. This can boost instance segmentation of small ships due to more position information and heavier network parameters. The resolution-2 branch starts from the stage-2; the resolution-3 branch starts from the stage-3; the resolution-3 branch starts from the stage-4. This leverages lighter networks for larger ships so as to adapt to their easier detection. Consequently, the network depth and resolution are decoupled smoothly, which can enable networks to optimize their parameters among a larger search space so as to further enhance fitting or learning capacity. Briefly, the above can be described by

$$\begin{aligned} \mathcal{N}_{11} \rightarrow \mathcal{N}_{21} \rightarrow \mathcal{N}_{31} \rightarrow \mathcal{N}_{41} \\ \searrow \mathcal{N}_{22} \rightarrow \mathcal{N}_{32} \rightarrow \mathcal{N}_{42} \\ \searrow \mathcal{N}_{33} \rightarrow \mathcal{N}_{43} \\ \searrow \mathcal{N}_{44} \end{aligned} \tag{1}$$

where $\mathcal{N}_{sr}$ denotes the sub-network of the s-th stage and the r-th resolution, $\rightarrow$ denotes the conv process, and $\searrow$ denotes the down-sampling process. Different from image pyramid in [43], the low-resolution in MRFEN comes from the previous high-resolution down-sampling, rather than the down-sampling on the input image. This is because feature maps from the front-end high-resolution sub-network are more representative.

3.1.2. Multi-Scale Attention-Based Feature Fusion (MSAFF)

There are no direct interactions between different resolution branches after the network depth and resolution are decoupled. This hampers network information flow, possibly increasing the risk of overfitting of their separate local optimization. Moreover, training within their own closed cyberspace may also slow down the training convergence speed, declining performance. Therefore, it is essential to perform multi-scale feature fusion. (In this paper, the resolution and the scale share the same meaning.) Integrating features with different scales, the down-sampling and up-sampling were widely adopted [37,44,45]. Still, different from these authors, we suggest to first use an attention module for a feature refinement, and then execute the down-sampling and up-sampling. This can enable more valuable features to be transmitted to another branch so as to avoid possible negative interferences. Taking $\mathcal{N}_{42}$ in Equation (1) as an example, we get its feature maps $\mathcal{F}_{42}$ by

$$\mathcal{F}_{42} = \mathcal{F}_{32} + DownSamp^{2\times}(f_{attention}(\mathcal{F}_{31})) + UpSamp^{2\times}(f_{attention}(\mathcal{F}_{33})) \tag{2}$$

where $\mathcal{F}_{sr}$ denotes the feature maps of $\mathcal{N}_{sr}$, $DownSamp^{n\times}$ denotes the n times down-sampling, $UpSamp^{n\times}$ denotes the n times up-sampling, and $f_{attention}$ denotes the refinement operation using an attention module. We implement $f_{attention}$ by a convolutional block attention module (CBAM) [46] with channel attention and space channel attention. One can also use other advanced attention modules [47] for better performance. In this work, we select CBAM, because it is the most famous and has been used by many scholars in the SAR community [17].

The network architecture of CBAM is shown in the orange dashed box in Figure 2b. Let its input be $F \in \mathbb{R}^{H\times W\times C}$ where H and W are the height and width of feature maps and C is the channel number. Then the channel attention is responsible for generating a channel-dimension weight matrix $W_{CA} \in \mathbb{R}^{1\times 1\times C}$ to measure the important levels of C

channels; the space attention is responsible for generating a space-dimension weight matrix $W_{SA} \in \mathbb{R}^{H\times W\times 1}$ to measure the important levels of space-elements across the entire $H \times W$ space. They range from 0 to 1 by a sigmoid activation. The result of the channel attention is denoted by $F_{CA} = F \cdot W_{CA}$. The result of the space attention is denoted by $F_{SA} = F_{CA} \cdot W_{SA}$. See [46] for CBAM's details.

3.1.3. Atrous Spatial Pyramid Pooling (ASPP)

Although the multi-scale attention-based feature fusion offers some other resolution responses from other branches, these kinds of responses are still limited among the total responses. Thence, we adopt the atrous spatial pyramid pooling (ASPP) [48] to deal with this problem. Its network architecture is depicted in the purple dashed box in Figure 2b. ASPP utilizes atrous convs [49,50] with different dilated rates to achieve multi-resolution feature responses in the single-resolution branch. It is described by

$$F_{\text{ASPP}} = f_{1\times 1}\left(\left[f^2_{3\times 3}(F), f^3_{3\times 3}(F), f^4_{3\times 3}(F), f^5_{3\times 3}(F)\right]\right) \tag{3}$$

where F denotes the input, F_{ASPP} denotes the output, $f^r_{3\times 3}$ denotes a 3×3 conv with a dilated rate of r, and $f_{1\times 1}$ denotes a 1×1 conv for channel reduction, i.e., from four atrous convs concatenation $4C$ to the raw C of F. In this way, different dilated rates will enable different resolution responses, as well yielding different scope contexts. We set four dilated rates for the accuracy-speed trade-off. More might offer better performance but must sacrifice speed. Different from [48], four dilated rates are set to 2, 3, 4 and 5 because of the small size of the low-solution branch ($L/32 \times L/32$). Especially, ASPP can also allow our MRFEN to enlarge receptive fields so as to receive more ship surrounding context information. This is conducive to alleviating background interferences, e.g., blur edges, sidelobes, ship wakes, speckle noise (SAR imaging mechanisms), tower crane [8], and inshore facilities, as in Figure 4.

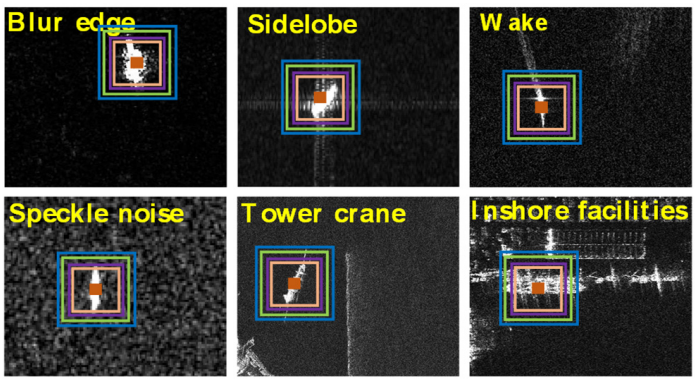

Figure 4. SAR ships and ships' surrounding contexts. Boxes with different colors and sizes denotes the atrous convs with different dilated rates.

Finally, the outputs of ASPP with different resolution levels will constitute the inputs of FPN (C_2, C_3, C_4 and C_5). It is also different from the previous network in Figure 2a that makes the last layers of all stages constitute the inputs of FPN. Figure 2b performs better than Figure 2a because the former offers richer high-resolution position representation and richer low-resolution semantic representation at the same time among each stage.

3.2. Enhanced Feature Pyramid Network (EFPN)

Existing approach. The vanilla HTC followed the standard FPN paradigm [51] to ensure multi-scale performance as shown in Figure 5a. In Figure 5, C_2, C_3, C_4 and C_5 are the inputs of FPN which are from the backbone network as shown in Figure 2. However, this standard FPN offers limited performance for SAR ship instance segmentation from the following three aspects.

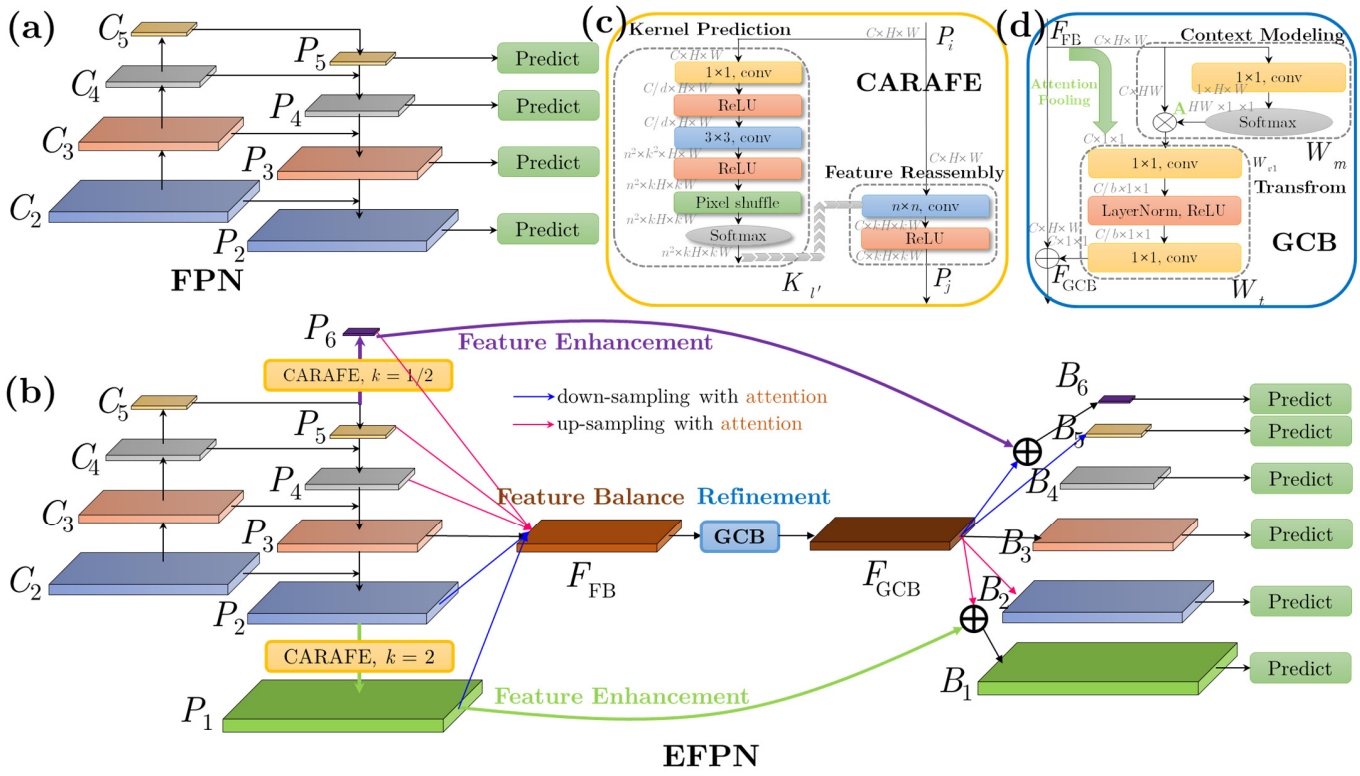

Figure 5. Feature pyramid networks. (**a**) Existing approach: FPN of HTC. (**b**) Proposed approach: EFPN of HTC+. (**c**) CARAFE implementation. (**d**) GCB implementation.

Firstly, on account of the huge resolution difference, e.g., 1 m resolution for TerraSAR-X and 20 m resolution for Sentinel-1, ships in SAR images present a huge scale difference as shown in Figure 6a,b. This situation is called the cross-scale effect [52], e.g., the extremely small ships in Figure 6c vs. the extremely large ships in Figure 6d. The standard FPN was designed for the natural object detection, e.g., COCO [53] and PASCAL [54] datasets. These datasets are with a relatively small scale difference. Thus, the raw FPN has some challenges to handle such cross-scale problems due to its limited FPN levels. From the clustering results of *K*-means, the raw four levels P_2, P_3, P_4 and P_5 are inferior to six levels in terms of the network multi-scale feature learning ability. The mean IOU of the former is 0.5913, lower than that of the latter 0.6490. Thus, we use more levels to deal with this special SAR ship cross-scale instance segmentation.

Secondly, small ships always constitute the majority among existing datasets for the characteristics of the "bird-eye" view of SAR. The raw bottom-level P_2 is with limited searching ability for small ships in Figure 6c, because small ships are diluted after multiple down-sampling operations (from L to $L/4$) due to their faint spatial features. As a result, a large number of small ships will be missed, reducing HTC's overall performance. Thus, we suggest to generate a lower level P_1 to handle this problem, because lower levels offer richer spatial position information, which is beneficial for small ship instance segmentation.

Thirdly, although the original top-level P_5 may detect the extremely large ships in Figure 6d successfully by using a rectangle bounding box, it is still with limited pixel-segmentation performance. Different parts of the ship hull have different materials, resulting in differential radar electromagnetic scatterings. This makes the pixel brightness distribution of the ship in a SAR image extremely uneven. This will bring huge difficulties to classifiers for their effective pixel-level discrimination. Therefore, we suggest to generate a higher level P_6 to handle this problem, because high levels can offer more semantic information by shrinking large ships to reach the purpose of removing the ship hull's internal "black" pixels; then in the mask recovery process, the nearest neighbor interpolation can fill

those internal "black" pixels using correct predicted ship "white" pixels, so as to achieve better segmentation performance, as shown in Figure 6d.

