remote sensing

MDPI

Article

MSSDet: Multi-Scale Ship-Detection Framework in Optical Remote-Sensing Images and New Benchmark

Weiming Chen [1], Bing Han [1,*], Zheng Yang [1] and Xinbo Gao [1,2]

[1] School of Electronic Engineering, Xidian University, Xi'an 710071, China
[2] Chongqing Key Laboratory of Image Cognition, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
* Correspondence: bhan@xidian.edu.cn

Abstract: Ships comprise the only and most important ocean transportation mode. Thus, ship detection is one of the most critical technologies in ship monitoring, which plays an essential role in maintaining marine safety. Optical remote-sensing images contain rich color and texture information, which is beneficial to ship detection. However, few optical remote-sensing datasets are open publicly due to the issue of sensitive data and copyrights, and only the HRSC2016 dataset is built for the ship-detection task. Moreover, almost all general object detectors suffer from the failure of multi-scale ship detection because of the diversity of spatial resolution and ship size. In this paper, we re-annotate the HRSC2016 dataset and supplement 610 optical remote-sensing images to build a new open source ship-detection benchmark dataset with rich multi-scale ship objects named the HRSC2016-MS dataset. In addition, we further explore the potential of a recursive mechanism in the field of object detection and propose a novel multi-scale ship-detection framework (MSSDet) in optical remote-sensing images. The success of detecting multi-scale objects depends on the hierarchical pyramid structure in the object-detection framework. However, the inherent semantic and spatial gaps among hierarchical pyramid levels seriously affect detection performance. To alleviate this problem, we propose a joint recursive feature pyramid (JRFP), which can generate semantically strong and spatially refined multi-scale features. Extensive experiments were conducted on the HRSC2016-MS, HRSC2016, and DIOR datasets. Detailed ablation studies directly demonstrated the effectiveness of the proposed JRFP architecture and also showed that the proposed method has excellent generalizability. Comparisons with state-of-the-art methods showed that the proposed method achieves competitive performance, i.e., 77.3%, 95.8%, and 73.3% mean average precision accuracy on the HRSC2016-MS, HRSC2016, and DIOR datasets, respectively.

Keywords: MSSDet; HRSC2016-MS; ship detection; multi-scale detection; optical remote-sensing images

check for updates

Citation: Chen, W.; Han, B.; Yang, Z; Gao, X. MSSDet: Multi-Scale Ship-Detection Framework in Optical Remote-Sensing Images and New Benchmark. *Remote Sens.* **2022**, *14*, 5460. https://doi.org/10.3390/rs14215460

Academic Editors: Gangyao Kuang, Olga Sykioti and Xin Su

Received: 15 September 2022
Accepted: 26 October 2022
Published: 30 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the past few decades, many countries have launched numerous satellites and rapidly developed drone technology. Along with the remarkable progress of imaging technology, optical remote-sensing images with high resolution can be easily acquired from the sensors on satellites or aerial cameras on drones nowadays. As the main mode of ocean transportation, the monitoring of ship targets is of great significance in the fields of fishery management, smuggling activities, vessel traffic services, and naval warfare [1,2]. In particular, due to political interest in security, maritime and dock surveillance has been highly prized. Remote sensing has the advantages of remote operation and having a wide monitoring range, and is thus widely used in ocean monitoring. Ship detection based on remote-sensing images is an important part of ocean monitoring, and is thus attracting increasing attention. Synthetic aperture radar (SAR) images used to be the main data source for ship detection because SAR can work day and night, and it can also resist interference from clouds or other factors. However, with the long-term development of optical remote-sensing technology, optical imagery can provide more details, such as color and texture

information, that will significantly help ship detectors distinguish the foreground and background of images. Therefore, ship detection in optical remote-sensing images has attracted increasing research interest in recent years [3–7].

Previously, the methods based on handcrafted features [8–10] or statistical distributions [1,11] held a dominant position. However, as the spatial resolution of optical remote-sensing images enters the sub-meter level, high-resolution optical remote-sensing images provide more detailed information about the objects, but more unexpected complex background information as well. Traditional ship-detection methods are often simple to calculate, but they are only designed for specific scenarios and usually perform with poor robustness in complex scenes. The tremendous progress of deep learning offered new hope for ship detection in complex scenes, particularly with the rapid development of deep convolutional neural networks (CNNs), and CNN-based methods have greatly facilitated the development of object detection in natural images. Encouraged by the object-detection methods in natural images, several CNN-based object detection methods have been introduced into the field of remote sensing [12–14].

In this paper, we study non-oriented ship detection in optical remote-sensing images. Owing to the particularity of optical remote-sensing images, a performance gap between expectation and reality exists when a general object detector is deployed on optical remote-sensing images directly. Three difficulties that hinder general object detectors from application to the field of remote sensing are the following.

1. The first difficulty is the detection of multi-scale objects. In general object detection, the most effective way to detect multi-scale objects is by extracting multi-scale features through a feature pyramid network (FPN) [15] or its variants [16–22]. However, the scale difference among targets in the ship-detection task is much larger, e.g., a fishery boat may only occupy a few pixels, while a cargo ship may occupy thousands of pixels in the same optical remote-sensing image with spatial resolution of 10 m, which is a great challenge for detectors with traditional hierarchical pyramid structures.
2. The second difficulty is that a complex scenario may confuse the detector. Unlike the natural images, the scenarios of remote-sensing images often contain more elements, e.g., the reflection of the sea surface, buildings on shore, ship targets, harbors, and so on. Recently, many researchers have introduced the mechanism of looking and thinking twice, and they have proposed several recursive methods that can deal with various difficult situations by accumulating useful signals [23–27]. In particular, [26,27] are designed for the object-detection task, but they appear weak when coping with complex scenes in optical remote-sensing images.
3. The third problem is the insufficiency of the dataset, which leads to difficulty measuring the robustness of different methods. Only the HRSC2016 [28] dataset is built for the ship-detection task in optical remote-sensing images. The HRSC2016 dataset only contains 1070 optical remote-sensing images with 2976 ship instances, which is obviously inadequate in terms of data volume and diversity. Thus, a new benchmark dataset for ship detection in optical remote-sensing images is urgently needed.

In the present work, we built a multi-scale benchmark for the ship-detection task in optical remote-sensing images by extending and re-annotating the HRSC2016 dataset, which is named HRSC2016-MS. Additionally, a novel multi-scale ship-detection framework (MSSDet) is proposed. A joint recursive feature pyramid (JRFP) is proposed by combining the advantages of the FPN and recursive mechanism to further improve the multi-scale feature-representation capability. The features are iteratively processed by the backbone network and pyramid levels, and the semantically strong and spatially refined multi-scale features are generated. Extensive experiments on the proposed HRSC2016-MS dataset, the original HRSC2016 dataset, and another challenging dataset, i.e., DIOR [29], demonstrate the effectiveness of the proposed method. Furthermore, we evaluate the generalizability of the method on the DIOR dataset, and the result shows that the proposed method also has a strong ability to detect other geospatial targets in optical remote-sensing images.

The rest of this paper is organized as follows. In Section 2, the related work, such as ship-detection methods, multi-scale detection methods, recursive methods, and existing optical remote-sensing datasets, are illustrated. Section 3 provides the details of the proposed HRSC2016-MS dataset. The proposed method is described in Section 4. Experimental results are presented in Section 5. Discussions about limitations of this work are given in Section 6. Finally, Section 7 concludes the paper.

2. Related Works

2.1. Ship Detection

Ship detection in remote-sensing images has been studied for many years. Previously, some researchers aimed to design more efficient handcrafted features [8–10], while other researchers focused on analyzing the statistical distribution of the dataset [1,11]. Inspired by object detection in natural images, many deep-learning-based methods were introduced into the ship-detection field. The two main categories of ship-detection methods are non-oriented and oriented methods.

2.1.1. Non-Oriented Ship-Detection Methods

The non-oriented ship-detection task is similar to the object-detection task in natural images, and it predicts a horizontal bounding box (HBB) for each object. Thus, many general object-detection methods can be adapted to the non-oriented ship-detection task, such as Faster R-CNN and its variants [18,30–37], YOLO and its variants [38–43], etc. In addition, several researchers have proposed algorithms for the non-oriented ship-detection task. Zou et al. [44] proposed a simple and effective ship-detection method named SVDNet based on CNN and the singular-value-decomposition algorithm. Yao et al. [45] used a deep CNN to extract features from the input image and then used a region proposal network (RPN) to discriminate ship targets and regress the detection bounding boxes, in which the anchors are designed according to the intrinsic shape of ship targets. Li et al. [46] first used a RPN to generate ship candidates and then used a hierarchical selective module to map the features in different scales into a space with the same scale, so that the proposed method can detect ships in different scales. Yang et al. [47] proposed a region-based deep forest to overcome the challenge of cluttered scenes and variable appearances of ships in remote-sensing images. Nie et al. [48] proposed a new method of achieving inshore ship detection based on Mask R-CNN, in which soft-non-maximum suppression (Soft-NMS) was introduced into the framework to improve the robustness to nearby inshore ships.

2.1.2. Oriented Ship-Detection Methods

Because ship targets usually have large aspect ratios and contain direction information, using HBBs to detect ship targets may contain unnecessary background information. In addition, non-oriented ship-detection methods cannot predict the directions of ship targets and have poor performance when it comes to detecting densely arranged ships. Oriented ship-detection methods are thus proposed to solve these issues, which predict an oriented bounding box (OBB) for each object. Jiang et al. [49] proposed an arbitrary-oriented text-detection method named a rotational region CNN (R2CNN). Owing to the similarity between scene text and oriented ship-detection tasks, a R2CNN was introduced in the oriented ship-detection task and achieved good performance. Ma et al. [50] proposed a rotation-based framework for arbitrary-oriented text detection named a rotation RPN (RRPN). RRPN is an improvement of R2CNN in which rotated anchor boxes are used in the stage of generating region proposals that can improve the quality of region proposals. RRPN also was introduced in the oriented ship-detection task and achieved good performance. Liu et al. [51] modified the R2CNN and introduced it to the oriented ship-detection task and achieved good detection performance. Yang et al. [4] proposed a new detection model based on a multi-task rotational region CNN to achieve oriented ship detection. Yang et al. [52] proposed a framework called a rotation-dense feature pyramid

network (R-DFPN), which can detect ship targets in different scenarios, e.g., in-shore and off-shore, effectively.

2.2. Multi-Scale Features

Previous object detectors directly use the multi-scale features extracted by the backbone network [53,54]. After that, the top-down FPN [15] structure was proposed to combine the features at different scales sequentially. Since the development of the FPN, many manually and automatically designed FPN variants have been proposed to further improve multi-scale feature-representation capability. G-FRNet [55] adds feedback with gating units to address the ambiguous information in the passing forward information flow. PANet [16] adds another bottom-up path on the top of the FPN to boost the information flow in the proposal-based instance segmentation framework. STDN [17] explores the inter-scale consistency nature across multiple detection scales by the scale-transfer module in the network. BFP [18] balances the feature levels in FPN by feature integration and refining operations. NAS-FPN [19], Auto-FPN [20], and NAS-FCOS [21] use neural architecture search [56] to find new FPN structures. EfficientDet [22] proposes a weighted bi-directional FPN (BiFPN) that allows easy and fast multi-scale feature fusion.