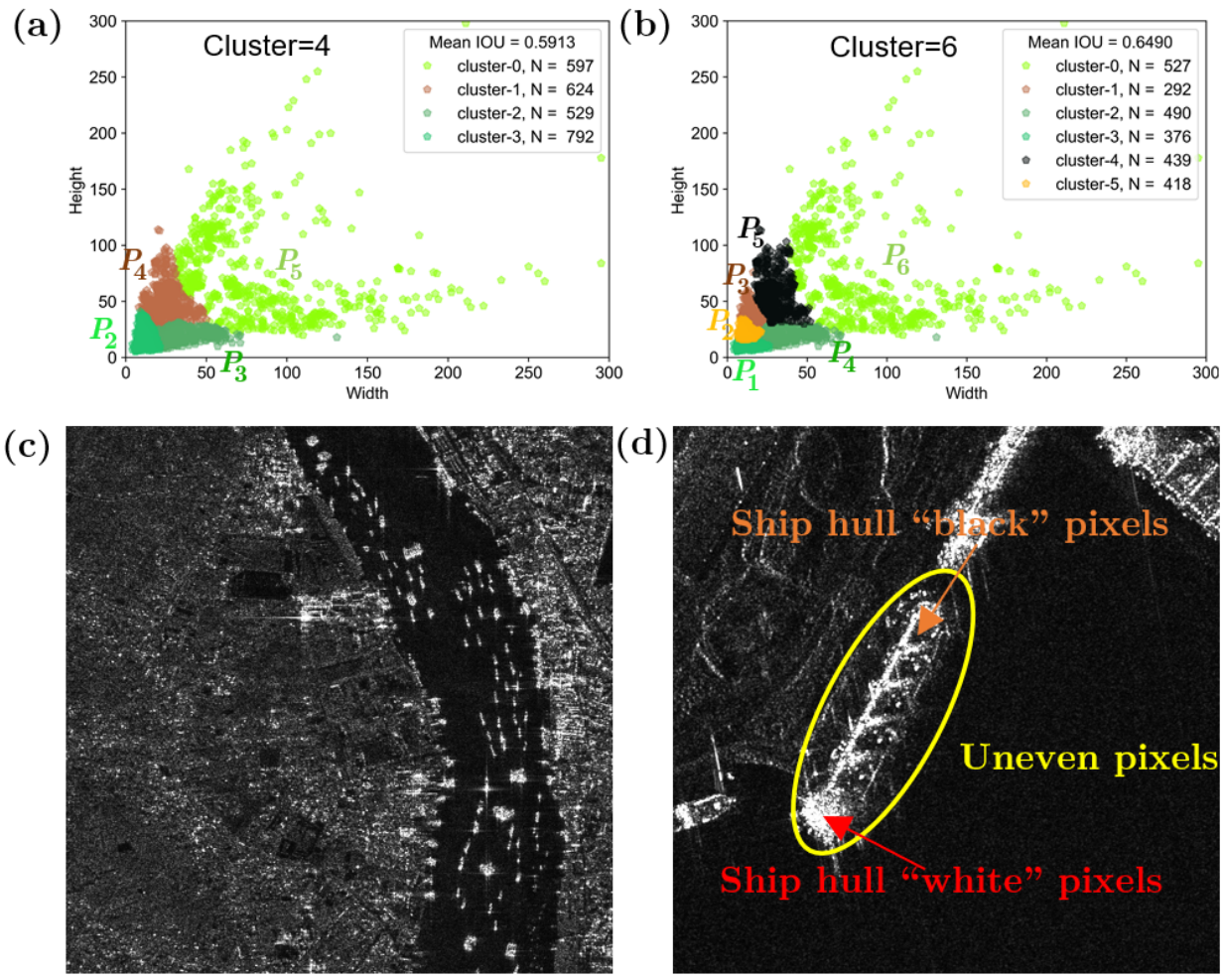

Figure 6. (**a**) Cluster results of four levels of HTC. (**b**) Cluster results of six levels of HTC+. Here, K-means is used for more intuitive presentation. (**c**) An SAR image with extremely small ships. (**d**) An SAR image with extremely large ships. Here, (**c**,**d**) have a huge scale difference due to different resolutions, i.e., cross-scale.

Proposed approach. Given the above, we propose an enhanced feature pyramid network (EFPN) to enhance multi-scale instance segmentation. Figure 5b shows its architecture. EFPN has four design concepts, i.e., (1) content-aware reassembly of features (CARAFE) in Section 3.2.1, (2) feature balance (FB) in Section 3.2.2, (3) feature refinement (FR) in Section 3.2.3, and (4) feature enhancement (FE) in Section 3.2.4.

3.2.1. Content-Aware ReAssembly of Features (CARAFE)

We draw lessons from the advanced Content-Aware ReAssembly of Features (CARAFE) [55] to generate an extra higher top-level P_6 and an extra lower bottom-level P_1. Note that the raw CARAFE did not offer a down-sampling operation; here, we expand it by adding an extra hyper-parameter k. $k = 2$ denotes the up-sampling. $k = 1/2$ denotes the down-sampling. (i) *Generate* P_1. Wang et al. [55] have confirmed that CARAFE was superior to the nearest neighbor and the bilinear interpolation which both focus on the subpixel neighborhoods, failing to capture the richer semantic information required by dense prediction tasks, and it was also superior to the adaptive *deconv* [56] which uses a fixed kernel for all samples, resulting in limited receptive fields. CARAFE can enable instance-specific

content-aware handling, offering a large field of view, which can generate adaptive kernels on-the-fly. It can also aggregate global contextual information, enabling preferable performance for object detection [55]. Thence, it is adopted to generate P_1. (ii) *Generate* P_6. Shelhamer et al. [57] pointed out that the simple max-pooling would increase the risk of feature loss. Thus, to leverage CARAFE's advantage, we extend it for more effective feature down-sampling. The above is described by

$$\begin{aligned} P_1 &= \text{CARAFE}^{\times 2}(P_2) \\ P_6 &= \text{CARAFE}^{\times \frac{1}{2}}(P_5) \end{aligned} \tag{4}$$

Here, $\text{CARAFE}^{\times k}$ denotes the k times down/up-sampling using CARAFE. It needs to be noted that one can follow the above operation to generate more levels, e.g., P_0, P_7, but the trade-off between speed and accuracy should be carefully considered.

Figure 5c shows the implementation of CARAFE. It contains a kernel prediction process and a feature reassembly one. (i) The former is to predict an adaptive k times down/up-sampling kernel $K_{l'}$ corresponding to the l' location of feature maps from the original l location. The kernel size is $n \times n$ which means $n \times n$ neighbors of the location l. Here, n is set to 5 empirically, the same as the raw report [55]. In other words, CARAFE considers the surrounding 5 pixels for down/up-sampling interpolation ($5 \times 5 = 25$ pixels). Moreover, the contribution weights of these 25 pixels are obtained by the adaptive learning. During the kernel prediction process, a 1×1 conv is to compress channel to refine salient features, where the compression ratio d is set to 4, i.e., the raw 256 channels are compressed to 64 ones. This can not only reduce the calculation burden, but also ensure the benefits of predicted kernels [58]. One 3×3 conv is to encode contents whose channel number is $n^2 \times k^2$ where k denotes the down/up-sampling ratio (i.e., from $H \times W$ to $kH \times kW$). The dimension transformation is completed using a pixel shuffle operation. Each reassembly kernel is normalized by a softmax function spatially so as to reflect the weight of each sub-content. Finally, the learned kernel $K_{l'}$ will serve as the kernel for the subsequent feature reassembly process. (ii) The latter is a simple $n \times n$ conv, but its conv kernel parameters are determined by $K_{l'}$. In the above way, the resulting down/up-sampling feature maps have the ability of content perception, yielding the better feature representation. More details can be found in [55].

3.2.2. Feature Balance (FB)

Cross-scale ships predicted in different FPN levels have a huge feature imbalance [59] especially with the increase of levels. This imbalance also potentially leads to unstable training coming from the huge number gap between small ships and large ships. Thence, we follow the practice from [59] to balance ship features with huge differences, i.e.,

$$\begin{aligned} F_{\text{FB}} = \tfrac{1}{6}\{&UpSamp^{8\times}(f_{attention}(P_6)) + UpSamp^{4\times}(f_{attention}(P_5)) + UpSamp^{2\times}(f_{attention}(P_4)) \\ &+P_3 + DownSamp^{2\times}(f_{attention}(P_2)) + DownSamp^{4\times}(f_{attention}(P_1))\} \end{aligned} \tag{5}$$

Here, to fully leverage the advantage of the attention in Equation (2), before up/down-sampling, each level is also processed by CBAM to further increase representation power. Moreover, we rescale all levels into the P_3 level empirically, because it is located at the middle of the pyramid, provided with both richer position information and semantic information. It can consider both lower levels P_1, P_2 and higher levels P_4, P_5, P_6. P_4 is also a middle level of the pyramid; still it is not used as the rescaled level, because we hold the view that the network should better contain more space position features for better small ship instance segmentation. Once all levels are rescaled to the same level, an average operation is performed for their balanced feature fusion. In this way, the resulting condensed multi-scale features contain balanced semantic features and position features from various resolutions. Finally, large ship features and small ones can complement

each other to facilitate the information flow, alleviating feature imbalance and promoting network smooth training.

3.2.3. Feature Refinement (FR)

To recover a more robust FPN, we also adopt a global context block (GCB) [60] to refine the condensed multi-scale features F_{FB} further. Such practice is in fact motivated by Libra R-CNN [59]. However, their used non-local block [61] only can capture spatial dependence, but the channel dependence is neglected. Therefore, we adopt the more advanced GCB to reach this aim, achieving global feature self-attention and meeting channel squeeze-and-excitation (SE) [62] simultaneously. Figure 5d shows its architecture. GCB contains a context modeling module and a transform module. (i) The former first uses 1×1 conv W_k and a *softmax* activation to generate the attention weights A; then, conducts a global attention pooling to achieve the global context features, i.e., from $C \times H \times W$ to $C \times 1 \times 1$. It is equivalent to the global average pooling in SE [62], but the average operation is replaced with the adaptive operation here. (ii) The latter is similar to SE, but before the rectified linear unit (ReLU) activation, the output of the 1×1 squeeze conv W_{v1} is first normalized to ensure better generalization, equivalent to the regularization of batch normalization (BN) [63]. Here, to refine more salient features of six FPN levels, the squeeze ratio r is set to 6. The last 1×1 conv W_{v2} is used to transform bottlenecks to capture channel-wise dependencies. Finally, an element-wise matrix addition is used for feature fusion. More details can be found in [60].

3.2.4. Feature Enhancement (FE)

To reduce the risk of feature loss from the boundary levels P_1 and P_6 due to their relatively large up/down-sampling ratios, we also add extra two skip connections for their feature enhancement while recovering them, i.e.,

$$\begin{aligned} B_1 &= P_1 + UpSamp^{4\times}(f_{attention}(F_{\mathrm{FB}})) \\ B_6 &= P_6 + DownSamp^{8\times}(f_{attention}(F_{\mathrm{FB}})) \end{aligned} \tag{6}$$

where B_j denotes the i-th level of the recovered FPN. Here, we obtain the other remaining levels using the reverse operation of Equation (5). Finally, the recovered FPN B_i will be able to predict cross-scale SAR ships in a more elegant and stable paradigm.

3.3. Semantic-Guided Anchor Adaptive Learning Network (SGAALN)

Existing approach. The raw HTC used the classic anchor generation scheme [31] as in Figure 7a. This scheme uniformly arranges dense anchors with fixed shapes to every location among an image. However, it is not suitable for SAR ship instance segmentation.

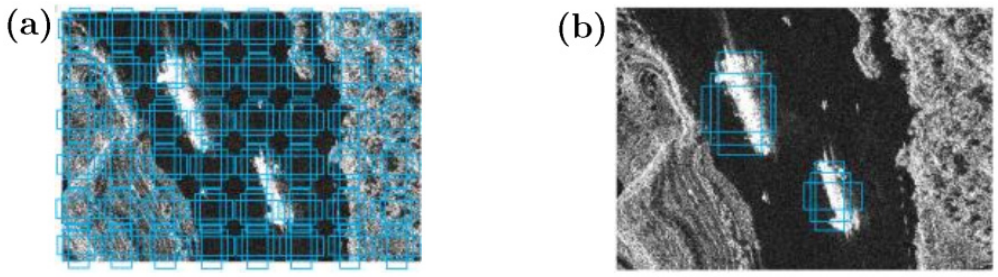

Figure 7. Different anchor generation schemes. (**a**) Existing approach: dense and shape-fixed anchors of HTC. (**b**) Proposed approach: Sparse and shape-adaptive anchors of HTC+. Blue boxes denote anchor boxes.

On the one hand, ships in SAR images exhibit a sparse distribution due to the characteristic of SAR "bird-eye" remote sensing view, e.g., there are only 2.12 ships on average among one image in the SSDD dataset. The uniform and dense anchor allocation to everywhere in a SAR image will potentially increase false alarms; additionally, numerous anchors are redundant, causing a heavier computational burden. Thus, one should better first adaptively predict possible positions where ships are more likely to occur; then arrange anchors on these possible positions to handle the above problem. On the other hand, the raw anchors with fixed shapes (width, height and aspect ratio) are not conducive to multi-scale prediction, with slower training speed. The hand-crafted preset anchors with fixed shapes are not in line with real ships with changeable shapes, reducing multi-scale performance. Although one can draw support from the *K*-means clustering on the specific dataset for better initial anchors [25], it is troublesome when this practice is applied to more datasets. Moreover, the initial anchors are still with fixed shapes, not resolving substantive issues. Thus, we should adaptively predict anchor shapes to handle the above problem.

Proposed approach. Given the above, we establish a semantic-guided anchor adaptive learning network (SGAALN) to achieve the adaptive anchor location prediction and the adaptive anchor shape prediction. The execution visual effect of SGAALN is shown in Figure 7b. Here, we leverage high-level semantic features to reach this aim, because they enable higher anchor quality than low-level ones [29]. SGAAL has three design concepts, i.e., (1) anchor location prediction (ALP) in Section 3.3.1, (2) anchor shape prediction (ASP) in Section 3.3.2, and (3) feature adaption (FA) in Section 3.3.3.

3.3.1. Anchor Location Prediction (ALP)

We use a 1×1 conv W_L whose channel number is set to 1 for the adaptive anchor location prediction, as shown in Figure 8a. This conv works on the inputted semantic features denoted by Q. The resulting feature maps are with a $H \times W \times 1$ dimension. $H \times W$ is the whole 2D space. This feature map is then activated by a *sigmoid* function to achieve the probability of ship occurrence $P_{ship} \in [0,\ 1]$ at the (x, y) position cross the whole $H \times W$ 2D space. When P_{ship} is bigger than the preset threshold ϵ_t, this (x, y) position will generate anchors; otherwise, this position will be removed where no anchors are generated. Here, ϵ_t is set to 0.01 empirically according to the experimental results in Section 6.3.

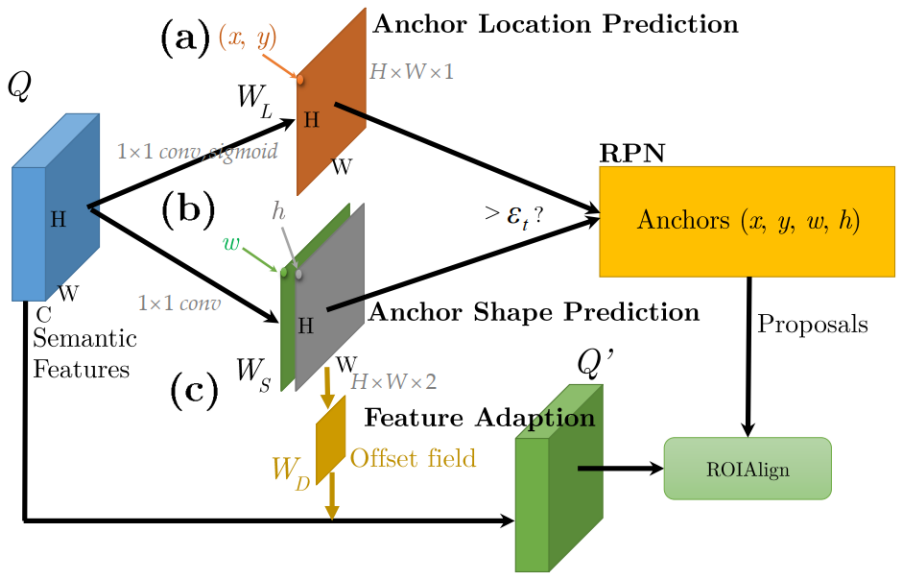

Figure 8. SGAALN architecture. (**a**) Anchor location prediction. (**b**) Anchor shape prediction. (**c**) Feature adaption.

3.3.2. Anchor Shape Prediction (ASP)

We arrange another one 1×1 conv W_S whose channel number is set to 2 for the adaptive anchor shape prediction, as shown in Figure 8b. This conv works on the input semantic features denoted by Q as well. The resulting feature maps are with a $H \times W \times 2$ dimension. This is because we need to estimate two parameters, i.e., the anchor width w and height h. Note that the anchor shape prediction works on all positions of the whole $H \times W$ 2D space in consideration of an easy network implementation. Finally, according to the previously obtained anchor prediction locations, the redundant shape predictions are filtered. As a result, the adaptive anchor parameter (x, y, w, h) is achieved which will be inputted to RPN for classification and regression.

3.3.3. Feature Adaption (FA)

Due to the cross-scale effect of SAR ships, the adaptively predicted anchors will also exhibit a huge shape difference. Yet, this will bring about a huge encoded content difference on the inputted semantic features when they are pooled for the subsequent box and mask prediction. Thus, large anchors should encode content in a large region, while small anchors should have a small region, accordingly. However, the raw semantic features Q do not meet this point because it is designed for fixed-shape anchors. Coincidentally, the existing deformable convs [64] can offer an effective solution for this issue, where the previously learned anchor shape (w, h) exactly corresponds to deformable conv kernel's bias (i.e., the offset field). Thence, we use a deformable convs W_D to process the inputted semantic features Q for such feature adaption, i.e.,

$$q_i^{'} = W_D(q_i, w_i, h_i) \tag{7}$$

where w_i and h_i are the i-th position anchors' width and height $i \in \mathbb{R}^{H \times W}$, $q_i \in Q$, and $q_i^{'} \in Q'$. Q' denotes the output of feature adaption. The optimized anchors enable better proposals to extract ROI feature subsets using ROIAlign.

3.4. Context Regions of Interest Extractor (CROIE)

Existing approach. The raw HTC followed the standard two-stage ROI extractor (ROIE) of Mask R-CNN [30] to extract feature subsets of ROIs for the subsequent mask prediction, as shown in Figure 9a. That is, the bounding box prediction is first conducted, and then, the mask prediction is performed in the resulting $h \times w$ box. However, this practice is still with limited SAR ship mask prediction performance. On the one hand, the mask prediction relies heavily on the box prediction. If the offered boxes are not accurate, the mask prediction must become poor. From Figure 9a, the features for mask prediction exist in the corresponding box with limited receptive fields, reducing the global field of vision. This will decline segmentation performance improvements. One the other hand, as in Figure 4, due to the special SAR imaging mechanism, SAR ships have many complex surroundings outside the compact box, e.g., blur edges, sidelobes, ship wakes, speckle noise, tower crane, and inshore facilities. They will bring some non-negligible effects for mask prediction. Thus, the compact box makes it impossible for mask prediction to observe more ship backgrounds, e.g., ship-like pixel noise and ship wakes. Although the box can eliminate the cross-sidelobe deviating from the ship center too far, a few sidelobe and noise pixels in the box can make it difficult to ensure segmentation learning benefits. Thus, it is necessary to expand the receptive field to explicitly find out the clear boundary between ship and its context surrounding.

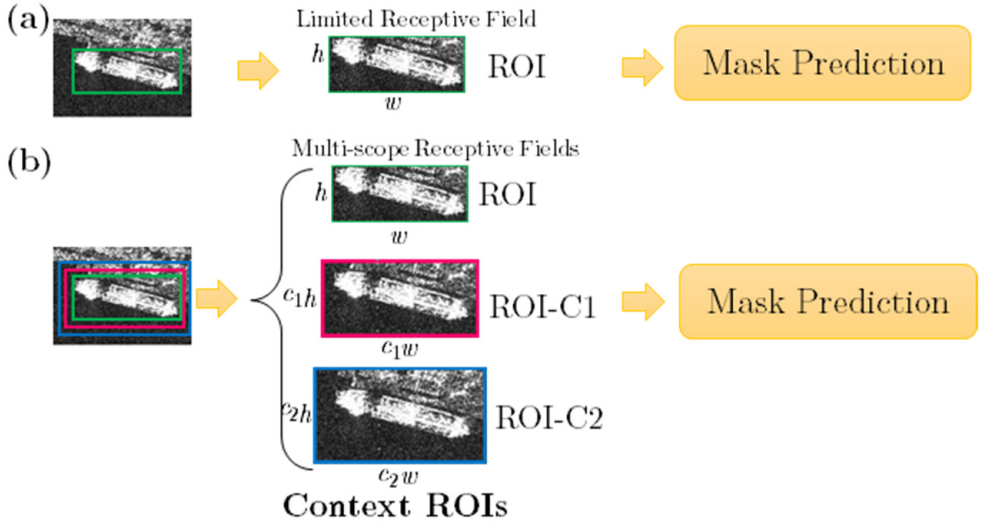

Figure 9. Different ROI extractors (ROIEs). (**a**) Existing approach: ROIE of HTC. (**b**) Proposed approach: CROIE of HTC+.

Proposed approach. Given the above, we design a context ROI extractor (CROIE) to add multi-scope contexts to the box for better mask prediction, as shown in Figure 9b. We arrange two different scope contexts denoted by ROI-C1 with a size of $c_1w \times c_1h$ and ROI-C2 with a size of $c_2w \times c_2h$. Here, c_1 and c_2 ($c_2 > c_1 > 1$) are two amplification factors which are set to 1.5 and 2.0, respectively, according to experiments in Section 6.4. This idea is motivated by Kang et al. [65]; however, differently, we consider multi-scope contexts. Moreover, we do not use more context ROIs, e.g., ROI-C3, considering the trade-off of speed and accuracy. Figure 10 shows CROIE's implementation. CROIE has three design concepts, i.e., (1) concatenation in Section 3.4.1, (2) channel shuffle in Section 3.4.2, and (3) dimension reduction squeeze-and excitation (DRSE) in Section 3.4.3.

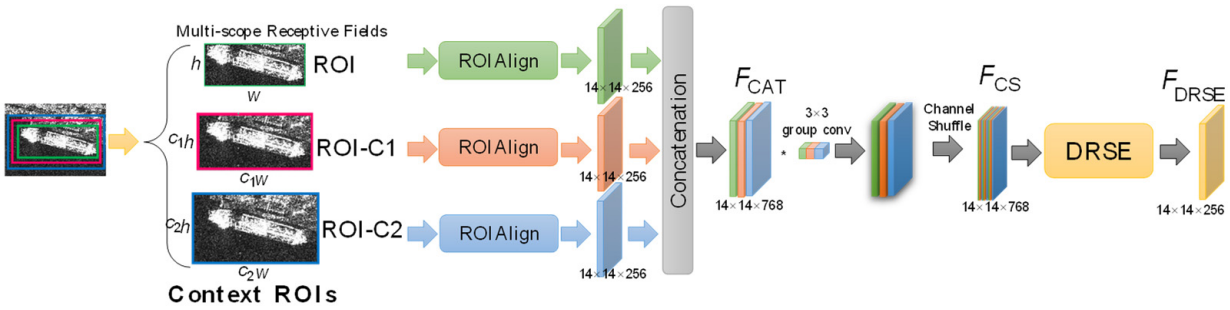

Figure 10. Implementation of CROIE.

3.4.1. Concatenation

We concatenate three feature subsets after ROIAlign. We find that the feature concatenation performs better than the feature adding, because the former achieves feature reuse for better deep supervision. Then, we use a 3×3 group conv to refine them for subsequent operations where the group division factor g is set to 3. The above is described by

$$F_{\mathrm{CAT}} = f_{3\times3,3}([\mathrm{ROIAlign}(\mathrm{ROI}), \mathrm{ROIAlign}(\mathrm{ROI}-\mathrm{C1}), \mathrm{ROIAlign}(\mathrm{ROI}-\mathrm{C2})]) \quad (8)$$

where $f_{3\times3,3}$ denotes the 3×3 group conv whose group division factor is 3, and F_{CAT} denotes the output of concatenation.

3.4.2. Channel Shuffle

To reduce the effect of feature collaboration consistency in the single ROI, we also shuffle the resulting features F_{CAT} along the channel dimension to enable more powerful representation. The result is denoted by F_{CS}.