2.3. Recursive Methods

The effectiveness of the looking and thinking twice mechanism has already been demonstrated in many modern computer-vision tasks. Liang et al. [23] proposed a recurrent CNN (RCNN) for object recognition by incorporating recurrent connections into each convolutional layer. Kim et al. [24] advanced a deeply recursive convolutional network (DRCN) for the image super-resolution task. Tai et al. [25] proposed a very deep CNN model named a deep recursive residual network (DRRN) for the image super-resolution task. Liu et al. [26] proposed a recursive method named CBNet for the object-detection task that cascades the output features of multiple backbones as the input of the FPN. Qiao et al. [27] explored the recursive backbone design and proposed DetectoRS for object detection.

2.4. Optical Remote-Sensing Datasets

Many optical remote-sensing datasets are not publicly available due to the issue of sensitive data and copyrights. Therefore, many existing works were based on private datasets with data sources that include SPOT-5, WorldView-2, Quickbird, the Venezuelan Remote Sensing Satellite, GaoFen-1, and Google Earth [44,57–61]. In addition, it is difficult for the public to access the original data from satellites or drones. Thus, many researchers cannot access research work on ship detection directly. For the purpose of allowing more researchers to participate in researching optical remote-sensing-image detection, several research groups have released their publicly available optical remote-sensing datasets [28,29,62–72] (see Table 1) in which only a few datasets contain the ship category [28,29,64,69–72]. It is worth mentioning that only the HRSC2016 dataset [28] is built for ship detection. The HRSC2016 dataset only contains 1070 images with 2976 ship instances, which is insufficient for the demands of developing ship detection in optical remote-sensing images.

Table 1. Existing public optical remote-sensing datasets. * denotes that corresponding dataset contains ship category.

Dataset	Categories	Images	Instances	Year
TAS [62]	1	30	1319	2008
SZTAKI-INRIA [63]	1	9	665	2012
* NWPU VHR-10 [64]	10	800	3775	2014
VEDAI [65]	9	1210	3640	2015

Table 1. *Cont.*

Dataset	Categories	Images	Instances	Year
UCAS-AOD [66]	2	910	6029	2015
DLR 3K Vehicle [67]	2	20	14,235	2015
* HRSC2016 [28]	1	1070	2976	2016
RSOD [68]	4	976	6950	2017
* DOTA-v1.0 [69]	15	2806	188,282	2018
* DOTA-v1.5 [70]	16	2806	403,318	2019
* DIOR [29]	20	23,463	192,472	2019
* HRRSD [71]	13	26,722	55,740	2019
* DOTA-v2.0 [72]	18	11,268	1,793,658	2021

3. HRSC2016-MS Dataset

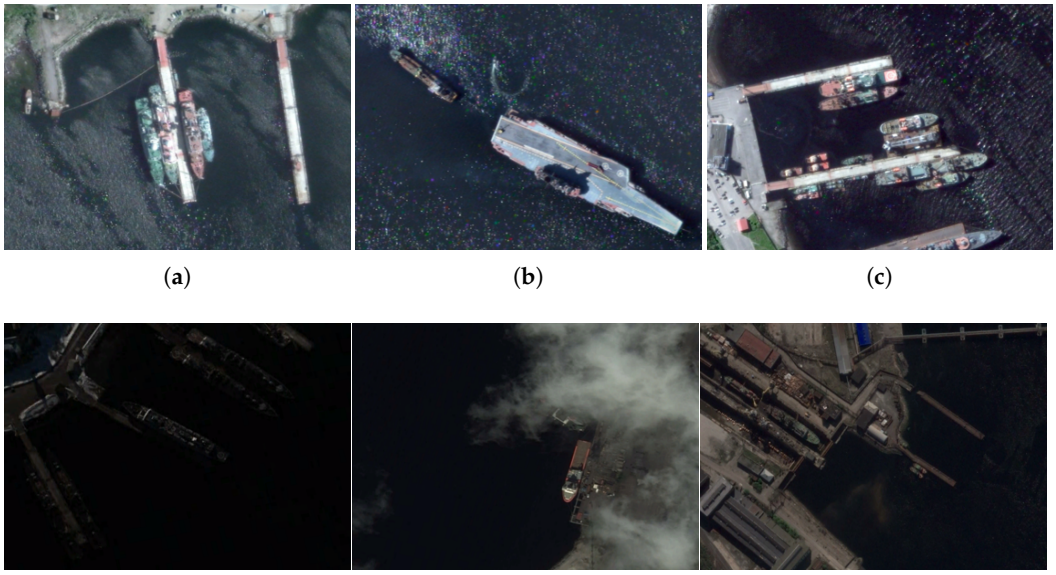
HRSC2016 is the only open-source optical remote-sensing ship dataset. Owing to the difficulty of data acquisition and annotation, this dataset only contains 1070 images with 2976 ship instances. Moreover, most of the objects in the HRSC2016 dataset are larger than those in natural images. As is well known, the detection of small and size-varied objects is a challenging problem. Therefore, we built an open-source ship detection benchmark with rich multi-scale ship targets named HRSC2016-MS. We present a detailed description of the proposed dataset herein.

3.1. Data Collection

The optical remote-sensing images in HRSC2016-MS come from two sources: previous studies on HRSC2016 and new collections from Google Earth. The first part of HRSC2016-MS is composed of the original data in HRSC2016, which is captured from the harbors in the United States. The original HRSC2016 dataset only contains 1070 optical remote-sensing images with 2976 ship instances, and a large number of small-ship instances are missing annotations, which motivated us to re-annotate the original data. The second part of HRSC2016-MS consists of 610 optical remote-sensing images collected from Google Earth captured from Murmansk harbor, Russia. Several examples of new collections are shown in Figure 1. The new collections include sea and sea–land images, day and night scenarios, clear and cloudy weather, and multi-scale ship instances. A comparison of the proposed HRSC2016-MS dataset and original HRSC2016 dataset is shown in Table 2. It can be seen that the proposed dataset contains a larger number of images and a wider range of image sizes than the original dataset. More importantly, the proposed dataset includes more multi-scale ship instances with a broader range of aspect ratios than the original dataset. Therefore, the diversity of the HRSC2016-MS dataset is increased compared with the original dataset, and the ship-detection task for this dataset is more challenging.

Table 2. Comparison of proposed and original datasets.

	HRSC2016 [28]	HRSC2016-MS
Images	1070	1680
Image sizes	481 × 411∼1238 × 837	361 × 339∼1329 × 830
Instances	2976	7655
Instance sizes	48 × 22∼713 × 505	5 × 10∼489 × 739
Instance aspect ratios	0.102∼10.0	0.092∼11.692



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 1. Several examples of new collections in proposed dataset. The new collections include multiple scenarios: Scenes of (**a**) ships docking in shore, (**b**) ships in the middle of the ocean, (**c**) multi-scale ship instances densely arranged in shore, (**d**) a dark night, (**e**) a cloudy day, and (**f**) ships on shore.

3.2. Category Selection

We visualized the original annotation files in the HRSC2016 dataset and found that numerous ship instances were mislabeled. The label set in the original dataset is a tree structure containing 1, 4, and 27 classes in ship-class, ship-category, and ship-type levels, respectively. Undoubtedly, an elaborate label set contributes to the convergence of the detector during the training period, but such a label set will create the problem of annotation omission because a limited label set cannot cover all types of ships. Therefore, in the proposed dataset, the label set only covers one class, i.e., ship.

3.3. Annotation Method

We used roLabelImg [73] to label the images in the proposed dataset. Each ship instance is annotated in two ways, i.e., using a HBB and an OBB. To ensure the consistency and reduce the subjectivity of annotators, we invited $P_a = 5$ subjects to annotate each instance. $H_i(x_{\min i}, y_{\min i}, x_{\max i}, y_{\max i})$, and $O_i(x_{ci}, y_{ci}, w_i, h_i, \theta_i)$ are used to denote the annotation by the ith subject for a HBB and an OBB, respectively, where $(x_{\min}, y_{\min})$ is the coordinate of the upper left-hand corner, $(x_{\max}, y_{\max})$ the coordinate of the bottom right-hand corner, (x_c, y_c) the center coordinate, w and h are the width and height of an OBB, respectively, and θ denotes the angle between an OBB and horizontal direction. The final annotation of a instance is calculated with the average annotating agreement from all of the subjects. For a HBB, the average annotating agreement from all of the subjects A_H is defined by

$$A_H = (\overline{x_{\min}}, \overline{y_{\min}}, \overline{x_{\max}}, \overline{y_{\max}}) \tag{1}$$

For an OBB, the average annotating agreement from all of the subjects A_O is defined by

$$A_O = \left(\overline{x_c}, \overline{y_c}, \overline{w}, \overline{h}, \overline{\theta}\right) \tag{2}$$

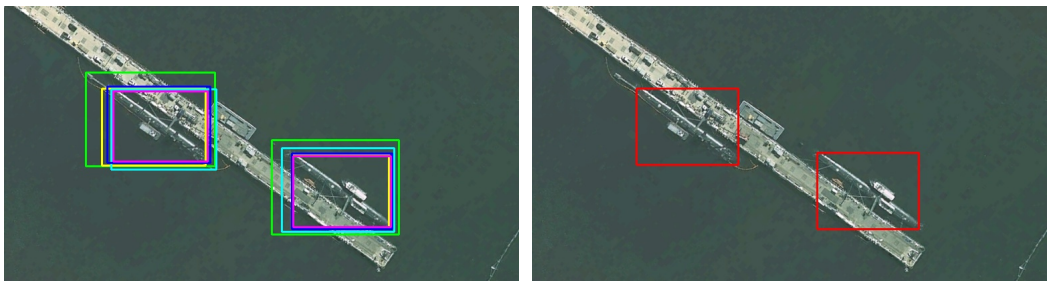
where all of the means of parameters are calculated with the same formula, i.e., $\overline{p} = \frac{1}{P_a}\sum_{i=1}^{P_a} p_i$. It is worth mentioning that the annotating outliers are discarded to avoid some serious labeling mistakes. We design a simple algorithm called the average annotating agreement algorithm (AAAA) to filter the annotating outliers, which is shown in Algorithm 1. The AAAA assumes that the annotating outliers are only a minority, and thus the outliers can simply be discarded

based on medians. As shown in Figure 2, the proposed method can mitigate the influence caused by annotating subjectivity.

Algorithm 1: Average Annotating Agreement.