3.4.3. Dimension Reduction Squeeze-and-Excitation (DRSE)

To balance allocation contributions of different ROIs with different context scopes, we also design a dimension reduction squeeze-and-excitation (DRSE) block, an extended version of SE [62] (the raw SE did not achieve dimension reduction), to model channel correlation. It can suppress useless channels and highlight valuable ones meanwhile reducing channel dimension, which reduces the risk of the training oscillation due to excessive irrelevant backgrounds. Consequently, moderate contexts can be offered for mask prediction. Figure 11 shows DRSE's implementation. In the collateral branch, the global average pooling is used to achieve the global spatial information; a 1×1 conv with a sigmoid activation is used to squeeze channels to focus on important ones. The squeeze ratio p is set to 3 ($256 \times 3 \rightarrow 256$). In the main branch, the input channel's number is reduced directly using a 1×1 conv with a ReLU activation. The broadcast element-wise multiplication is used for compressed channel weighting. DRSE will model channel correlation of the input feature maps in a reduced dimension space. Then, it leverages the learned weights from the reduced dimension space to pay attention to the important features of the main branch. In this way, the potential information loss from the crude dimension reduction is avoided. The above is described by

$$F_{\mathrm{DRSE}} = \mathrm{ReLU}(conv_{1\times1}(F_{\mathrm{CS}})) \odot \sigma(conv_{1\times1}(GAP(F_{\mathrm{CS}}))) \tag{9}$$

where F_{CS} denotes the input, i.e., the output of channel shuffle, F_{DRSE} denotes the output, σ denotes the sigmoid function, and $\odot$ denotes the broadcast element-wise multiplication.

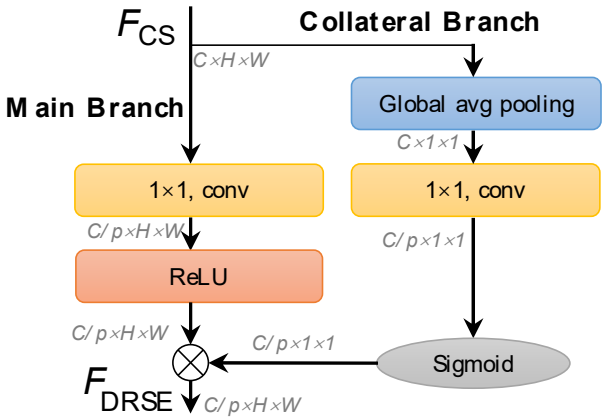

Figure 11. Implementation of DRSE.

3.5. Enhanced Mask Interaction Network (EMIN)

Existing approach. The raw HTC designed the mask interaction network (MIN) as shown in Figure 12a to establish a connection between different stages. Mask features of previous stage M_{i-1} are refined by a 1×1 conv for next stage M_i. A simple addition is used for feature fusion, i.e., $conv_{1\times1}(M_{i-1}(F)) + F$ where F is feature maps of backbone networks. We observe that a 1×1 conv offers limited feature refinement effects; a direct feature addition also offers limited fusion benefits.

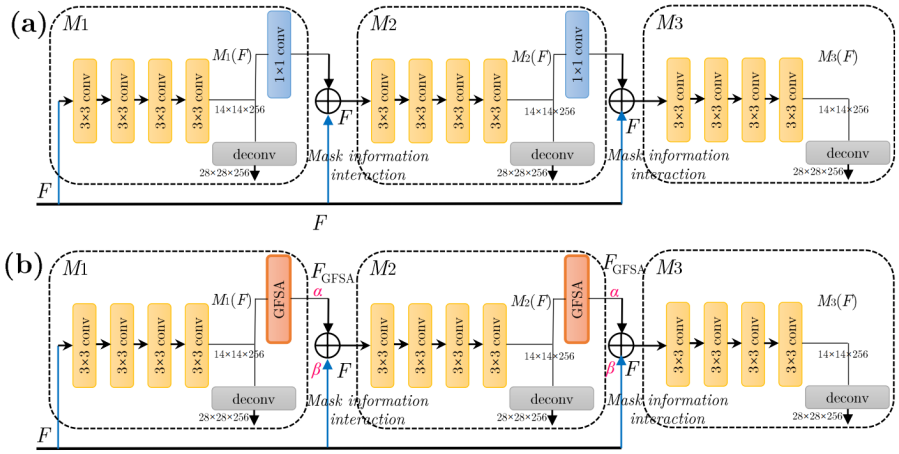

Figure 12. Mask interaction network (MIN) (**a**) Existing approach: MIN of HTC. (**b**) Proposed approach: EMIN of HTC+.

Proposed approach. Thus, we design an enhanced mask interaction network (EMIN) whose architecture is shown in Figure 12b. EMIN has two design concepts, i.e., (1) global feature self-attention (GFSA) in Section 3.5.1, and (2) adaptive mask feature fusion (AMFF) in Section 3.5.2.

3.5.1. Global Feature Self-Attention (GFSA)

We design a global feature self-attention block (GFSA) to replace the raw 1×1 conv, inspired by the advanced non-local neural networks [61]. GFSA can capture long-range dependencies of each mask pixel in the whole space, to enable a global receptive field, which is conducive to the efficient flow of information and context modeling. Figure 13a shows its implementation. In Figure 13a, features at the i-position are denoted by ϕ using a 1×1 conv W_ϕ. Features at the j-position are denoted by θ using a 1×1 conv W_θ. f is obtained from adaptive learning between ϕ and θ where the normalization process is equivalent to a *softmax* calculation function. The representation of the input at the j-position g is learned using another one 1×1 conv W_g. The response at the i-position y_i is obtained by a matrix multiplication. Note that we embed all features into $C/4$ channel space to reduce computational burdens. To apply response to the input readily, we use another one 1×1 conv W_z to transform dimension for the adding operation ($\oplus$). Finally, we achieve the global feature self-attention output F_{GFSA} that will be transmitted to the next stage.

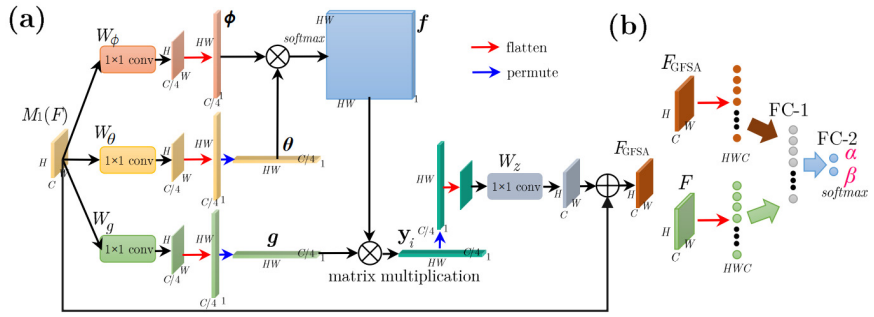

Figure 13. (**a**) Implementation of the global feature self-attention (GFSA). (**b**) Implementation of the adaptive mask feature fusion.

3.5.2. Adaptive Mask Feature Fusion (AMFF)

We arrange an adaptive mask feature fusion (AMFF) scheme for reasonably allocating contributions, instead of the direct feature adding. AMFF is depicted in Figure 13b. Firstly,

the previous mask feature F_{GFSA} and the backbone network feature F are flattened into 1D column vectors respectively, i.e., from $H \times W \times C$ to $HWC \times 1$.

Then, they are concatenated directly to be inputted into a fully-connected (FC-1) layer. Finally, the terminal 2-element FC-2 layer with a *softmax* activation is used to achieve two adaptive weight parameters α and β. Here, due to the use of the *softmax* activation, α plus β equals 1. The above is described by

$$W = softmax\{FC_2(FC_1([flatten(F_{\mathrm{GFSA}}), flatten(F)]))\} \tag{10}$$

where $W = [\alpha, \beta]^T$ denotes the weight vector. Finally, the mask interaction is implemented by

$$M_i(F) = \alpha \cdot F_{\mathrm{GFSA}} + \beta \cdot F \tag{11}$$

where $F_{\mathrm{GFSA}} = \mathrm{GFSA}(M_{i-1}(F))$. In summary, Equation (11) will be used to replace the original expression $M_i(F)$=$conv_{1\times1}(M_{i-1}(F)) + F$ in Figure 12a to form the final Figure 12b.

3.6. Post-Processing Technique (PPT)

Existing approach. The raw HTC offers NMS [27] and Soft-NMS [28] to remove duplicate detections, but the two both did not consider the prior knowledge of ships, that is, in most cases, only those ships with similar aspect ratios just dock together side by side, as shown in Figure 14. Not considering this prior knowledge will cause that the boxes which should be retained are removed when NMS is used; the boxes which should be deleted are retained when Soft-NMS is used. We think that when the ship aspect ratios are similar, one should better use Soft-NMS to avoid missed detections; however, when the ship aspect ratios have a huge difference, one should better use NMS to delete redundant boxes.

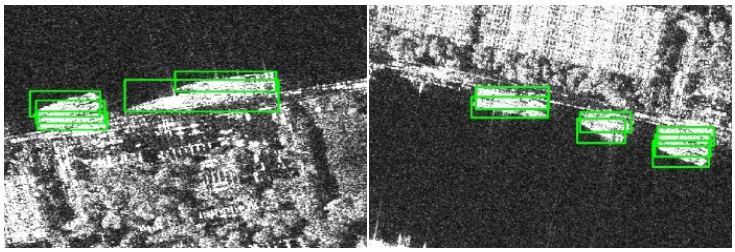

Figure 14. Densely moored ships. There are some hull overlaps between ships. In most cases, only ships with similar aspect ratios dock together. Green boxes denote the ground truths.

Proposed approach. Thus, we propose a post-process technique (PPT) guided by the prior of ship aspect ratios to adaptively select NMS and Soft-NMS. Algorithm 1 shows the implementation of PPT. We set a similarity threshold of ship aspect ratios ϵ_r to judge two cases, i.e., (i) a huge aspect ratio difference $|r_i - r_m| > \epsilon_r$ where r_m denotes the aspect ratio with the highest score s_m and r_i denotes the remaining boxes needed to traverse; (ii) a small aspect ratio difference $|r_i - r_m| \leq \epsilon_r$. Here, ϵ_r is set to 0.20 empirically, (see Section 6.5). For the former, NMS is executed to remove boxes safely (i.e., $\mathcal{B} \leftarrow \mathcal{B} - b_i$, $\mathcal{S} \leftarrow \mathcal{S} - s_i$). For the latter, Soft-NMS is executed to retain boxes (i.e., $\mathcal{B} \leftarrow \mathcal{B}$, $\mathcal{S} \leftarrow \mathcal{S}$) but the current box is given with a penalty score $s_i \leftarrow s_i \exp\left(\frac{IOU(\mathcal{M},b_i)^2}{\sigma}\right)$ for the densely moored ship detections in Figure 14.

Algorithm 1: PPT guided by the prior of ship aspect ratios.

Input: $\mathcal{B} = \{b_1, b_2, \cdots, b_N\}$, $\mathcal{R} = \{r_1, r_2, \cdots, r_N\}$, $\mathcal{S} = \{s_1, s_2, \cdots, s_N\}$, N_t, ϵ_r
$\mathcal{B}$ denotes the list of initial detection boxes. $\mathcal{R}$ denotes the list of initial detection aspect ratios. $\mathcal{S}$ denotes the list of initial detection scores. N_t denotes the IOU threshold. ϵ_r denotes the similarity threshold of aspect ratios.

Begin
$\mathcal{D} \leftarrow \{\}$
while $\mathcal{B} \neq \varnothing$ **do**
  $m \leftarrow$ argmax $\mathcal{S}$
  $\mathcal{M} \leftarrow b_m$
  $\mathcal{P} \leftarrow r_m$
  $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{M}$; $\mathcal{B} \leftarrow \mathcal{B} - \mathcal{M}$; $\mathcal{R} \leftarrow \mathcal{R} - \mathcal{P}$
  for (b_i, r_i) in ***zip*** $(\mathcal{B}, \mathcal{R})$ **do**
    if $IOU(\mathcal{M}, b_i) \geq N_t$ **then**
      Case 1: $|r_i - r_m| > \epsilon_r$
        $\mathcal{B} \leftarrow \mathcal{B} - b_i$; $\mathcal{R} \leftarrow \mathcal{R} - r_i$; $\mathcal{S} \leftarrow \mathcal{S} - s_i$ # NMS
      Case 2: $|r_i - r_m| \leq \epsilon_r$
        $\mathcal{B} \leftarrow \mathcal{B}$; $\mathcal{R} \leftarrow \mathcal{R}$; $\mathcal{S} \leftarrow \mathcal{S}$
        $s_i \leftarrow s_i e^{\frac{IOU(\mathcal{M}, b_i)^2}{\sigma}}$, $\forall b_i \notin \mathcal{D}$ # Soft-NMS
  Existing approach: consider one of Case 1 and Case 2. Proposed approach: consider both Case 1 and Case 2.
  end
  end
return $\mathcal{D}$, $\mathcal{S}$
end

Output: $\mathcal{D}$, $\mathcal{S}$

3.7. Hard Sample Mining Training Strategy (HSMTS)

Existing approach. The raw HTC did not offer useful training strategies to boost learning benefits as shown in Figure 15a, so we propose a hard sample mining training strategy (HSMTS) to supplement this blank for better accuracy. HSMTS is inspired by the extreme imbalance between positive and negative samples in SAR images, i.e., ships exhibit a sparse distribution in the whole SAR image, but backgrounds occupy more pixels. This causes the number of negative samples being much larger than the number of positive samples, so one should select more typical ones among a large number of negative samples to train the network so as to ensure the background discrimination ability of models. Otherwise, the network will fall into the over fitting of a large number of simple negative samples with low values. More typical negative samples should be those which are difficult to distinguish (close to positive). Only by emphasizing learning on them can the network improve its discrimination ability; repeated and mechanical learning on simple samples is worthless.