Input: Annotation from all subjects H_i and O_i where $i = 1, 2, \ldots, 5$
Output: Average agreement A_H and A_O

while *not end if there exist outliers or* $P_a < 3$ **do**
  Pick the median from each parameter;
  for $j = 1$ *to* $P_a - 1$ **do**
    Calculate the distances d_j from medians to all the other values;
  end
  Calculate the average distance m;
  for $j = 1$ *to* $P_a - 1$ **do**
    if $d_j > m$ *and* $\frac{d_j - m}{m} > 0.1$ **then**
      The corresponding annotation is an outlier which should be discarded;
      $P_a - -$;
    end
  end
end
Calculate A_H and A_O based on the remaining annotations;



(**a**) Original annotation (**b**) Average annotation

Figure 2. Annotation examples: (**a**) annotations marked by different subjects in different color rectangles, and (**b**) average annotation without subjectivity.

We annotated the new images and re-annotated the original images with the proposed annotation method. We reduced the dimensions of the dataset to two based on principal component analysis (PCA) and then visualized the data distribution. As shown in Figure 3, we analyzed the data distribution of the original and proposed datasets and found that the two datasets have similar data distributions.

Following the standard COCO [74] evaluation approach, objects with sizes smaller than 32×32 pixels are small, objects with sizes larger than 96×96 pixels are large, and the rest are medium objects. The number of small, medium, and large objects in the original HRSC2016 dataset is 0, 319, and 2657, respectively, while it is 880, 2447, and 4328 in the proposed HRSC2016-MS dataset. Obviously, the proposed dataset contains more multi-scale ship instances, especially small objects.

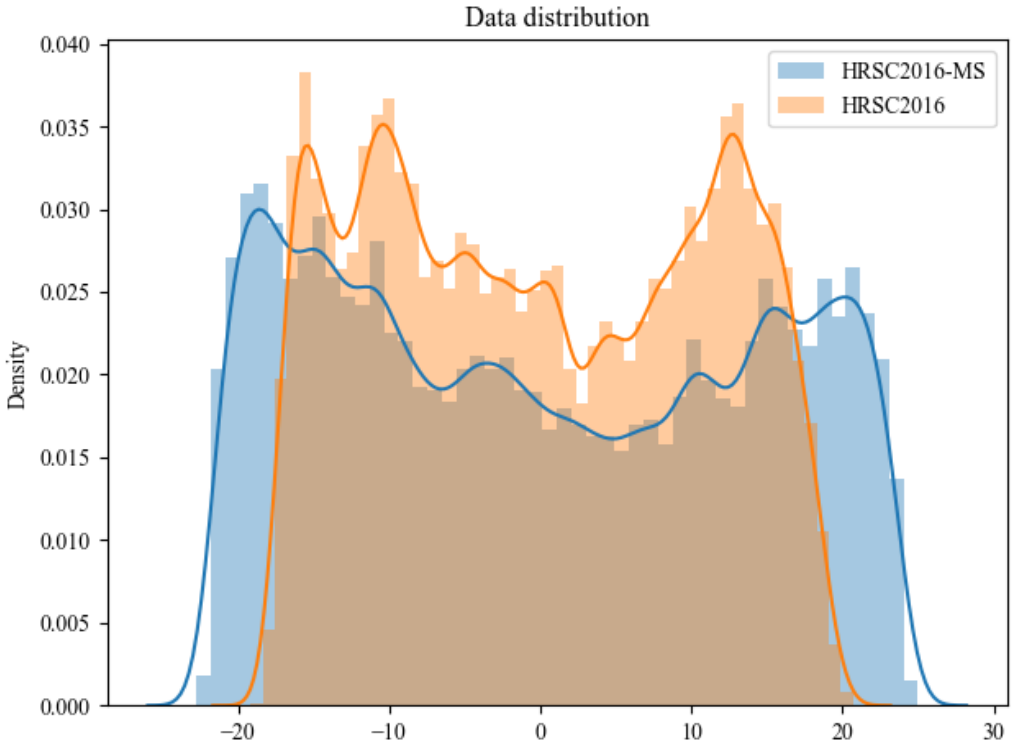


Figure 3. Data distribution of original HRSC2016 and proposed HRSC2016-MS datasets.

4. Proposed Method

HBBs are usually applied in the fields of both non-oriented ship detection and general object detection to locate objects. Thus, it is easy to deploy a general object-detection model to the non-oriented ship-detection task. Moreover, non-oriented ship-detection methods have the advantage of strong robustness and can perform well in complex scenarios compared with oriented ship-detection methods. In particular, for small-scale ship targets, with orientations that are difficult to recognize, it is unsuitable to detect them using an oriented method. Therefore, we developed a non-oriented detection algorithm for the multi-scale ship-detection task.

As discussed above, the existing non-oriented detectors suffer from the failure caused by weak multi-scale feature-representation capability. Therefore, we proposed the JRFP to improve the multi-scale feature-representation capability by accumulating useful signals in the recursive structure and narrowing semantic gaps among levels.

4.1. Joint Recursive Feature Pyramid

The JRFP architecture is shown in Figure 4, and is composed of two parts, i.e., the bottom-up backbone network and the top-down pyramid levels. The backbone network extracts features by using several cascaded stages, which include some combinations of convolutional layers, pooling layers, normalization layers, and activation functions. In the backbone network, the features extracted by higher stages are small in size but correspond to large reception fields, and thus, these features include poor spatial information but rich semantic information. Letting B_i denote the feature-extraction operation in the ith stage of the backbone network, the output feature C_i of the ith stage in the backbone network is defined by

$$C_i = B_i(C_{i-1}) \tag{3}$$

Along the top-down pyramid levels, the spatially coarser but semantically stronger features from higher levels are up-sampled twice and then added to lower levels. As a result, higher-resolution features with strong semanticity are generated.

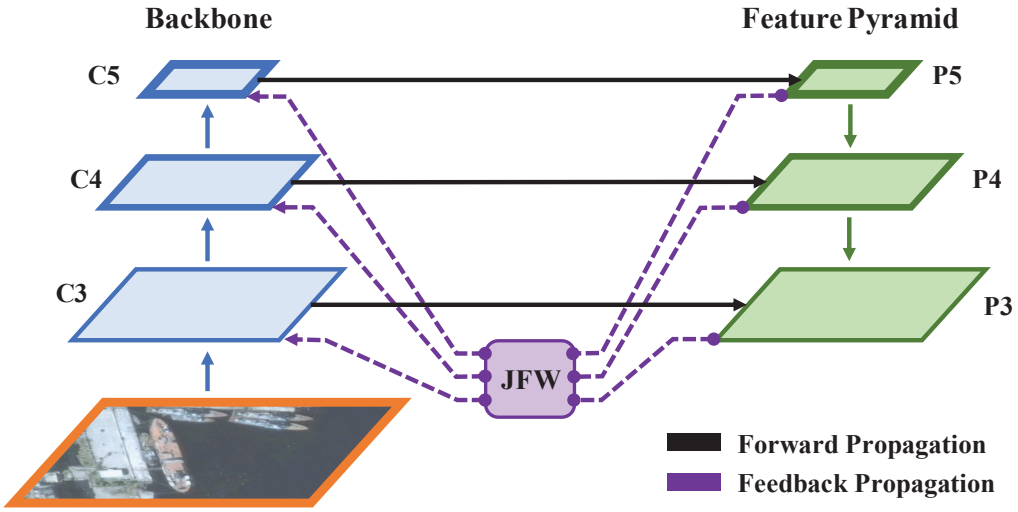


Figure 4. In the schema of JRFP, C3, C4, and C5 denote the output features of stages in the backbone network, respectively, and P3, P4, and P5 denote the features in pyramid levels, respectively.

The JRFP is a recursive structure, and the signals in the JRFP have two methods of propagation, i.e., forward propagation and feedback propagation. The forward propagation in the JRFP is the same as in the FPN, i.e., the output features from each stage in the backbone network go through a separate 1×1 convolutional layer with a small channel number to become a part of the level in pyramid levels. Doing the above-mentioned top-down operation results in the pyramid levels. Letting U_i denote the ith up-sampling operation in pyramid levels, the output feature P_i of the ith level in pyramid levels is defined by Equation (4):

$$P_i = \begin{cases} \mathrm{U}_i(P_{i+1}) + \mathrm{Conv}_{i,1\times1}(C_i), & i = 3,4 \\ \mathrm{Conv}_{i,1\times1}(C_i), & i = 5 \end{cases} \tag{4}$$

The feedback propagation in the JRFP is that the feedback signals from pyramid levels are fused by the joint feedback worker (JFW) and then allotted back to the backbone network. Letting W_i denote the ith output of JFW and JF denote the feedback signal fusing operation in JFW, the output feature C_i of the ith stage in the backbone network can be redefined by

$$C_i = \mathrm{B}_i(C_{i-1} + \mathrm{W}_i) \tag{5}$$

where

$$[\mathrm{W}_3, \mathrm{W}_4, \mathrm{W}_5] = \mathrm{JF}(P_3, P_4, P_5) \tag{6}$$

For the convenience of implementation, we unrolled the recursive JRFP architecture to a sequential architecture in practice. In addition, a hyper-parameter named the recursive index (RI) is introduced to control the recursive times in the model. Therefore, the output feature of the ith stage in backbone and the output feature of the ith level in pyramid levels can be redefined as $\forall t = 1, \ldots, \mathrm{RI}$,

$$C_i^t = \mathrm{B}_i(C_{i-1}^t + \mathrm{W}_i^t) \tag{7}$$

$$P_i^t = \begin{cases} \mathrm{U}_i(P_{i+1}^t) + \mathrm{Conv}_{i,1\times1}(C_i^t), & i = 3,4 \\ \mathrm{Conv}_{i,1\times1}(C_i^t), & i = 5 \end{cases} \tag{8}$$

where

$$[\mathrm{W}_3^t, \mathrm{W}_4^t, \mathrm{W}_5^t] = \mathrm{JF}(P_3^t, P_4^t, P_5^t) \tag{9}$$

In particular, the JRFP architecture degrades to the FPN architecture when RI = 0.

4.2. Joint Feedback Worker

As discussed above, because previous recursive methods ignore the inherent semantic gaps among different levels, they only achieve sub-optimal performance when detecting multi-scale objects in complex scenarios. The semantic gap among different pyramid levels is intrinsically caused by the huge difference between reception fields in the backbone network, as shown in Figure 5. In the bottom-up backbone network, the filters in shallow layers correspond to very small reception fields towards the original image, and these small regions only contain some texture and color information that is semantically weak. After a stage-by-stage feature-extraction step, the filters in deeper layers correspond to larger reception fields toward the original image. These large reception fields may include some core features of the target, which are beneficial when it comes to recognizing the target accurately. In addition, in previous FPN-based methods, the highest level in the feature pyramid only loses information because the highest feature map only goes through a 1×1 convolutional layer to reduce the channel number but does not have a higher level to fuse with. The information loss at the highest level of the feature pyramid may restrict the multi-scale detection capability of the detector. The JFW processes feedback signals jointly, which lets the information from low levels have the potential to flow to the highest level and thus improves the multi-scale detection capability of the detector.

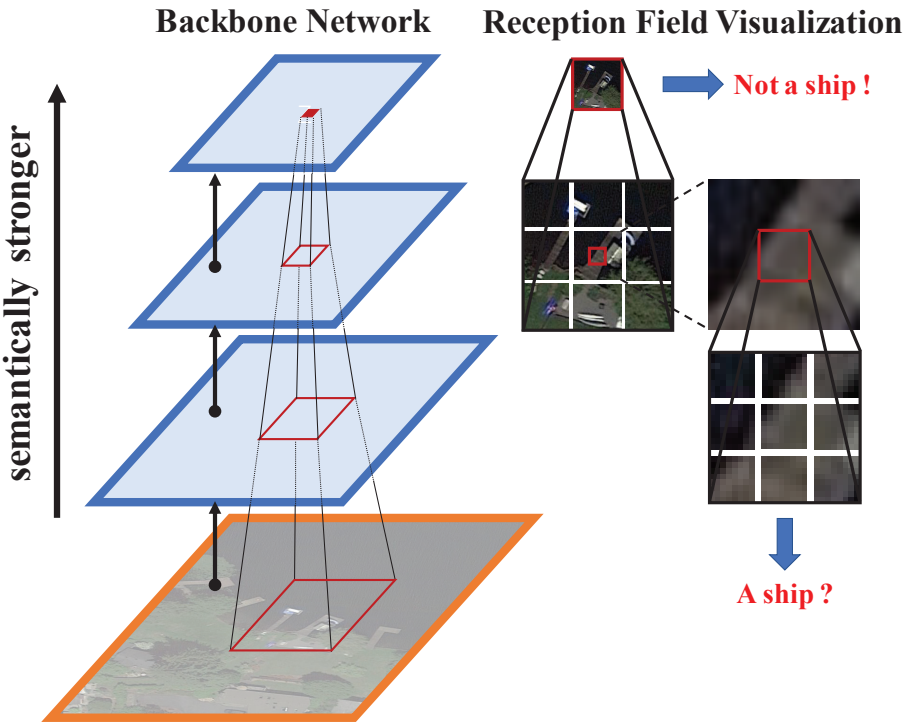


Figure 5. Visualization example of reception field. Along with the bottom-up backbone network, the filters in higher layers correspond to the larger reception fields, and thus, the output features of higher layers are semantically stronger. As shown in the example, we cannot confirm whether it is a ship based on low-layer features, while it is easy to confirm it is not a ship based on high-layer features.

JFW is the core component of the proposed JRFP, and can narrow the semantic gap by fusing and liberating all of the feedback signals jointly. The structure of JFW is shown in Figure 6. The input of JFW is the feedback signals from pyramid levels, and its outputs are connected to the corresponding stages in the backbone network. As the shapes of features vary at different levels, up-sampling operations are needed to make all of the input features uniform, and down-sampling operations are needed to recover the processed features into the corresponding shapes. In addition, channel numbers in pyramid levels are different from those in the backbone network, and thus down-sampling operations are followed by 1×1 convolutional layers, which are used to adjust channel numbers to match those

in the backbone network. The core of JFW is the attention blocks in which the semantic gap among different levels can be narrowed and the specificity of each level reserved. The attention operation in JFW is a cascade of channel and spatial attention.

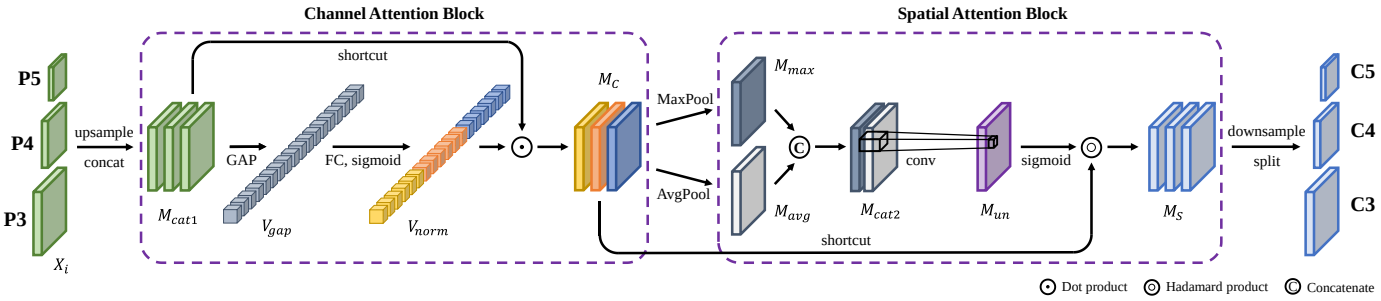


Figure 6. JFW structure. The channel attention and spatial attention blocks are two main components of JFW, in which the channel attention block is cascaded, followed by the spatial attention block. In the channel attention block, the feature map is weighted by a channel descriptor that is squeezed by global spatial information. In the spatial attention block, the feature map is further weighted by a spatial descriptor that focuses on the informative part of the feature map.

The channel attention block is a variant of the SE block [75], which squeezes the global spatial information into a channel descriptor. There are two pathways in the channel attention block, i.e., the channel-wise pathway and short-cut pathway. In the channel-wise pathway, the inputs $X_i \in \mathbf{R}^{\mathbf{H}\times\mathbf{W}\times\mathbf{C_i'}}$ are concatenated to be one feature map $M_{cat1} \in \mathbf{R}^{\mathbf{H}\times\mathbf{W}\times\sum_i \mathbf{C_i}'}$. Then, the global average pooling layer squeezes M_{cat1} to a feature vector $V_{gap} \in \mathbf{R}^{\mathbf{1}\times\mathbf{1}\times\sum_i \mathbf{C_i}'}$. After that, V_{gap} goes through a group of fully connected (FC) layers, a batch normalization (BN) layer, and a ReLU activation operation, and becomes a further compressed feature vector $V_{fc1} \in \mathbf{R}^{\mathbf{1}\times\mathbf{1}\times\overline{\mathbf{C'}}}$. Then, V_{fc1} is liberated by another FC layer, obtaining $V_{fc2} \in \mathbf{R}^{\mathbf{1}\times\mathbf{1}\times\sum_i \mathbf{C_i}'}$. Next, V_{fc2} is normalized into the region of $[0, 1]$ by the sigmoid function, and the normalized feature vector is denoted $V_{norm} \in \mathbf{R}^{\mathbf{1}\times\mathbf{1}\times\sum_i \mathbf{C_i}'}$. Finally, the output channel attention map $M_C \in \mathbf{R}^{\mathbf{H}\times\mathbf{W}\times\sum_i \mathbf{C_i}'}$ is the dot product of the short-cut connection M_{cat1} and the normalized feature vector V_{norm}, which can be defined by

$$M_C = M_{cat1} \cdot V_{norm} \tag{10}$$

The spatial attention block is similar to that in CBAM [76], which utilizes the inter-spatial relationship of features. The channel information of the channel attention map is aggregated by max-pooling and average-pooling operations, and results in two feature maps: $M_{max} \in \mathbf{R}^{\mathbf{H}\times\mathbf{W}\times\mathbf{1}}$ and $M_{avg} \in \mathbf{R}^{\mathbf{H}\times\mathbf{W}\times\mathbf{1}}$. Then, the max-pooled and average-pooled feature maps are concatenated into one feature map $M_{cat2} \in \mathbf{R}^{\mathbf{H}\times\mathbf{W}\times\mathbf{2}}$. After that, the concatenated feature map M_{cat2} is convolved by a 7×7 convolutional layer and obtains the non-normalized feature map $M_{un} \in \mathbf{R}^{\mathbf{H}\times\mathbf{W}\times\mathbf{1}}$. Next, M_{un} is normalized by the sigmoid function and the normalized feature map $M_{norm} \in \mathbf{R}^{\mathbf{H}\times\mathbf{W}\times\mathbf{1}}$ obtained. The output spatial attention map $M_S \in \mathbf{R}^{\mathbf{H}\times\mathbf{W}\times\sum_i \mathbf{C_i}'}$ is the Hadamard product of the channel attention map M_C and normalized feature map M_{norm}, i.e.,

$$M_S = M_C \circ M_{norm} \tag{11}$$

Finally, the spatial attention map is split and aligned to the corresponding stage in the backbone network after the down-sampling and splitting operation.

5. Experimental Section

5.1. Datasets and Evaluation Metric

We evaluated the proposed MSSDet on the proposed HRSC2016-MS dataset, original HRSC2016 dataset, and another challenging non-oriented detection dataset, namely, the DIOR dataset.

The HRSC2016 dataset was split into training, validation, and testing sets with 436, 181, and 453 images, respectively. For the HRSC2016-MS dataset, we randomly split the dataset into training, validation, and testing sets, which contain 610, 460, and 610 images, respectively. The DIOR dataset contains 20 categories: airplane (PL), airport (PO), baseball field (BF), basketball court (BC), bridge (BR), chimney (CN), dam (DM), expressway service area (ESA), expressway toll station (ETS), golf course (GC), ground track field (GTF), harbor (HB), overpass (OP), ship (SH), stadium (SD), storage tank (ST), tennis court (TC), train station (TS), vehicle (VC), and windmill (WM). Following the original setting in the DIOR dataset, the number of images in training, validation, and testing sets are 5863, 5862, and 11,738, respectively.

We adopted mean average precision (mAP) as the evaluation metric. Note that the definition of mAP in our experiments is the same as that in the PASCAL VOC 2012 object detection challenge [77], and the IoU threshold of mAP is 0.5. In addition, we use AP_S, AP_M, and AP_L to evaluate the detection performance of small, medium, and large ship targets in the ablation study Section 5.4. The definitions of the small, medium, and large targets are the same as those in the standard COCO evaluation approach.

5.2. Implementation Details

To evaluate the ship-detection performance of the proposed method on the DIOR dataset, we only used the images that contain ship instances to train the detector. We adopted ResNet-34 [78] as the data selector to select the images that contain ship targets. Specifically, we used full training and validation sets to train the selector and only used the images with ship instances in training and validation sets to train the detector during the training procedure. During the testing procedure, the well-trained ResNet-34 was used to filter the images with ship targets and discard those without ship targets, and the detector only inferences the images that are selected by the selector.