Proposed approach. HSMTS is in fact motivated by [66], but we add the extra supervision of the mask prediction loss. Figure 15b shows its implementation. From Figure 15b, we monitor the terminal training loss of EMIN where $Loss = Loss_{CLS} + Loss_{REG} + Loss_{MASK}$. Here, $Loss_{CLS}$ denotes the box classification loss. $Loss_{REG}$ denotes the box regression loss. $Loss_{MASK}$ denotes the mask prediction loss. The total training loss is first ranked. In training, the K negative samples with top K losses are collected to a hard negative sample pool. When the number of samples in the pool reaches a batch size, these hard negative samples are mapped into the feature maps of the backbone network via CROIE to extract feature subsets again for the next box and mask prediction. The above process repeated until the end of the training does not destroy the end-to-end training. The total number of the required samples is 512. The positive negative ratio is 1:3, in line with the raw report, so the number of positive samples is 128; as a result, K is equal to 384.

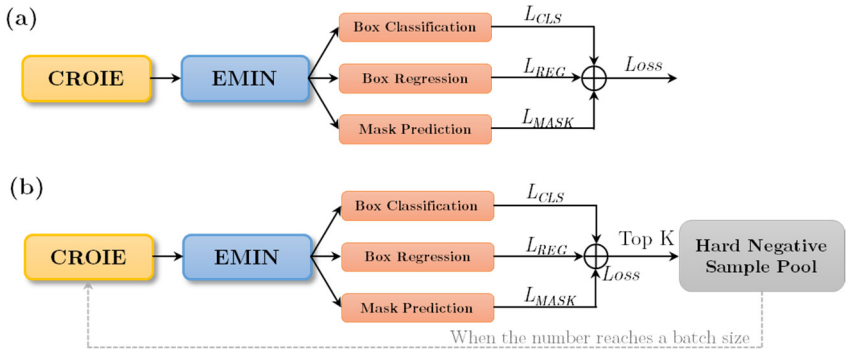

Figure 15. Different training strategies. (**a**) Existing approach: There are no hard negative sample mining mechanism of HTC. (**b**) Proposed approach: Hard sample mining training strategy (HSMTS) of HTC+.

4. Experiments

4.1. Dataset

SSDD [13] offers 1160 image samples with 512 × 512 size from RadarSat-2, TerraSAR-X and Sentinel-1. The training-test ratio is 4:1 [13]. There are 2551 ships in SSDD. The smallest ship is 28 pixel^2 and the largest one is 62,878 pixel^2 (pixel^2 denotes the product of width pixel and height one). SAR ships in SSDD are provided with various resolutions from 1m to 10m, and HH, HV, VV, and VH polarizations. In SSDD, images with suffix 1 and 9 (232 samples) are selected as the test set, and the others as the training set (928 samples).

HRSID [15] offers 5604 image samples with 800 × 800 average size from TerraSAR-X and Sentinel-1. The training set has 3643 samples and the rest serves as the test set, same as [15]. There are 16,965 ships in HRSID. SAR ships in HRSID are provided with resolutions from 0.1 m to 3 m, and HH, HV, and VV polarizations. The smallest one is 3 pixel^2, and the largest one is 522,400 pixel^2. HRSID is divided into a training and test set by the ratio of 13:7, same as [15].

4.2. Training Details

The backbone sub-network MRFEN of HTC+ and other backbone networks for performance comparison are pretrained on ImageNet [67]. Following Faster R-CNN [31], we first train the backbone network, the EFPN sub-network, and the region generation sub-network SGAAL jointly. Then, we fix them to train the CROIE sub-network and the EMIN sub-network. Moreover, the same as for HTC, we end-to-end to train and test HTC+. We train all networks (HTC+ and the other 9 models) by 12 epochs on the SAR ship datasets using SGD [68]. The learning rate is set to 0.008 that will be reduced 10 times at 8- and 11-epoch. The batch size is set to 4. Other training details are consistent with HTC. Experiments were run on a personal computer with RTX 3090 GPU, i9-9900 CPU, and 32G memory based on mmdet [69] and Pytorch. In the test, the trained weights are loaded to evaluate performance.

4.3. Evaluation Criteria

Similar to COCO [53], AP is the mean of different IOU thresholds with 0.50:0.05:0.95. AP_{50} is the accuracy with a 0.50 IOU threshold. AP_{75} is that with a 0.75 IOU threshold. AP_S is that of small ships (<32^2 pixels). AP_M is that of medium ships (>32^2 pixels and <96^2 pixels). AP_L is that of large ships (>96^2 pixels). In this paper, AP serves as the core index for accuracy comparison.

5. Results

To reveal the state-of-the-art performance of our HTC+, nine competitive models are reproduced for comparison of accuracy. The comparative experiments are conducted on SSDD and HRSID datasets.

5.1. Quantitative Results

Tables 2 and 3 show the quantitative comparison results on SSDD and HRSID. HTC is the experimental baseline reproduced basically in line with the raw report. Its box and mask AP are comparable to reports [17,18].

From Tables 2 and 3, the following conclusions can be drawn:

1. Compared with HTC, HTC+ offers 6.7% box AP and 5.0% mask AP increments on SSDD; they are 4.9% and 3.9% on HRSID. Seven novelties enable such excellent accuracy. Accuracy increments of different novelties are different, but the resulting accuracy always presents an upward trend, showing the effectiveness of each novelty. AP_L has some fluctuations because the proportion of large ships in datasets is relatively small as in Figure 3a. Sensitivity analysis of different novelties on the total performance will be introduced in Section 6. Certainly, the speed indeed becomes lower and lower as expected. The trade-off between accuracy and speed is an eternal topic which will be further considered in the future.
2. Compared with the second-class model, HTC+ offers 6.7% box AP and 4.7% mask AP increments on SSDD; they are 4.8% and 3.7% on HRSID. This shows the state-of-the-art performance of HTC+. The increment of the box AP is lower than that of the mask AP, which is in line with common sense, because the pixel-level mask segmentation is more challenging.
3. Compared with the other methods, HTC+ offers the modest operation speed (i.e., 3.36 FPS on SSDD and 2.18 FPS on HRSID) due to its heavier network. This disadvantage needs to be resolved in the future. Moreover, although YOLACT [36] indeed offers the optimal speed, its accuracy is too poor to meet applications.

Table 2. Quantitative results on SSDD. The suboptimal method is marked by underline "—". FPS: frames per second.

MRFEN	EFPN	SGAALN	CROIE	EMIN	PPT	HEMTS	Box (%)						Mask (%)						FPS
							AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	
–	–	–	–	–	–	–	65.6	93.6	76.3	65.2	68.4	27.5	59.3	91.7	73.1	58.7	61.6	34.8	11.60
√							67.0	90.9	81.3	67.8	65.8	34.6	61.6	90.9	78.8	61.5	61.8	37.6	8.26
√	√						67.9	93.5	81.5	68.0	68.1	43.4	62.0	92.7	77.1	62.0	62.3	50.2	7.25
√	√	√					68.3	94.3	81.5	68.4	68.1	43.3	62.4	93.5	77.1	62.4	62.3	50.2	6.27
√	√	√	√				69.0	95.4	82.8	69.4	68.7	34.6	62.8	93.6	79.6	62.9	62.5	37.6	3.93
√	√	√	√	√			69.8	96.3	83.8	69.6	71.4	34.2	63.2	93.5	80.0	63.3	63.6	32.6	3.63
√	√	√	√	√	√		71.6	96.8	86.8	71.5	72.7	42.3	63.7	94.5	80.9	63.6	64.4	51.2	3.36
√	√	√	√	√	√	√	72.3	96.8	87.2	72.0	74.0	51.0	64.3	94.7	82.3	64.1	64.9	65.0	3.36
							+6.7	+3.2	+10.9	+6.8	+5.6	+23.5	+5.0	+3.0	+9.2	+5.4	+3.3	+30.2	

Method	Backbone	Box (%)						Mask (%)						FPS
		AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	
Mask R-CNN [30]	ResNet-101	62.0	91.5	75.4	62.0	64.4	19.7	57.8	88.5	72.1	57.2	60.8	27.4	11.05
Mask Scoring R-CNN [32]	ResNet-101	62.4	91.0	75.1	61.9	66.0	15.7	58.6	89.4	73.2	58.0	61.4	22.6	12.88
Cascade Mask R-CNN [33]	ResNet-101	63.0	89.6	75.2	62.4	66.0	12.0	56.6	87.5	70.5	56.3	58.8	22.6	10.55
PANet [34]	ResNet-101	63.3	93.4	75.4	63.4	65.5	40.8	59.6	91.1	74.0	59.3	61.0	52.1	13.65
YOLACT [36]	ResNet-101	54.0	90.6	61.2	56.9	48.2	12.6	48.4	88.0	52.1	47.3	53.5	40.2	15.47
GRoIE [35]	ResNet-101	61.2	91.5	71.6	62.2	59.8	8.7	58.3	89.8	72.7	58.6	58.7	21.8	9.67
HQ-ISNet [16]	HRNetV2-W18	64.9	91.0	76.3	64.7	66.6	26.0	58.6	89.3	73.6	58.2	60.4	37.2	8.59
HQ-ISNet [16]	HRNetV2-W32	65.5	90.7	77.3	65.6	66.9	23.2	59.3	90.4	75.5	58.9	61.1	37.3	8.00
HQ-ISNet [16]	HRNetV2-W40	63.6	87.8	75.3	62.6	67.8	27.9	57.6	86.0	72.6	56.7	61.3	50.2	7.73
SA R-CNN [17]	ResNet-50-GCB	63.2	92.1	75.2	63.8	64.0	7.0	59.4	90.4	73.3	59.6	60.3	20.2	13.65
FL-CSE-ROIE [21]	ResNet-101	68.0	95.9	81.1	67.6	70.1	56.2	62.6	93.7	78.3	63.3	61.2	75.0	8.92
GCBANet [22]	ResNet-101	68.4	95.4	82.2	68.9	68.0	45.6	63.1	93.5	78.8	63.2	63.0	55.1	6.11
HTC [24]	ResNet-101	65.6	93.6	76.3	65.2	68.4	27.5	59.3	91.7	73.1	58.7	61.6	34.8	11.60
HTC+ (Ours)	MRFEN	72.3	96.8	87.2	72.0	74.0	51.0	64.3	94.7	82.3	64.1	64.9	65.0	3.36
		+6.7	+3.2	+9.9	+6.4	+7.4	+10.2	+4.7	+3.0	+6.8	+4.5	+3.3	+12.9	

Table 3. Quantitative results on HRSID. FPS: frames per second.