We employed MMDetection [79] to implement the proposed method. In all experiments, we adopted the stochastic gradient descent (SGD) optimizer to optimize models. In addition, we used random flipping, brightness adjustment, contrast adjustment, saturation adjustment, and hue adjustment for the standard data-augmentation pipeline. Considering that the data volume of the HRSC2016 and HRSC2016-MS datasets is not very large, we added an extra Mosaic-MixUp data-augmentation method [43] into the standard pipeline. For the experiments on the HRSC2016 and HRSC2016-MS datasets, the initial learning rate, momentum, and weight decay were set to 0.002, 0.9, and 0.0001, respectively, and we trained all of the models for 100 epochs. The learning rate warmed up for 5 epochs with the warm-up ratio set to 0.001, then the learning rate updating schedule followed the cosine annealing policy in which the minimum learning rate was set as 0.05 times the initial learning rate, and the learning rate of the last 15 epochs was fixed to the initial learning rate. For the main comparison experiments on the DIOR dataset, the training procedure was composed of two parts, i.e., training the data selector and training the detector. During the training period of the data selector, the initial learning rate, momentum, and weight decay were set to 0.01, 0.9, and 0.0001, respectively. The data selector was trained for 100 epochs and the learning rate decay to 10 times at 30, 60, and 90 epochs, respectively. During the detector training period, the initial learning rate was changed to 0.0001 while the momentum and weight decay remained the same, and the other training settings were the same as those of the HRSC2016 and HRSC2016-MS datasets. For the model generalizability evaluation experiment on the DIOR dataset, the entire dataset was used to train and evaluate models. The initial learning rate, momentum, and weight decay were set to 0.002, 0.9, and 0.0001, respectively. Considering the time and computational

resource consumption, we trained all models except the YOLOv3 for 30 epochs, in which the learning rate warmed up for 1000 iterations with the warm-up ratio set to 0.001 and the learning rate decay to 10 times at 15 and 25 epochs. Because the single-stage model needs more training epochs, we trained the YOLOv3 model for 100 epochs with the same learning rate settings except for the learning rate decay strategy, which decays 10 times at 50 and 75 epochs. All of the experiments were conducted on a single TITAN X (Pascal) graphics card with a total batch size of 1 for training.

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

We conducted three experiments on HRSC2016, the proposed HRSC2016-MS, and DIOR datasets to compare the proposed MSSDet with several state-of-the-art methods. All of the detection results of comparison methods were obtained by implementing these methods given in the literature and not taken from the original papers. The best results are shown in bold for clarity.

5.3.1. Results on HRSC2016 Dataset

Most existing works on the HRSC2016 focus on oriented detection because the ship targets in this dataset usually have large aspect ratios. However, this work concentrates on multi-scale non-oriented detection. Thus, for fairness on the HRSC2016, the comparison methods are selected according to two principles. The first principle is having special neck structure designs, such as NAS-FPN, PANet, Libra R-CNN, DetectoRS, and YOLOF. The second principle is representative state-of-the-art methods, such as SSD, RetinaNet, Faster R-CNN, FCOS, Mask R-CNN, Cascade R-CNN, HTC, and YOLOX. The detection results on the HRSC2016 test set (with asterisks) are shown in Table 3. Note that most of these methods use the same backbone ResNet-50 as us, while only two methods, SSD and YOLOX, adopt different backbones. It is obvious that most methods with neck structure have better performance than other methods. In addition, HTC achieves better mAP than all of the competitors, while our method brings a 0.4% improvement above HTC when using ResNet-50 as the backbone.

Table 3. Comparison with state-of-the-art methods on proposed HRSC2016-MS dataset and original HRSC2016 dataset. * denotes results on HRSC2016 dataset. Note that the semantic branch in HTC is removed, as semantic segmentation annotations are not available in these two datasets. The best result is highlighted in bold.

Method	Backbone	mAP
SSD [54]	VGG16	86.6 */45.8
YOLOF [80]	ResNet-50	91.2 */56.5
RetinaNet [81]	ResNet-50	86.7 */58.6
NAS-FPN [19]	ResNet-50	91.1 */60.8
FCOS [82]	ResNet-50	91.6 */61.0
PANet [16]	ResNet-50	94.3 */65.1
Mask R-CNN [32]	ResNet-50	93.2 */65.1
Faster R-CNN [30]	ResNet-50	93.5 */66.1
Cascade R-CNN [33]	ResNet-50	93.0 */66.5
DetectoRS [27]	ResNet-50	94.2 */66.6
Libra R-CNN [18]	ResNet-50	93.7 */67.0
YOLOX [43]	CSPDarknet	82.4 */68.3
HTC [83]	ResNet-50	94.5 */69.0
MSSDet (proposed)	ResNet-50	94.9 */75.7
	ResNet-101	95.3 */76.6
	ResNet-152	**95.8** */**77.3**

MSSDet is implemented with not only ResNet-50, but also ResNet-101 and ResNet-152 in further experiments to get a convincing comparative study. The results are given in the last two rows of Table 3. Our MSSDet equipped with ResNet-152 achieves the

best performance with 95.8% mAP, which is 1.6% better than DetectoRS and 1.3% better than HTC.

5.3.2. Results on HRSC2016-MS Dataset

We conducted two experiments to demonstrate the effectiveness of the proposed method and the diversity of the proposed dataset. For a fair comparison, the same methods as those in the experiments on the HRSC2016 were chosen. The results on the HRSC2016-MS testing set (without asterisks) are shown in Table 3. It can be seen that the proposed MSSDet achieves superior results among these methods. The proposed MSSDet significantly outperforms those methods with special neck structure designs, which achieve improvements of 18.8%, 14.9%, 10.6%, 9.1%, and 8.7% over YOLOF, NAS-FPN, PANet, DetectoRS, and Libra R-CNN, respectively. In particular, the proposed recursive method overcomes DetectoRS, which is known as the best recursive method in the field of object detection so far. YOLOF demonstrated that the success of the FPN is because of its divide-and-conquer strategy for the optimization problem in object detection rather than multi-scale feature fusion. The proposed method combines the advantages of divide-and-conquer strategy, multi-scale feature fusion, and the recursive mechanism, and thus achieves improvements by a large margin up to 19.1% mAP over YOLOF. In addition, we conducted an experiment to compare the ship-detection capability of the proposed MSSDet with different backbone networks. We replaced ResNet-50 with ResNet-101 and ResNet-152, and obtained 0.9% and 1.6% mAP improvements, respectively.

It is noteworthy that the accuracies of almost all of the comparison methods drop by over 20% compared to the results on the HRSC2016 dataset. The proposed HRSC2016-MS dataset contains more multi-scale ship targets than the original HRSC2016 dataset, and thus the ship-detection task on the proposed dataset is more challenging.

5.3.3. Results on DIOR Dataset

The DIOR dataset is the largest optical remote-sensing image dataset for horizontal object detection so far. It contains 20 common geospatial categories, including the ship category. As the present work is concentrated on non-oriented ship detection in optical remote-sensing images, it is worth evaluating the proposed model on the DIOR dataset.

As discussed before, to accelerate the inference speed and save computational resources, we adopted ResNet-34 as a data selector to filter the images without ship instances. We first used the full training and validation sets to train ResNet-34 and obtain a well-trained data selector with a test accuracy of 98.6%. Then, a manually selected training set that contains 1302 images was used to train the detector. Several representative methods were selected for comparisons, such as SSD, YOLOv3, RetinaNet, Mask R-CNN, PANet, CornerNet, Faster R-CNN, Cascade R-CNN, HTC, and DetectoRS. The detection results are shown in Table 4. The proposed MSSDet achieves the best, i.e., 70.6%, mAP detection performance among the comparisons, which not only demonstrates the effectiveness of the proposed method, but also demonstrates the method's robustness on different datasets. In particular, the proposed method outperforms DetectoRS without bells and whistles, which further demonstrates the success of the proposed JFRP architecture. In addition, we conducted an additional experiment in which the backbone network was replaced by with ResNet-101 and ResNet-152. It can be seen from the last two rows of Table 4 that ResNet-101 and ResNet-152 bring 1.3% and 2.7% mAP detection improvements, respectively. Some of the comparison methods adopt different backbone networks from that of the proposed method, but different backbones do not provide significant changes. Thus, a deeper backbone network can improve detection performance, but it cannot play a decisive role.

Table 4. Comparison with state-of-the-art methods on DIOR dataset. Note that the semantic branch in HTC is removed due to semantic segmentation annotations not being available in the DIOR dataset. The best result is highlighted in bold.

Method	Backbone	mAP
SSD [54]	VGG16	56.1
YOLOv3 [40]	Darknet-53	63.7
RetinaNet [81]	ResNet-50	64.1
Mask R-CNN [32]	ResNet-50	64.3
PANet [16]	ResNet-50	65.2
CornerNet [84]	Hourglass-104	65.8
Faster R-CNN [30]	ResNet-50	66.6
Cascade R-CNN [33]	ResNet-50	67.3
HTC [83]	ResNet-50	67.6
DetectoRS [27]	ResNet-50	68.1
MSSDet (proposed)	ResNet-50	70.6
	ResNet-101	71.9
	ResNet-152	**73.3**

5.4. Ablation Studies

We conducted two ablation studies to evaluate the module in MSSDet on the proposed HRSC2016-MS dataset. For a fair comparison, the baseline model is HTC with the FPN structure and the semantic branch removed. In addition, all of the methods were equipped with ResNet-50 as the backbone and utilized the same training settings.

5.4.1. Evaluation of Joint Recursive Feature Pyramid

To validate the effectiveness of the proposed JRFP, we compared it with five manually designed neck structures, i.e., FPN, PAFPN, BFP, BiFPN, and RFP. We ignored the neck structures that search by NAS, such as Auto-FPN and NAS-FPN, because they often suffer from failure caused by dependency on the specific dataset. The results are reported in Table 5. The baseline method achieves a detection performance of 52.5% mAP. Compared with the baseline method, the existing multi-scale approaches FPN, PAFPN, BPN, BiFPN, and RFP gain 13.9%, 15.7%, 15.7%, 9.6%, and 15.6% mAP detection improvement, respectively. The proposed JRFP achieves the top performance, i.e., 75.7% mAP, which is 23.2% mAP higher than that of the baseline method. To illustrate the multi-scale detection capability of the proposed JRFP, we further evaluated the detection performance of small, medium, and large ship targets. The proposed JRFP shows overwhelming advancement in detecting small and medium ship targets, i.e., 11% and 8.4% AP greater than the second-best method, respectively, which further demonstrates that the proposed JRFP can offer a considerable improvement in multi-scale detection.

Table 5. Ablation study on JRFP effectiveness. The best result is highlighted in bold.

Method	Backbone	mAP	AP_S	AP_M	AP_L
Baseline	ResNet-50	52.5	5.90	41.7	70.8
Baseline + FPN [15]	ResNet-50	66.3	16.4	61.3	79.7
Baseline + PAFPN [16]	ResNet-50	68.2	14.4	62.7	82.9
Baseline + BFP [18]	ResNet-50	68.2	17.0	63.2	81.3
Baseline + BiFPN [22]	ResNet-50	62.1	12.2	59.4	75.0
Baseline + RFP [27]	ResNet-50	68.1	14.5	64.2	81.0
Baseline + JRFP (proposed)	ResNet-50	**75.7**	**28.0**	**72.6**	**85.5**

Several visual comparisons of ship-detection results of different hierarchical pyramid structures on challenging examples on the HRSC2016-MS dataset are illustrated in Figure 7. Four representative optical remote-sensing images containing multi-scale ship instances

are selected to display the visualization results, which intuitively show that the baseline method with the proposed JRFP as the neck structure has the best detection performance, directly demonstrating the effectiveness of the proposed JRFP architecture. In contrast, the baseline methods with other neck structures suffer from missing detection and detecting false alarms.

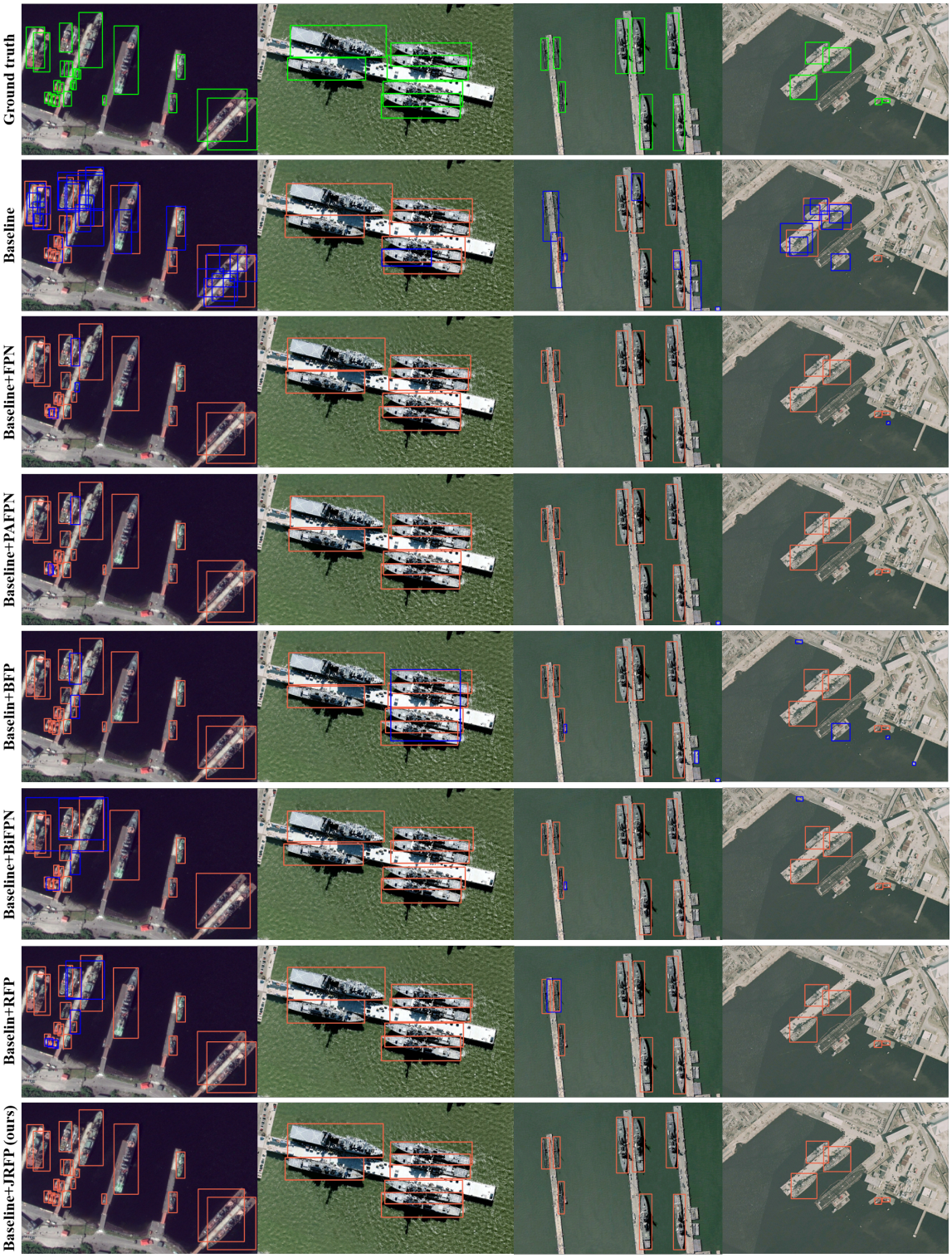


Figure 7. Visual results of hierarchical pyramid structure ablation study on HRSC2016-MS dataset. Bounding boxes with **green**, **orange**, and **blue** colors represent ground truths, true positive predictions, and false alarms, respectively.

5.4.2. Recursive Index Evaluation

RI is an important hyper-parameter that controls the recursive times in JRFP, which may not only affect the detection performance, but also the memory consumption and inference speed significantly. We evaluated the performance of JRFP by different RIs. The model used in this experiment is the baseline method equipped with the JRFP, in which the baseline method is the same as that in the previous ablation study. The results are listed in Table 6. It can be observed that the mAP first increases and then slightly decreases when RI changes from 1 to 3. Moreover, the model achieves the highest, i.e., 75.7%, mAP detection performance when RI = 2. Essentially, RI balances the uniformity and specificity of each level by controlling the times of feature fusion. If RI is too small, the semantic gap among different levels may affect the detection performance. If RI is too large, all of the levels tend to be similar and lose the specific information of each level. In addition, with increasing RI, the required computational resources also increase accordingly.

Table 6. Ablation study on recursive index. The best result is highlighted in bold.

RI	1	2	3
mAP	73.9	**75.7**	74.2

5.5. Evaluation of Model Generalizability

As ships are not the only targets in optical remote-sensing images, it is necessary to evaluate the generalizability of the proposed model. The DIOR dataset was selected to assess the generalizability of the proposed MSSDet because it contains the most target categories among optical remote-sensing datasets. We used the entire training and validation sets to train the detector and test the well-trained detector on the test set directly. Fifteen representative or state-of-the-art methods were selected for extensive comparisons, i.e., R-CNN, RICNN, RICAOD, RIFD-CNN, SSD, Faster R-CNN, Mask R-CNN, YOLOv3, PANet, CornerNet, RetinaNet, Cascade R-CNN, DetectoRS, HTC, and AFPN. The selected comparisons are widely used for object detection in natural images and remote-sensing images. The detection results are shown in Table 7.

Table 7. Evaluation of generalizability of MSSDet on DIOR dataset. Results colored **red** or **blue** indicate best or second-best result, respectively, of each category.

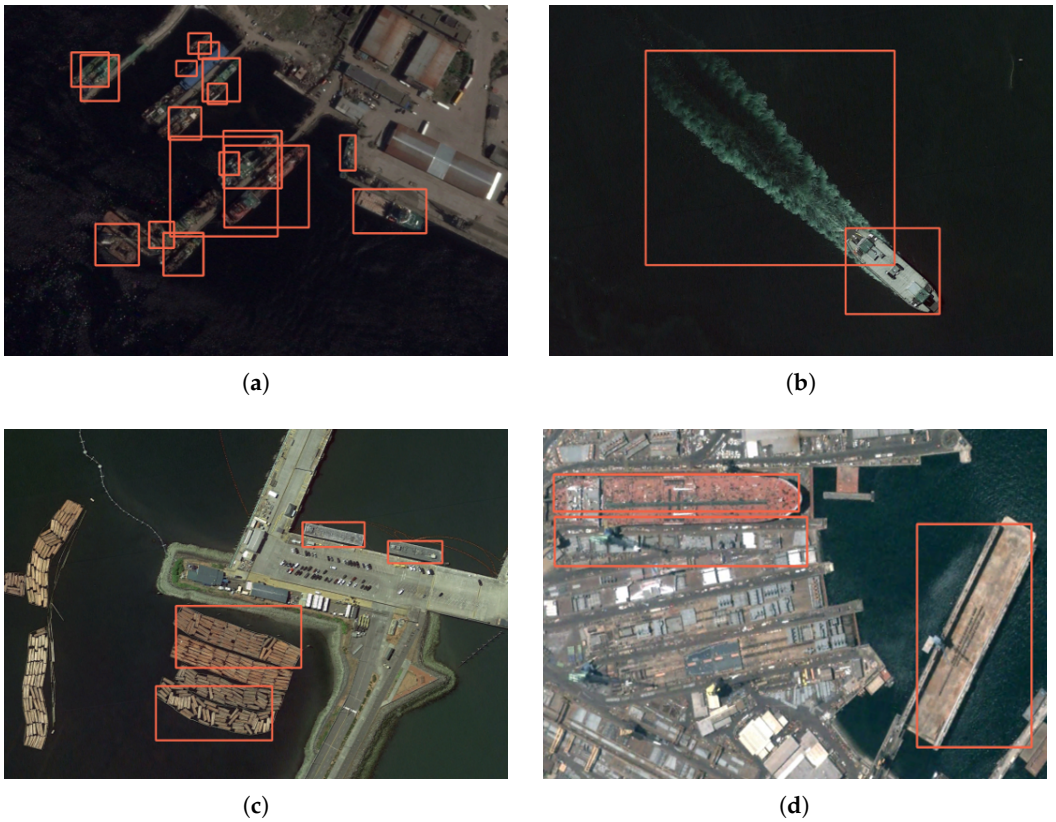
Method	Backbone	PL	PO	BF	BC	BR	CN	DM	ESA	ETS	GC	GTF	HB	OP	SH	SD	ST	TC	TS	VC	WM	mAP
R-CNN [85]	VGG16	35.6	43.0	53.8	62.3	15.6	53.7	33.7	50.2	33.5	50.1	49.3	39.5	30.9	9.1	60.8	18.0	54.0	36.1	9.1	16.4	37.7
RICNN [86]	VGG16	39.1	61.0	60.1	66.3	25.3	63.3	41.1	51.7	36.6	55.9	58.9	43.5	39.0	9.1	61.1	19.1	63.5	46.1	11.4	31.5	44.2
RICAOD [87]	VGG16	42.2	69.7	62.0	79.0	27.7	68.9	50.1	60.5	49.3	64.4	65.3	42.3	46.8	11.7	53.5	24.5	70.3	53.3	20.4	56.2	50.9
RIFD-CNN [88]	VGG16	56.6	53.2	79.9	69.0	29.0	71.5	63.1	69.0	56.0	68.9	62.4	51.2	51.1	31.7	73.6	41.5	79.5	40.1	28.5	46.9	56.1
SSD [54]	VGG16	59.5	72.7	72.4	75.7	29.7	65.8	56.6	63.5	53.1	65.3	68.6	49.4	48.1	59.2	61.0	46.6	76.3	55.1	27.4	65.7	58.6
Faster R-CNN [30]	ResNet-50	54.1	71.4	63.3	81.0	42.6	72.5	57.5	68.7	62.1	73.1	76.5	42.8	56.0	71.8	57.0	53.5	81.2	53.0	43.1	80.9	63.1
Mask R-CNN [32]	ResNet-50	53.8	72.3	63.2	81.0	38.7	72.6	55.9	71.6	67.0	73.0	75.8	44.2	56.5	71.9	58.6	53.6	81.1	54.0	43.1	81.1	63.5
CornerNet [84]	Hourglass-104	58.8	84.2	72.0	80.8	46.4	75.3	64.3	81.6	76.3	79.5	79.5	26.1	60.6	37.6	70.7	45.2	84.0	57.1	43.0	75.9	64.9
RetinaNet [81]	ResNet-50	53.7	77.3	69.0	81.3	44.1	72.3	62.5	76.2	66.0	77.7	74.2	50.7	59.6	71.2	69.3	44.8	81.3	54.2	45.1	83.4	65.7
Cascade R-CNN [33]	ResNet-50	57.8	82.4	69.6	87.1	48.8	79.6	67.7	82.7	70.9	84.2	81.6	56.5	63.6	72.5	67.1	56.5	85.5	63.0	43.5	85.1	70.3
YOLOv3 [40]	DarkNet-53	67.8	81.0	78.6	88.0	50.2	77.3	64.2	85.9	72.5	78.7	75.4	52.9	59.8	73.6	67.2	**62.5**	87.0	58.9	**50.8**	86.8	71.0
PANet [16]	ResNet-50	62.5	84.8	72.2	87.9	48.6	78.4	69.3	83.4	69.9	81.6	82.9	54.3	62.6	73.4	73.3	58.1	87.2	64.8	42.5	84.9	71.1
DetectoRS [27]	ResNet-50	56.5	83.4	80.3	87.7	44.0	**81.7**	**72.6**	86.1	72.0	81.2	84.4	60.2	56.5	73.1	79.2	61.3	85.5	67.3	43.4	78.9	71.8
HTC [83]	ResNet-50	68.8	83.8	75.5	87.9	50.6	80.6	64.7	84.1	73.3	83.1	83.5	58.3	64.1	74.7	75.0	62.0	87.5	62.7	45.9	85.7	72.6
AFPN [89]	ResNet-50	68.0	87.0	74.9	88.9	47.8	77.7	68.8	84.2	71.3	76.9	83.1	59.0	61.3	73.6	76.2	62.1	87.6	67.8	46.7	**88.6**	72.6
MSSDet (proposed)	ResNet-50	68.7	87.4	79.9	88.5	54.8	81.2	70.4	87.3	74.2	82.7	84.1	60.4	63.8	79.8	**81.6**	61.6	87.4	68.3	45.5	85.6	74.7
	ResNet-101	**70.2**	**88.7**	**80.4**	**89.2**	**56.3**	81.4	72.2	**89.3**	**77.4**	**85.9**	**84.7**	**62.5**	**65.7**	**81.4**	81.3	59.0	**88.6**	**71.6**	45.6	87.5	**75.9**
	ResNet-152	**70.7**	**88.6**	**81.8**	**90.4**	**56.5**	**82.5**	**73.0**	**90.1**	**78.6**	**86.6**	**85.6**	**63.5**	**66.5**	**82.5**	**82.0**	**63.3**	**88.7**	**71.7**	**46.7**	**89.2**	**76.9**