MRFEN	EFPN	SGAALN	CROIE	EMIN	PPT	HEMTS	Box (%)						Mask (%)						FPS
							AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	
–	–	–	–	–	–	–	66.6	86.0	77.1	67.6	69.0	28.1	55.2	84.9	66.5	54.7	63.8	19.2	7.42
√							67.3	85.8	77.8	68.2	68.6	21.7	55.6	84.7	67.8	55.3	63.5	22.7	5.30
√	√						68.1	87.9	78.4	69.0	69.5	24.8	56.3	86.1	68.4	55.8	64.0	24.2	4.75
√	√	√					68.5	88.2	78.5	69.4	69.5	29.0	56.7	86.3	68.7	56.3	63.8	25.7	4.01
√	√	√	√				68.7	88.3	78.7	69.6	70.0	31.3	56.9	86.4	68.7	56.4	64.6	27.5	2.55
√	√	√	√	√			69.2	88.9	79.3	70.1	70.6	31.7	57.2	87.1	69.3	56.8	64.9	27.7	2.30
√	√	√	√	√	√		70.5	91.3	81.3	71.5	70.6	36.7	58.1	89.2	69.4	57.8	64.6	26.5	2.18
√	√	√	√	√	√	√	71.5	92.3	82.5	72.6	71.4	38.2	59.1	90.3	71.0	58.7	65.7	26.8	2.18
							+4.9	+6.3	+5.4	+5.0	+2.4	+10.1	+3.9	+5.4	+4.5	+4.0	+1.9	+7.6	

Method	Backbone	Box (%)						Mask (%)						FPS
		AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	
Mask R-CNN [30]	ResNet-101	65.1	87.7	75.5	66.1	68.4	14.1	54.8	85.7	65.2	54.3	62.5	13.3	7.07
Mask Scoring R-CNN [32]	ResNet-101	65.2	87.6	75.4	66.5	67.4	13.4	54.9	85.1	65.9	54.5	61.5	12.9	8.24
Cascade Mask R-CNN [33]	ResNet-101	65.1	85.4	74.4	66.0	69.0	17.1	52.8	83.4	62.9	52.2	62.2	17.0	6.75
PANet [34]	ResNet-101	65.4	88.0	75.7	66.5	68.2	22.1	55.1	86.0	66.2	54.7	62.8	17.8	8.74
YOLACT [36]	ResNet-101	47.9	74.4	53.3	51.7	34.9	3.3	39.6	71.1	41.9	39.5	46.1	7.3	10.02
GRoIE [35]	ResNet-101	65.4	87.8	75.5	66.5	67.2	21.8	55.4	85.8	66.9	54.9	63.5	19.7	6.19
HQ-ISNet [16]	HRNetV2-W18	66.0	86.1	75.6	67.1	66.3	8.9	53.4	84.2	64.3	53.2	59.7	10.7	5.50
HQ-ISNet [16]	HRNetV2-W32	66.7	86.9	76.3	67.8	68.3	16.8	54.6	85.0	65.8	54.2	61.7	13.4	5.12
HQ-ISNet [16]	HRNetV2-W40	66.7	86.2	76.3	67.9	68.6	11.7	54.2	84.3	64.9	53.9	61.9	12.8	4.95
SA R-CNN [17]	ResNet-50-GCB	65.2	88.3	75.2	66.4	65.4	10.2	55.2	86.2	66.7	54.9	60.9	12.3	8.74
FL-CSE-ROIE [21]	ResNet-101	69.0	90.2	79.5	69.9	71.1	32.3	57.9	88.6	69.5	57.3	65.7	26.1	5.24
GCBANet [22]	ResNet-101	69.4	89.8	79.2	70.4	71.3	32.2	57.3	88.6	68.9	57.0	64.3	25.9	4.06
HTC [24]	ResNet-101	66.6	86.0	77.1	67.6	69.0	28.1	55.2	84.9	66.5	54.7	63.8	19.2	7.42
HTC+ (Ours)	MRFEN	71.5	92.3	82.5	72.6	71.4	38.2	59.1	90.3	71.0	58.7	65.7	26.8	2.18
		+4.8	+4.3	+5.4	+4.7	+2.4	+10.1	+3.7	+4.1	+4.1	+3.8	+1.9	+7.6	

Table 4 shows the computational complexity calculations of different methods. Here, we adopt the floating point of operations (FLOPs) to measure calculations whose unit is the giga multiply add calculations (GMACs) [70]. From Table 4, the calculation amount of GCBANet is more than other models, so the future model computational complexity optimization is needed.

Table 4. Computational complexity calculations of different methods. Here, we adopt the floating point of operations (FLOPs) to measure calculations whose unit is the giga multiply add calculations (GMACs) [70].

Method	Backbone	FLOPs (GMACs)
Mask R-CNN [30]	ResNet-101	121.32
Mask Scoring R-CNN [32]	ResNet-101	121.32
Cascade Mask R-CNN [33]	ResNet-101	226.31
PANet [34]	ResNet-101	127.66
YOLACT [36]	ResNet-101	67.14
GRoIE [35]	ResNet-101	581.28
HQ-ISNet [16]	HRNetV2-W18	201.84
HQ-ISNet [16]	HRNetV2-W32	226.90
HQ-ISNet [16]	HRNetV2-W40	247.49
SA R-CNN [17]	ResNet-50-GCB	101.87
GCBANet [22]	ResNet-101	947.96
HTC [24]	ResNet-101	228.90
HTC+ (Ours)	ResNet-101	1289.45

5.2. Qualitative Results

Figures 16 and 17 show the qualitative results on SSDD and HRSID. Due to limited pages, we only show the comparison results with HTC. From Figures 16 and 17, HTC+ can detect more real ships than HTC (e.g., the #5 sample in Figure 16). It can suppress false alarms (e.g., the #2 sample in Figure 16). It also offers better positioning performance (e.g., the #1 sample in Figure 16) and higher confidence scores (e.g., the #6 sample in Figure 16). These reveal the state-of-the-art performance of HTC+.

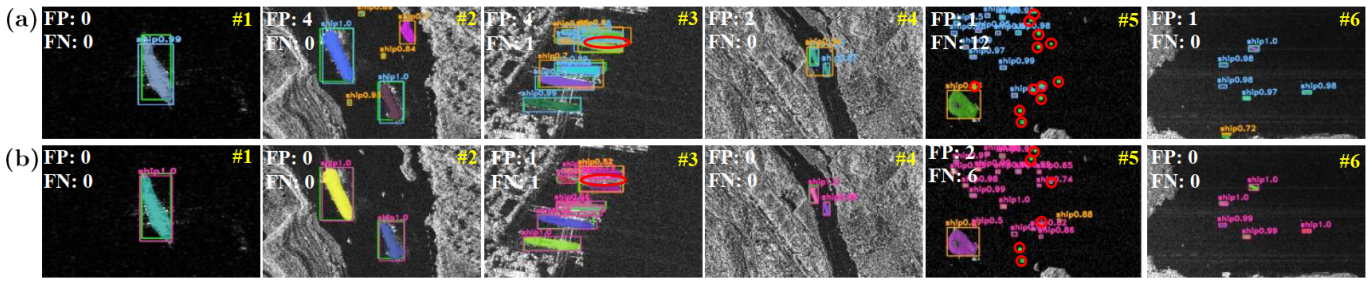

Figure 16. Qualitative results on SSDD. (**a**) The vanilla HTC. (**b**) Our HTC+. Green boxes denote the ground truths. Orange boxes denote the false alarms (i.e., false positives, FP). Red circles denote the missed detections (i.e., false negatives, FN).

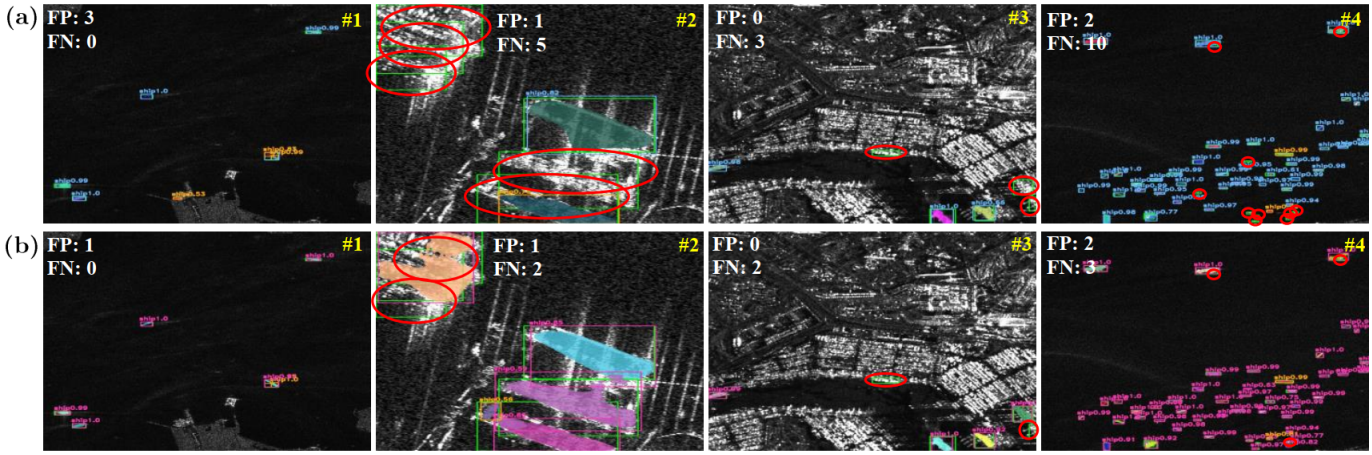

Figure 17. Qualitative results on HRSID. (**a**) The vanilla HTC. (**b**) Our HTC+. Green boxes denote the ground truths. Orange boxes denote the false alarms (i.e., false positives, FP). Red circles denote the missed detections (i.e., false negatives, FN).

6. Ablation Study

In this section, we will conduct ablation studies for sensitivity analysis of each technique. All the following experiments are conducted on SSDD. These ablation studies can also confirm the effectiveness of each technique. For rigorous comparison, we install and remove the particular technique while freezing the other six.

6.1. Ablation Study on MRFEN

6.1.1. Effectiveness of MRFEN

Table 5 shows the quantitative results with and without MRFEN. MRFEN offers a better accuracy than the common plain structure, because it expands the range of network spatial search to achieve multi-resolution analysis, enabling better multi-scale ship feature representation.

Table 5. Quantitative results on with and without MRFEN.

MRFEN	Box AP (%)	Mask AP (%)	FPS
× [1]	70.9	63.7	6.52
√ [2]	72.3	64.3	3.36

[1] ResNet-101 in Figure 2a. [2] Our MRFEN in Figure 2b.

6.1.2. Component Analysis in MRFEN

Table 6 shows the results of component analysis in MRFEN. All components are conducive to improving accuracy, showing their effectiveness. Three components enhance the multi-scale feature description of the network, enabling better performance. MRFE reduces the speed greatly because it makes the network heavier. Although MRFE offers an only 0.2% box AP gain, its mask AP gain is more significant than others, so it is still effective. One can also find that after the feature fusion is established, the box AP is further improved by 0.4%. Therefore, it is essential to exchange information in multi-resolution parallel branches. Moreover, CBAM can improve accuracy further because it can focus on more important features.

Table 6. Quantitative results of component analysis in MRFEN.

MRFE [1]	FF in MSAFF [2]	CBAM in MSAFF [3]	ASPP [4]	Box AP (%)	Mask AP (%)	FPS
–	–		–	70.9	63.7	6.52
√				71.1	64.0	3.84
√	√			71.5	64.1	3.72
√	√	√		71.7	64.2	3.53
√	√	√	√	72.3	64.3	3.36

[1] MRFE denotes the multi-resolution feature extraction. Feature fusion and CBAM are deleted. [2] FF denotes the feature fusion in the multi-scale attention-based feature fusion (MSAFF). [3] CBAM denotes the convolutional block attention module in the MSAFF. [4] ASPP denotes the atrous spatial pyramid pooling.

6.1.3. Compared with Other Backbones

We compare performance of other backbones in Table 7. MRFEN offers the optimum accuracy. HRNetV2-W40 [16] offers a comparable 72.2% box AP, but its mask AP is lower than MRFEN (63.5% < 64.3%). Furthermore, in our MRFEN, MRFE+MSAFF can be regarded as an improved version of HRNet [41], because CBAM is added during feature fusion. We also study the advantage of HRNet compared with other backbones under the condition that ASPP is used. The results are shown in Table 8. From Table 8, HRNet achieves the best box AP and mask AP, thus the multi-resolution parallel structure is better than the plain structure.

Table 7. Quantitative results of different backbones. ASPP is not used in other backbones.

Backbone	Box AP (%)	Mask AP (%)	FPS
ResNet-101 [39]	70.9	63.7	6.52
ResNeXt-101-32x4d [40]	71.0	63.5	5.15
ResNeXt-101-32x8d [40]	71.3	63.8	4.93
ResNeXt-101-64x4d [40]	71.2	64.1	3.85
HRNetV2-W18 [16]	71.0	63.2	3.85
HRNetV2-W32 [16]	70.6	64.0	3.76
HRNetV2-W40 [16]	72.2	63.5	3.40
Res2Net-101 [38]	71.9	64.0	4.62
MRFEN (Ours)	72.3	64.3	3.36

Table 8. Quantitative results on with and without MRFEN. ASPP is used in other backbones.

	Backbone [1]	Box AP (%)	Mask AP (%)	FPS
MRFEN	ResNet-101 [39]	71.1	63.7	6.50
	ResNeXt-101-32x4d [40]	71.3	63.6	5.10
	ResNeXt-101-32x8d [40]	71.6	64.0	4.86
	ResNeXt-101-64x4d [40]	71.9	64.1	3.81
	Res2Net-101 [38]	72.0	64.1	4.58
	HRNet (Ours) [2]	72.3	64.3	3.36

[1] The backbone (MRFE+MSAFF) inside MRFEN is replaced. [2] In MRFEN, MRFE+MSAFF can be regarded as an improved version of HRNet because CBAM is added during feature fusion.

6.2. Ablation Study on EFPN

6.2.1. Effectiveness of EFPN

Table 9 shows the results with or without EFPN. EFPN offers a ~3% box AP gain; a ~2% mask AP gain. The speed is sacrificed, but it offers better multi-scale performance, considering large and small ships simultaneously.

Table 9. Quantitative results on with and without EFPN.

EFPN	Box AP (%)	Mask AP (%)	FPS
× [1]	69.4	62.2	6.89
√ [2]	72.3	64.3	3.36

[1] FPN in Figure 5a. [2] Our EFPN in Figure 5b.