In Table 7, the results of R-CNN, RICNN, RICAOD, RIFD-CNN, SSD, Faster R-CNN, Mask R-CNN, RetinaNet, CornerNet, and AFPN are taken from the original papers [29,89], while the rest are inferred by our re-implemented versions. It can be seen that the proposed method achieves the top overall performance, i.e., 74.7% mAP detection accuracy, which

demonstrates the great generalizability of the proposed MSSDet. In addition, deeper backbone networks can also provide detection improvement, i.e., the proposed method achieves 75.9% and 76.9% mAP overall detection performance when ResNet-101 and ResNet-152, respectively, are employed as the backbone network. In particular, the proposed method achieves the best ship-detection performance, i.e., 79.8%, 81.4%, and 82.5% mAP. Note that the detection accuracy of the ship category in this experiment is higher than that in Table 4. The reason is that the entire test set was directly used in this experiment, while there are a few images with ship targets that were wrongly discarded by the data selector in the corresponding experiments in Table 4. Moreover, the proposed method performs well in many categories, i.e., it (equipped with ResNet-152 as the backbone network) obtains 18 of the best results and 2 of the second-best results among 20 categories, which shows that it also has a great ability to detect other geospatial targets in optical remote-sensing images.

6. Discussion

Some undesirable detection results are shown in Figure 8. In Figure 8a, some densely arranged ship instances are missed. This case is mainly due to the limitations of non-oriented detection methods. In Figure 8b, the sea wave trail is incorrectly included in the bounding box. This phenomenon is largely due to an insufficient ability to extract context information. In Figure 8c, the bamboo rafts are detected by mistake. This case is due to the lack of corresponding training samples in the training set. In Figure 8d , the buildings on shore are detected by mistake. This phenomenon is because some ship targets are on the shore, which confuses the detector. We believe that these problems can be solved by introducing the oriented detection mechanism, more abundant context information, and a more extensive and diverse dataset, which will be further researched in planned future work.



(**a**) (**b**)

(**c**) (**d**)

Figure 8. Examples of several failure cases on HRSC2016-MS dataset. (**a**) False and duplicate detection due to densely arranged objects. (**b**) False detection due to the sea wave trail. (**c**) False detection due to similar objects on the ocean. (**d**) False detection due to similar objects on shore.

Our approach is concentrated on the non-oriented ship-detection task because it is difficult or even not sensible to detect the orientations of small ship targets. In contrast, it is necessary to detect the orientations of the medium-sized and large ship targets because ship objects in remote-sensing images are often shown as large aspect ratios. Along with the development of optical remote-sensing imaging technology, the spatial resolutions of optical remote-sensing images are becoming increasingly high, and the sizes of ship targets in the images are becoming increasingly large. Thus, detecting the orientations of ship objects in high-resolution optical remote-sensing images seems to be more critical. In addition, the ships docking beside harbors are often densely arranged. Detection of such densely arranged ship targets may cause omissions, as the intersection over union (IoU) among these objects is often large, and NMS may delete the corresponding predicted HBBs in the post-processing procedure. In contrast, the oriented ship-detection task does not have such trouble because the OBBs of densely arranged ship targets have low IoUs. Therefore, the design of oriented ship detection will be considered in future work.

7. Conclusions

A new optical remote-sensing benchmark dataset for ship detection is constructed in the present work. The proposed dataset, named HRSC2016-MS, consists of two parts, i.e., the original data in the HRSC2016 dataset and the new collections from Google Earth. Specifically, we re-annotated the original HRSC2016 dataset and extended 610 optical remote-sensing images to build the HRSC2016-MS dataset. Compared with the original HRSC2016 dataset, the proposed dataset is more diverse and contains more multi-scale ship instances.

Moreover, we propose a novel recursive ship-detection method for multi-scale ship detection in optical remote-sensing images, named MSSDet. The core of MSSDet is the JRFP architecture, which can extract semantically strong and spatially refined multi-scale features. In the JRFP structure, the feedback features from pyramid levels are combined and processed jointly by the JFW module. Detailed ablation studies on the proposed HRSC2016-MS dataset and extensive comparison experiments on the HRSC2016-MS, HRSC2016, and DIOR datasets demonstrate the effectiveness of the proposed method. In addition, a further experiment on the DIOR dataset shows that the proposed method also has a great ability to detect other geospatial targets in optical remote-sensing images.

Although the proposed method is competitive in terms of accuracy, we must admit that it is slow and consumes a significant amount of GPU memory. The recursive structure in the JRFP and attention operation in JFW lead to slow inference speed and high memory consumption. Therefore, we will try to accelerate the inference speed and reduce the memory consumption of the proposed method in future work without sacrificing too much detection accuracy.

Author Contributions: Conceptualization, W.C.; methodology, W.C. and B.H.; software, W.C.; validation, W.C., B.H., Z.Y. and X.G.; data curation, W.C.; writing—original draft preparation, W.C.; writing—review and editing, W.C. and B.H.; visualization, W.C.; supervision, B.H.; project administration, B.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Natural Science Foundation of China (Grant Nos. 62076190, 61572384, and 41831072) and in part by The Key Industry Innovation Chain of Shaanxi (Grant No. 2022ZDLGY01-11).

Data Availability Statement: The proposed method and HRSC2016-MS dataset will be available when the paper is accepted.

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

References

1. Proia, N.; Pagé, V. Characterization of a Bayesian Ship Detection Method in Optical Satellite Images. *IEEE Geosci. Remote Sens. Lett.* **2010**, *7*, 226–230. [CrossRef]
2. Zhu, C.; Zhou, H.; Wang, R.; Guo, J. A Novel Hierarchical Method of Ship Detection from Spaceborne Optical Image Based on Shape and Texture Features. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 3446–3456. [CrossRef]
3. Liu, W.; Ma, L.; Chen, H. Arbitrary-Oriented Ship Detection Framework in Optical Remote-Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 937–941. [CrossRef]
4. Yang, X.; Sun, H.; Sun, X.; Yan, M.; Guo, Z.; Fu, K. Position Detection and Direction Prediction for Arbitrary-Oriented Ships via Multitask Rotation Region Convolutional Neural Network. *IEEE Access* **2018**, *6*, 50839–50849. [CrossRef]
5. Guo, H.; Yang, X.; Wang, N.; Song, B.; Gao, X. A Rotational Libra R-CNN Method for Ship Detection. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 5772–5781. [CrossRef]
6. Li, L.; Zhou, Z.; Wang, B.; Miao, L.; Zong, H. A Novel CNN-Based Method for Accurate Ship Detection in HR Optical Remote Sensing Images via Rotated Bounding Box. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 686–699. [CrossRef]
7. Yu, Y.; Yang, X.; Li, J.; Gao, X. A Cascade Rotated Anchor-Aided Detector for Ship Detection in Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14.
8. Sun, H.; Sun, X.; Wang, H.Q. A ship detection method with high-resolution remote sensing images. *Sci. Surv. Mapp.* **2013**, *38*, 112–115.
9. Li, S.; Zhou, Z.; Wang, B.; Wu, F. A Novel Inshore Ship Detection via Ship Head Classification and Body Boundary Determination. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1920–1924. [CrossRef]
10. Wang, H.; Zhu, M.; Lin, C.; Chen, D.B. Ship detection in optical remote sensing image based on visual saliency and AdaBoost classifier. *Optoelectron. Lett.* **2017**, *13*, 151–155. [CrossRef]
11. Corbane, C.; Najman, L.; Pecoudl, E.; Demagistrit, L.; Petit, M. A complete processing chain for ship detection using optical satellite imagery. *Int. J. Remote Sens.* **2010**, *31*, 5837–5854. [CrossRef]
12. Wang, C.; Shi, J.; Yang, X.; Zhou, Y.; Wei, S.; Li, L.; Zhang, X. Geospatial Object Detection via Deconvolutional Region Proposal Network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 3014–3027. [CrossRef]
13. Wang, Y.; Wang, C.; Zhang, H.; Dong, Y.; Wei, S. Automatic Ship Detection Based on RetinaNet Using Multi-Resolution Gaofen-3 Imagery. *Remote Sens.* **2019**, *11*, 531. [CrossRef]
14. Wei, H.; Zhang, Y.; Wang, B.; Yang, Y.; Li, H.; Wang, H. X-LineNet: Detecting Aircraft in Remote Sensing Images by a Pair of Intersecting Line Segments. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 1645–1659. [CrossRef]
15. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
16. 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, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
17. Zhou, P.; Ni, B.; Geng, C.; Hu, J.; Xu, Y. Scale-Transferrable Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 528–537.
18. 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 Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 821–830.
19. Ghiasi, G.; Lin, T.Y.; Le, Q.V. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7029–7038.
20. Xu, A.; Yao, A.; Li, A.; Liang, A.; Zhang, A. Auto-FPN: Automatic Network Architecture Adaptation for Object Detection Beyond Classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6648–6657.
21. Wang, N.; Gao, Y.; Chen, H.; Wang, P.; Tian, Z.; Shen, C.; Zhang, Y. NAS-FCOS: Fast Neural Architecture Search for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11940–11948.
22. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 10778–10787.
23. Liang, M.; Hu, X. Recurrent convolutional neural network for object recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3367–3375.
24. Kim, J.; Lee, J.K.; Lee, K.M. Deeply-Recursive Convolutional Network for Image Super-Resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1637–1645.
25. Tai, Y.; Yang, J.; Liu, X. Image Super-Resolution via Deep Recursive Residual Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2790–2798.
26. Liu, Y.; Wang, Y.; Wang, S.; Liang, T.; Ling, H. CBNet: A Novel Composite Backbone Network Architecture for Object Detection. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 11653–11660. [CrossRef]
27. Qiao, S.; Chen, L.C.; Yuille, A. DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution. *arXiv* **2020**, arXiv:2006.02334.

28. Liu, Z.; Yuan, L.; Weng, L.; Yang, Y. A High Resolution Optical Satellite Image Dataset for Ship Recognition and Some New Baselines. In Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods, Porto, Portugal, 20 October 2016; pp. 324–331.
29. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [CrossRef]
30. 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]
31. Li, Z.; Peng, C.; Yu, G.; Zhang, X.; Deng, Y.; Sun, J. Light-Head R-CNN: In Defense of Two-Stage Object Detector. *arXiv* **2017**, arXiv:1711.07264.
32. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the International Confenrece Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
33. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving Into High Quality Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6154–6162.
34. Lu, X.; Li, B.; Yue, Y.; Li, Q.; Yan, J. Grid R-CNN. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7355–7364.
35. Wu, Y.; Chen, Y.; Yuan, L.; Liu, Z.; Wang, L.; Li, H.; Fu, Y. Rethinking Classification and Localization for Object Detection. *arXiv* **2019**, arXiv:1904.06493.
36. Zhang, H.; Chang, H.; Ma, B.; Wang, N.; Chen, X. Dynamic R-CNN: Towards High Quality Object Detection via Dynamic Training. *arXiv* **2020**, arXiv:2004.06002.
37. Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Tomizuka, M.; Li, L.; Yuan, Z.; Wang, C.; et al. SparseR-CNN: End-to-End Object Detection with Learnable Proposals. *arXiv* **2020**, arXiv:2011.12450.
38. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
39. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
40. Joseph, R.; Ali, F. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
41. Alexey, B.; Chien-Yao, W.; Hong-Yuan, M.L. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
42. Jocher, G. YOLOv5. 2020. Available online: https://github.com/ultralytics/yolov5 (accessed on 20 April 2022).
43. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
44. Zou, Z.; Shi, Z. Ship Detection in Spaceborne Optical Image With SVD Networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 5832–5845. [CrossRef]
45. Yuan, Y.; Jiang, Z.; Zhang, H.; Zhao, D.; Cai, B. Ship detection in optical remote sensing images based on deep convolutional neural networks. *J. Appl. Remote Sens.* **2017**, *11*, 1.
46. Li, Q.; Mou, L.; Liu, Q.; Wang, Y.; Zhu, X.X. HSF-Net: Multiscale Deep Feature Embedding for Ship Detection in Optical Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 7147–7161. [CrossRef]
47. Yang, F.; Xu, Q.; Li, B.; Ji, Y. Ship Detection from Thermal Remote Sensing Imagery through Region-Based Deep Forest. *IEEE Trans. Geosci. Remote Sens.* **2018**, *15*, 449–453. [CrossRef]
48. Nie, S.; Jiang, Z.; Zhang, H.; Cai, B.; Yao, Y. Inshore Ship Detection Based on Mask R-CNN. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 693–696.
49. Jiang, Y.; Zhu, X.; Wang, X.; Yang, S.; Li, W.; Wang, H.; Fu, P.; Luo, Z. R2CNN: Rotational Region CNN for Orientation Robust Scene Text Detection. *arXiv* **2017**, arXiv:1706.09579.
50. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrary-Oriented Scene Text Detection via Rotation Proposals. *IEEE Trans. Multimed.* **2018**, *20*, 3111–3122. [CrossRef]
51. Liu, Z.; Hu, J.; Weng, L.; Yang, Y. Rotated region based CNN for ship detection. In Proceedings of the IEEE International Conference Image Process, Beijing, China, 17–20 September 2017; pp. 900–904.
52. Yang, X.; Hao, S.; Fu, K.; Yang, J.; Xian, S.; Yan, M.; Zhi, G. Automatic Ship Detection in Remote Sensing Images from Google Earth of Complex Scenes Based on Multiscale Rotation Dense Feature Pyramid Networks. *Remote Sens.* **2018**, *10*, 132. [CrossRef]
53. Cai, Z.; Fan, Q.; Feris, R.S.; Vasconcelos, N. A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 354–370.
54. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
55. Islam, M.A.; Rochan, M.; Bruce, N.D.B.; Wang, Y. Gated Feedback Refinement Network for Dense Image Labeling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4877–4885.
56. Barret, Z.; Quoc, V.L. Neural architecture search with reinforcement learning. *arXiv* **2016**, arXiv:1611.01578.
57. Corbane, C.; Pecoul, E.; Demagistri, L.; Petit, M.; Frouin, R.J.; Andrefouet, S.; Kawamura, H.; Lynch, M.J.; Pan, D.; Platt, T. Fully automated procedure for ship detection using optical satellite imagery. *Int. Soc. Opt. Photonics* **2008**, *7150*, 71500R.
58. Yokoya, N.; Iwasaki, A. Object Detection Based on Sparse Representation and Hough Voting for Optical Remote Sensing Imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2053–2062. [CrossRef]

59. Liu, G.; Zhang, Y.; Zheng, X.; Sun, X.; Fu, K.; Wang, H. A New Method on Inshore Ship Detection in High-Resolution Satellite Images Using Shape and Context Information. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 617–621. [CrossRef]
60. Qi, S.; Ma, J.; Lin, J.; Li, Y.; Tian, J. Unsupervised Ship Detection Based on Saliency and S-HOG Descriptor From Optical Satellite Images. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1451–1455.
61. Shi, Z.; Yu, X.; Jiang, Z.; Li, B. Ship Detection in High-Resolution Optical Imagery Based on Anomaly Detector and Local Shape Feature. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 4511–4523.
62. Heitz, G.; Koller, D. Learning Spatial Context: Using Stuff to Find Things. In Proceedings of the European Conference on Computer Vision, Marseille, France, 12–18 October 2008; pp. 30–43.
63. Benedek, C.; Descombes, X.; Zerubia, J. Building Development Monitoring in Multitemporal Remotely Sensed Image Pairs with Stochastic Birth-Death Dynamics. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 33–50. [CrossRef]
64. Cheng, G.; Han, J. A survey on object detection in optical remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2016**, *117*, 11–28. [CrossRef]
65. Razakarivony, S.; Jurie, F. Vehicle detection in aerial imagery: A small target detection benchmark. *J. Vis. Commun. Image Represent.* **2016**, *34*, 187–203. [CrossRef]
66. Zhu, H.; Chen, X.; Dai, W.; Fu, K.; Ye, Q.; Jiao, J. Orientation robust object detection in aerial images using deep convolutional neural network. In Proceedings of the IEEE International Conference Image Process, Quebec City, QC, Canada, 27–30 September 2015; pp. 3735–3739.
67. Liu, K.; Mattyus, G. Fast Multiclass Vehicle Detection on Aerial Images. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1938–1942.
68. Xiao, Z.; Liu, Q.; Tang, G.; Zhai, X. Elliptic Fourier transformation-based histograms of oriented gradients for rotationally invariant object detection in remote-sensing images. *Int. J. Remote Sens.* **2015**, *36*, 618–644. [CrossRef]
69. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
70. Ding, J.; Xue, N.; Long, Y.; Xia, G.S.; Lu, Q. Learning RoI Transformer for Detecting Oriented Objects in Aerial Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
71. Zhang, Y.; Yuan, Y.; Feng, Y.; Lu, X. Hierarchical and Robust Convolutional Neural Network for Very High-Resolution Remote Sensing Object Detection. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5535–5548. [CrossRef]
72. Ding, J.; Xue, N.; Xia, G.S.; Bai, X.; Yang, W.; Yang, M.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; et al. Object Detection in Aerial Images: A Large-Scale Benchmark and Challenges. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 7778–7796. [CrossRef]
73. You, H. roLabelImg. 2017. Available online: https://github.com/cgvict/roLabelImg (accessed on 14 September 2022).
74. 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 European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
75. 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]
76. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 3–19.
77. Everingham, M.; Gool, L.V.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
78. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
79. 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.
80. Chen, Q.; Wang, Y.; Yang, T.; Zhang, X.; Cheng, J.; Sun, J. You Only Look One-level Feature. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 13034–13043.
81. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the International Conference Computer Vision, Venice, Italy, 20–29 October 2017; pp. 2999–3007.
82. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. In Proceedings of the International Confenrece Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9626–9635. [CrossRef]
83. 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 Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4969–4978.
84. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the 15th European Conference on Computer Vision, ECCV, Munich, Germany, 8–14 September 2018; pp. 765–781.
85. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
86. Cheng, G.; Zhou, P.; Han, J. Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7405–7415. [CrossRef]

87. Li, K.; Cheng, G.; Bu, S.; You, X. Rotation-Insensitive and Context-Augmented Object Detection in Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 2337–2348. [CrossRef]
88. Cheng, G.; Han, J.; Zhou, P.; Xu, D. Learning Rotation-Invariant and Fisher Discriminative Convolutional Neural Networks for Object Detection. *IEEE Trans. Image Process.* **2019**, *28*, 265–278. [CrossRef] [PubMed]
89. Cheng, G.; He, M.; Hong, H.; Yao, X.; Qian, X.; Guo, L. Guiding Clean Features for Object Detection in Remote Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]