6.2.2. Component Analysis in EFPN

Table 10 shows the results of component analysis in EFPN. Each component is conducive to improving accuracy more or less, showing their effectiveness. CARAFE reduces the speed more obviously, because it adds FPN levels with increased calculation costs.

Table 10. Quantitative Results of Component Analysis in EFPN.

CARAFE [1]	FB [2]	FR [3]	FE [4]	Box AP (%)	Mask AP (%)	FPS
–	–	–	–	69.4	62.2	6.89
√				70.1	63.4	3.96
√	√			70.7	64.0	3.53
√	√	√		71.5	64.2	3.40
√	√	√	√	72.3	64.3	3.36

[1] CARAFE denotes the content-aware reassembly of features. [2] FB denotes the feature balance. [3] FR denotes the feature refinement. [4] FE denotes the feature enhancement.

6.2.3. Compared with Other FPNs

We compare performance of other FPNs in Table 11. MFPN offers the best accuracy except for the mask AP of Quad-FPN [58]. Still, MFPN is better than Quad-FPN, because its box AP is higher (72.3% > 71.8%).

Table 11. Quantitative results of different FPNs.

Type	Box AP (%)	Mask AP (%)	FPS
FPN [51]	69.4	62.2	6.89
B-FPN [59]	69.8	62.9	5.45
CARAFE-FPN [55]	71.3	63.5	5.62
Quad-FPN [58]	71.8	64.3	3.48
MFPN (Ours)	72.3	64.3	3.36

6.3. Ablation Study on SGAALN

6.3.1. Effectiveness of SGAALN

Table 12 shows the results with and without SGAALN. SGAALN boosts the box AP and the mask AP by ~1%. It can generate more optimized location- and shape-adaptive anchors to better match SAR ships. This can ease background interferences for better performance.

Table 12. Quantitative results on with and without SGAALN.

SGAALN	Box AP (%)	Mask AP (%)	FPS
× [1]	71.2	63.5	3.63
√ [2]	72.3	64.3	3.36

[1] Fixed anchors and aspect ratios used in the vanilla HTC. [2] Our SGAALN in Figure 8.

6.3.2. Component Analysis in SGAALN

Table 13 shows the results of component analysis in SGAALN. ALP boosts accuracy in any case, but ASP must be equipped with FA to give full play to its advantages, because FA aligns the raw feature maps to width-height of anchors to eliminate feature differences.

Table 13. Quantitative results of component analysis in SGAALN.

ALP [1]	ASP [2]	FA [3]	Box AP (%)	Mask AP (%)	FPS
–	–	–	71.2	63.5	3.63
√			72.0	64.0	3.58
√	√		71.9	64.0	3.54
√	√	√	72.3	64.3	3.36

[1] ALP denotes the anchor location prediction. [2] ASP denotes the anchor shape prediction. [3] FA denotes the feature adaption.

6.3.3. Different Probability Thresholds

We adjust probability thresholds to determine their optimal value in Table 14. One finds that when $\epsilon_t = 0.10$, the accuracy reaches the peak, so it is selected, as also suggested by [29], because it can remove many false positives meanwhile maintaining an unaffected recall rate.

Table 14. Quantitative results of component analysis in SGAALN.

ϵ_t	Box AP (%)	Mask AP (%)	FPS
0.00	71.5	63.9	3.35
0.05	71.8	64.0	3.36
0.10	72.3	64.3	3.36
0.15	72.0	64.1	3.34
0.20	70.8	63.7	3.26

6.4. Ablation Study on CROIE

6.4.1. Effectiveness of CROIE

Table 15 shows the results with/without CROIE. CROIE improves the accuracy by ~0.5%, because it offers more context information to the network, conducive to enhancing the background discrimination ability.

Table 15. Quantitative results on with and without CROIE.

CROIE	Box AP (%)	Mask AP (%)	FPS
× [1]	71.2	63.5	3.63
√ [2]	72.3	64.3	3.36

[1] ROI is used as shown in Figure 9a. [2] ROI, ROI-C1 and ROI-C2 are used as shown in Figure 9b.

6.4.2. Different Range Contexts

We survey the influences of different range contexts on performance as shown in Table 16. We observe that moderate contexts are beneficial, but excessive ones will lead to negative effects. When using CROIE, a special parameter adjustment is required to be in line with actual applications. For the best accuracy, we set the two amplification factors c_1 and c_2 to 1.5 and 2.0 respectively.

Table 16. Quantitative results of different range contexts.

c_1	c_2	Box AP (%)	Mask AP (%)	FPS
1.5	2.0	72.3	64.3	3.36
1.5	2.5	72.0	64.1	3.20
2.0	2.5	71.5	63.7	3.19
2.5	3.0	70.7	62.5	2.96

6.5. Ablation Study on EMIN

6.5.1. Effectiveness of EMIN

Table 17 shows the results with and without EMIN. EMIN offers better accuracy than the raw MIN. It transmits more important mask features to the next stage. It further balances the contributions between the backbone network's features and the previous stage's features. Consequently, the efficiency of information flow is improved, bringing better learning benefits.

Table 17. Quantitative results with and without EMIN.

EMIN	Box AP (%)	Mask AP (%)	FPS
× [1]	71.7	63.9	3.87
√ [2]	72.3	64.3	3.36

[1] The raw MIN in Figure 12a. [2] Our EMIN in Figure 12b.

6.5.2. Component Analysis in EMIN

Table 18 shows the results of component analysis in EMIN. Each component offers an observable accuracy gain, showing their effectiveness. They do not impose great impacts on speed, so they are cost-effective.

Table 18. Quantitative results of component analysis in EMIN.

GFSA [1]	AMFF [2]	Box AP (%)	Mask AP (%)	FPS
–	–	71.7	63.9	3.87
√		72.1	64.1	3.58
√	√	72.3	64.3	3.36

[1] GFSA denotes the global feature self-attention. [2] AMFF denotes the adaptive mask feature fusion.

6.6. Ablation Study on PPT

6.6.1. Effectiveness of PPT

Table 19 shows the results with or without PPT. PPT has a slightly better accuracy than NMS and Soft-NMS with little sacrifice of speed. It considers the ship aspect ratio prior to determine whether to suppress boxes, with the advantages of NMS and Soft-NMS, enabling better performance.

Table 19. Quantitative results on with and without PPT.

PPT	NMS	Soft-NMS	Box AP (%)	Mask AP (%)	FPS
	√		71.5	63.7	3.40
		√	72.0	64.1	3.38
√	√	√	72.3	64.3	3.36

6.6.2. Different Similarity Thresholds of Aspect Ratios

We survey the influences of different similarity thresholds of aspect ratios on the performance as in Table 20. Due to SAR imaging error and annotation deviation, it is impossible to be sure that ships moored in parallel have absolutely-equal aspect ratios. Therefore, setting this threshold reasonably is needed. In our work, we set ϵ_r to 0.20 because it offers the best accuracy.

Table 20. Quantitative results of different similarity thresholds of aspect ratios.

ϵ_r	Box AP (%)	Mask AP (%)	FPS
0.10	71.8	63.9	3.34
0.15	72.1	63.9	3.36
0.20	72.3	64.3	3.36
0.25	72.2	64.1	3.36
0.30	72.0	64.0	3.35

6.7. Ablation Study on HSMTS

6.7.1. Effectiveness of HSMTS

Table 21 shows the results with and without HEMTS. HEMTS further improves the accuracy; the network boosts learning benefits by focusing on difficult samples to boost foreground-background recognition ability. HEMTS is only used in training, so the speed is not affected.

Table 21. Quantitative results on with and without HSMTS.

HSMTS	Mask Loss	Box AP (%)	Mask AP (%)	FPS
× [1]		71.6	63.7	3.36
√ [2]		72.0	63.9	3.36
√ [3]	√ [3]	72.3	64.3	3.36

[1] The raw random sampling. [2] OMEM. Here, the mask prediction loss is not monitored. [3] Our HSMTS in Figure 15. Here, the mask prediction loss is monitored.

6.7.2. Compared with OHEM

In Table 21, HEMTS (the second row) performs better than OHEM (the third row) because HEMTS adds the extra supervision of the mask prediction loss. This is conducive to mining more representative difficult negative samples.

7. Discussions

7.1. Multi-Scale Training and Test

We also discuss the multi-scale training and test on SSDD in Table 22. The single-scale input is [512 × 512]; the multi-scale input is [416 × 416, 512 × 512, 608 × 608] inspired by YOLOv3 [25]. Multi-scale training and test can further improve the accuracy but the speed becomes lower for all models. Our single-scale HTC+ surpasses all other multi-scale models. This advantage comes from the multi-resolution feature extraction. Our multi-scale HTC+ enables the better performance from 72.3% to 72.9% box AP and from 64.3% to 65.1% mask AP. It is always far superior to all other competition models, which shows its better performance.

7.2. Extension to Mask R-CNN

To confirm the universal effectiveness of the proposed techniques on other instance segmentation models, we extend them to the mainstream Mask R-CNN [30]. Here, EMIN is not applicable, because Mask R-CNN does not have mask information interaction branches, whose mask head is not cascaded. The results are shown in Table 23. From Table 23, six novelties all offer continuous accuracy growth, from the initial 62.0% to the final 70.8% box AP, i.e., a huge 8.8% incremental improvement, and from the initial 57.8% to the final 62.5% mask AP, i.e., a huge 4.7% incremental improvement. The above reveals the universal validity of our proposed techniques.

Table 22. Quantitative results of multi-scale training and test on SSDD. The suboptimal method is marked by underline "—".

Method	Backbone	Box AP (%)	Mask AP (%)	FPS
Mask R-CNN-Multi	ResNet-101	64.1	60.6	7.48
Mask Scoring R-CNN-Multi	ResNet-101	65.8	60.4	7.25
Cascade Mask R-CNN-Multi	ResNet-101	65.4	60.0	5.80
HTC-Multi	ResNet-101	66.8	60.7	5.52
PANet-Multi	ResNet-101	65.4	60.4	7.48
YOLACT-Multi	ResNet-101	55.2	51.4	10.78
GRoIE-Multi	ResNet-101	63.5	60.4	4.64
HQ-ISNet-Multi	HRNetV2-W18	65.6	59.4	4.07
HQ-ISNet-Multi	HRNetV2-W32	66.0	59.5	3.87
HQ-ISNet-Multi	HRNetV2-W40	63.8	59.5	3.57
SA R-CNN-Multi	ResNet-50-GCB	64.1	60.3	8.00
HTC+-Single [1]	MRFEN	72.3	64.3	3.36
HTC+-Multi [2]	MRFEN	72.9	65.1	2.02

[1] Single denotes the input size [512 × 512]. [2] Multi denotes the input size [416 × 416, 512 × 512, 608 × 608] inspired by YOLOv3 [25].

Table 23. Quantitative results of extension to Mask R-CNN on SSDD.

	MRFEN	EFPN	SGAALN	CROIE	PPT	HSMTS	Box AP (%)	Mask AP (%)	FPS
Mask R-CNN [30] *	–	–	–	–	–	–	62.0	57.8	11.05
	√						63.6	59.0	9.34
	√	√					65.2	60.6	8.02
	√	√	√				65.9	61.0	7.85
	√	√	√	√			68.5	61.2	3.96
	√	√	√	√	√		69.6	62.0	3.55
	√	√	√	√	√	√	70.8	62.5	3.55

* Mask R-CNN does not have mask information interaction branches, because its mask head is not cascaded. Thus, EMIN is not applicable to Mask R-CNN.

7.3. Extension to Faster R-CNN

We also extend the proposed techniques (except EMIN only used in segmentation models) to the pure detection model. We take the mainstream two-stage model Faster R-CNN [31] as an example. The results are shown in Table 24. From Table 24, six novelties all offer continuous accuracy growth, from the initial 62.1% to the final 69.1% box AP, i.e., a huge 7% incremental improvement, which shows their universal validity. Certainly, these benefits are achieved at a certain sacrifice of speed, which will be considered in the future.

Table 24. Quantitative results of extension to Faster R-CNN on SSDD.

	MRFEN	EFPN	SGAALN	CROIE	PPT	HSMTS	Box AP (%)	FPS
Faster R-CNN [31] *	–	–	–	–	–	–	62.1	13.65
	√						64.5	10.56
	√	√					66.8	8.74
	√	√	√				67.2	8.38
	√	√	√	√			68.0	7.69
	√	√	√	√	√		68.5	6.72
	√	√	√	√	√	√	69.1	6.72

* EMIN is only used in segmentation models, but Faster R-CNN is a detection model. Therefore, EMIN cannot be applied to Faster R-CNN.

8. Conclusions

We propose HTC+ to boost SAR ship instance segmentation. Seven techniques (MR-FEN, EFPN, SGAALN, CROIE, EMIN, PPT, and HSMTS) are used ensure the state-of-the-art accuracy of HTC+. HTC+ is elaborately designed for SAR ship tasks in consideration of the targeted SAR characteristics. HTC+ is superior to the vanilla HTC by 6.7% box AP and 5.0% mask AP and by 4.9% and 3.9% on HRSID. It outperforms the other nine advanced models. Moreover, we also extend the proposed techniques to Faster R-CNN to confirm their effectiveness for pure detection tasks; results reveal that they can offer continuous accuracy growth.

In the future, the speed optimization [71,72] will be considered; other tricks [73] will also be considered for better accuracy.

Author Contributions: Conceptualization, T.Z.; methodology, T.Z.; software, T.Z.; validation, T.Z.; formal analysis, T.Z.; investigation, T.Z.; resources, T.Z.; data curation, T.Z.; writing—original draft preparation, T.Z.; writing—review and editing, X.Z.; visualization, T.Z.; supervision, T.Z.; project administration, X.Z.; funding acquisition, X.Z. 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 (61571099).

Data Availability Statement: Not applicable. No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

1. 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.* **2020**, *30*, 781–794. [CrossRef]
2. Shan, Y.; Zhou, X.; Liu, S.; Zhang, Y.; Huang, K. Siamfpn: A Deep Learning Method for Accurate and Real-Time Maritime Ship Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 315–325. [CrossRef]
3. Ribeiro, R.; Cruz, G.; Matos, J.; Bernardino, A. A Data Set for Airborne Maritime Surveillance Environments. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *29*, 2720–2732. [CrossRef]
4. Shao, Z.; Wu, W.; Wang, Z.; Du, W.; Li, C. SeaShips: A Large-Scale Precisely Annotated Dataset for Ship Detection. *IEEE Trans. Multimedia* **2018**, *20*, 2593–2604. [CrossRef]
5. Zhang, T.; Zhang, X. A Polarization Fusion Network with Geometric Feature Embedding for SAR Ship Classification. *Pattern Recognit.* **2021**, *123*, 108365. [CrossRef]
6. Zhang, T.; Zhang, X.; Ke, X.; Liu, C.; Xu, X.; Zhan, X.; Wang, C.; Ahmad, I.; Zhou, Y.; Pan, D.; et al. HOG-ShipCLSNet: A Novel Deep Learning Network with HOG Feature Fusion for SAR Ship Classification. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *60*, 5210322. [CrossRef]
7. Zhang, T.; Zhang, X. Squeeze-and-Excitation Laplacian Pyramid Network with Dual-Polarization Feature Fusion for Ship Classification in SAR Images. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 4019905. [CrossRef]
8. Oh, J.; Youm, G.Y.; Kim, M. Spam-Net: A CNN-Based SAR Target Recognition Network with Pose Angle Marginalization Learning. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 701–714. [CrossRef]
9. Ma, F.; Gao, F.; Wang, J.; Hussain, A.; Zhou, H. A Novel Biologically-Inspired Target Detection Method Based on Saliency Analysis for Synthetic Aperture Radar (SAR) Imagery. *Neurocomputing.* **2020**, *402*, 66–79. [CrossRef]
10. Tao, D.; Anfinsen, S.N.; Brekke, C. Robust CFAR Detector Based on Truncated Statistics in Multiple-Target Situations. *IEEE Trans. Geosci. Remote. Sens.* **2015**, *54*, 117–134. [CrossRef]
11. Zhang, T.; Zhang, X.; Liu, C.; Shi, J.; Wei, S.; Ahmad, I.; Zhan, X.; Zhou, Y.; Pan, D.; Li, J.; et al. Balance Learning for Ship Detection from Synthetic Aperture Radar Remote Sensing Imagery. *ISPRS J. Photogramm. Remote Sens.* **2021**, *182*, 190–207. [CrossRef]
12. Zhang, T.; Zhang, X.; Shi, J.; Wei, S.; Wang, J.; Li, J.; Su, H.; Zhou, Y. Balance Scene Learning Mechanism for Offshore and Inshore Ship Detection in SAR Images. *IEEE Geosci. Remote Sens. Lett.* **2020**, *19*, 4004905. [CrossRef]
13. Zhang, T.; Zhang, X.; Li, J.; Xu, X.; Wang, B.; Zhan, X.; Xu, Y.; Ke, X.; Zeng, T.; Su, H.; et al. SAR Ship Detection Dataset (SSDD): Official Release and Comprehensive Data Analysis. *Remote Sens.* **2021**, *13*, 3690. [CrossRef]
14. Xu, Z.; Zhang, H.; Wang, Y.; Wang, X.; Xue, S.; Liu, W. Dynamic Detection of Offshore Wind Turbines by Spatial Machine Learning from Spaceborne Synthetic Aperture Radar Imagery. *J. King Saud Univ. Com. Inf. Sci.* **2022**, *34*, 1674–1686. [CrossRef]
15. Wei, S.; Zeng, X.; Qu, Q.; Wang, M.; Su, H.; Shi, J. HRSID: A High-Resolution SAR Images Dataset for Ship Detection and Instance Segmentation. *IEEE Access.* **2020**, *8*, 120234–120254. [CrossRef]
16. Su, H.; Wei, S.; Liu, S.; Liang, J.; Wang, C.; Shi, J.; Zhang, X. HQ-ISNet: High-Quality Instance Segmentation for Remote Sensing Imagery. *Remote Sens.* **2020**, *12*, 989. [CrossRef]

17. Zhao, D.; Zhu, C.; Qi, J.; Qi, X.; Su, Z.; Shi, Z. Synergistic Attention for Ship Instance Segmentation in SAR Images. *Remote Sens.* **2021**, *13*, 4384. [CrossRef]
18. Gao, F.; Huo, Y.; Wang, J.; Hussain, A.; Zhou, H. Anchor-Free SAR Ship Instance Segmentation with Centroid-Distance Based Loss. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 11352–11371. [CrossRef]
19. Tianwen, Z.; Xiaowo, X.; Xiaoling, Z. SAR Ship Instance Segmentation Based on Hybrid Task Cascade. In Proceedings of the International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 17–19 December 2021; pp. 530–533.
20. Fan, F.; Zeng, X.; Wei, S.; Zhang, H.; Tang, D.; Shi, J.; Zhang, X. Efficient Instance Segmentation Paradigm for Interpreting SAR and Optical Images. *Remote Sens.* **2022**, *14*, 531. [CrossRef]
21. Zhang, T.; Zhang, X. A Full-Level Context Squeeze-and-Excitation ROI Extractor for SAR Ship Instance Segmentation. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 4506705. [CrossRef]
22. Ke, X.; Zhang, X.; Zhang, T. GCBANet: A Global Context Boundary-Aware Network for SAR Ship Instance Segmentation. *Remote Sens.* **2022**, *14*, 2165. [CrossRef]
23. Zhang, T.; Zhang, X.; Li, J.; Shi, J. Contextual Squeeze-and-Excitation Mask R-CNN for SAR Ship Instance Segmentation. In Proceedings of the IEEE Radar Conference (RadarConf), New York City, NY, USA, 21–25 March 2022; pp. 1–6.
24. Chen, K.; Pang, J.; Wang, J.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Shi, J.; Ouyang, W.; et al. Hybrid Task Cascade for Instance Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4969–4978.
25. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
26. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
27. Hosang, J.; Benenson, R.; Schiele, B. Learning Non-Maximum Suppression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6469–6477.
28. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S. Soft-Nms—Improving Object Detection with One Line of Code. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5562–5570.
29. Wang, J.; Chen, K.; Yang, S.; Loy, C.C.; Lin, D. Region Proposal by Guided Anchoring. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 2960–2969.
30. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
31. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
32. Huang, Z.; Huang, L.; Gong, Y.; Huang, C.; Wang, X. Mask Scoring R-CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 6402–6411.
33. Cai, Z.; Vasconcelos, N. Cascade R-CNN: High Quality Object Detection and Instance Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 1483–1498. [CrossRef]
34. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
35. Rossi, L.; Karimi, A.; Prati, A. A Novel Region of Interest Extraction Layer for Instance Segmentation. In Proceedings of the International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 2203–2209.
36. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. YOLACT: Real-Time Instance Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 9156–9165.
37. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep High-Resolution Representation Learning for Human Pose Estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5686–5696.
38. Gao, S.-H.; Cheng, M.-M.; Zhao, K.; Zhang, X.-Y.; Yang, M.-H.; Torr, P.H. Res2net: A New Multi-Scale Backbone Architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 652–662. [CrossRef]
39. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
40. Xie, S.; Girshick, R.; Dollar, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5987–5995.
41. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep High-Resolution Representation Learning for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3349–3364. [CrossRef]
42. Newell, A.; Yang, K.; Deng, J. Stacked Hourglass Networks for Human Pose Estimation. In Proceedings of the European Conference on Computer Vision, Cham, Switzerland, 8–16 October 2016; pp. 483–499.
43. MacLean, J.; Tsotsos, J. Fast Pattern Recognition Using Gradient-Descent Search in an Image Pyramid. In Proceedings of the International Conference on Pattern Recognition (ICPR), Barcelona, Spain, 3–7 September 2000; pp. 873–877.
44. Zhang, T.; Zhang, X. ShipDeNet-20: An Only 20 Convolution Layers and <1-Mb Lightweight SAR Ship Detector. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 1234–1238. [CrossRef]

45. Zhang, T.; Zhang, X.; Shi, J.; Wei, S. HyperLi-Net: A hyper-light deep learning network for high-accurate and high-speed ship detection from synthetic aperture radar imagery. *ISPRS J. Photogramm. Remote Sens.* **2020**, *167*, 123–153. [CrossRef]
46. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
47. Niu, Z.; Zhong, G.; Yu, H. A Review on the Attention Mechanism of Deep Learning. *Neurocomputing* **2021**, *452*, 48–62. [CrossRef]
48. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected Crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
49. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. In Proceedings of the 4th International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016; pp. 1–13.
50. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the European Conference on Computer Vision, Cham, Switzerland, 8–14 September 2018; pp. 833–851.
51. Lin, T.-Y.; Dollar, 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
52. Zhou, Z.; Guan, R.; Cui, Z.; Cao, Z.; Pi, Y.; Yang, J. Scale Expansion Pyramid Network for Cross-Scale Object Detection in SAR Images. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Brussels, Belgium, 11–16 July 2021; pp. 5291–5294.
53. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft Coco: Common Objects in Context. In Proceedings of the 13th European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
54. Everingham, M.; Eslami, S.M.A.; Gool, L.V.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
55. Wang, J.; Chen, K.; Xu, R.; Liu, Z.; Loy, C.C.; Lin, D. Carafe: Content-Aware Reassembly of Features. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 3007–3016.
56. Zeiler, M.D.; Krishnan, D.; Taylor, G.W.; Fergus, R. Deconvolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 2528–2535.
57. Shelhamer, E.; Long, J.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651. [CrossRef] [PubMed]
58. Zhang, T.; Zhang, X.; Ke, X. Quad-FPN: A Novel Quad Feature Pyramid Network for SAR Ship Detection. *Remote Sens.* **2021**, *13*, 2771. [CrossRef]
59. 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 (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 821–830.
60. Cao, Y.; Xu, J.; Lin, S.; Wei, F.; Hu, H. GCNet: Non-Local Networks Meet Squeeze-Excitation Networks and Beyond. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea, 27–28 October 2019; pp. 1971–1980.
61. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-Local Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
62. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef]
63. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32 nd International Conference on Machine Learning (ICML), Lile, France, 6–11 July 2015; pp. 448–456.
64. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 764–773.
65. Kang, M.; Ji, K.; Leng, X.; Lin, Z. Contextual Region-Based Convolutional Neural Network with Multilayer Fusion for SAR Ship Detection. *Remote Sens.* **2017**, *9*, 860. [CrossRef]
66. Shrivastava, A.; Gupta, A.; Girshick, R. Training Region-Based Object Detectors with Online Hard Example Mining. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 761–769.
67. He, K.; Girshick, R.; Doll´ar, P. Rethinking ImageNet Pre-Training. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 4917–4926.
68. Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; He, K. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *arXiv* **2017**, arXiv:1706.02677.
69. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv* **2019**, arXiv:1906.07155.
70. Eric, Q. Floating-Point Fused Multiply–Add Architectures. Ph.D. Thesis, The University of Texas at Austin, Austin, TX, USA, 2007.
71. Zhang, T.; Zhang, X.; Shi, J.; Wei, S. Depthwise Separable Convolution Neural Network for High-Speed SAR Ship Detection. *Remote Sens.* **2019**, *11*, 2483. [CrossRef]

72. Zhang, T.; Zhang, X. High-Speed Ship Detection in SAR Images Based on a Grid Convolutional Neural Network. *Remote Sens.* **2019**, *11*, 1206. [CrossRef]
73. Zhang, T.; Zhang, X. Injection of Traditional Hand-Crafted Features into Modern CNN-Based Models for SAR Ship Classification: What, Why, Where, and How. *Remote Sens.* **2021**, *13*, 2091. [CrossRef]