An Antijamming and Lightweight Ship Detector Designed for Spaceborne Optical Images

Huanqian Yan, Bo Li, Hong Zhang, and Xingxing Wei

***Abstract*—Ship detection in spaceborne optical images is a challenging task because ships have various orientations and scales, especially complex backgrounds, i.e., ships are easily obscured by various jamming. Moreover, most object detectors have enormous computation and parameter numbers, which are unsuitable for resource-bounded spaceborne platforms that contain restrictive memory access and computation. In this article, in order to mitigate the influence of complex backgrounds and jamming on detection, and improve the practicality of detection algorithms, a new satellite optical image dataset and a novel ship detector are proposed. We have collected a new dataset from the satellite, which contains images of different time periods, different illuminations, and different levels of jamming. The proposed dataset is different from the widely used public remote sensing datasets, it is more practical and challenging. The proposed ship detector can deal with various images well and is robust to various complex backgrounds. Specifically, a feature refining module is designed to extract features effectively, which can improve detection performance significantly. An antijamming module is proposed to highlight the features of objects in the whole feature map. In contrast to mainstream ship detectors, the proposed method is effective and lightweight. It can also predict objects with oriented bounding boxes. Moreover, due to the lightweight and simple network design, the designed detector can be easily embedded into edge devices. Extensive experiments demonstrate that the proposed detector is efficient and robust to various complex backgrounds, and the new dataset is more suitable for application scenarios and is quite challenging.**

***Index Terms*—Object detection, remote sensing satellite imagery, robust detection, ship detection, spaceborne ship dataset.**

I. Introduction

OBJECT detection is widely applied in various scenarios and is the core of many vision tasks [1]–[5]. It is used to locate the objects and predict the categories of the objects from the input images or videos, which is a kind of multitask learning algorithm. Spaceborne optical image object detection has many applications, such as military monitoring, fishery management, vessel traffic services, and naval warfare [1], [6]–[9]. Object detection technologies are the core means of remote sensing data interpretation, which have been applied in many aspects of human life in recent years and have been drawing extensive attention.

Ship detection in spaceborne optical images is a challenging task because of the object's huge variation in the scale, orientation, occlusion, and background clusters [1], [5], [6]. Although most of the proposed ship detection algorithms can identify ships well in the ideal background environment, they are often unable to detect ships when there are various jamming in the background environment, such as shadows and clouds. In addition, the extensive computational burden also limits some accurate but complicated object detectors in spaceborne resource-bounded scenarios [11]–[13]. To alleviate these challenges, various effective approaches have been explored, which are mainly reflected in making more practical datasets and designing more accurate and lightweight detection algorithms to meet the requirements of the application platform.

Practical and challenging datasets are very important for ship detection in spaceborne optical images. In recent years, many satellite remote sensing image datasets are collected to meet the training requirements of detection algorithms like *HRRSD* [14] and *HRSC2016* [10]. Although there are a lot of satellite remote sensing ship datasets, the images usually do not reflect the real detection scenes. These images are generally obtained through Google Earth, they have high contrast and clarity. Therefore, it is important to construct a dataset containing images captured under different weathers and environments.

Recently, many object detectors based on deep learning have been applied to remote sensing images [6], [15]–[18]. However, different from most remote sensing images, spaceborne optical image objects are often affected by various jamming and uneven illumination. In practical application, complex background and various jamming seriously restrict the reliability of the algorithm [5]–[7], [19], [20]. Besides, the reliability of most deep learning-based algorithms is strongly dependent on the high-performance computing platform. However, in some mobile application scenarios, only some resource-bounded devices are available. The deep learning-based detectors can only use some restrictive memory access and computation. How to design a ship detector that meets practical requirements is of great significance. In this article, to deal with ship detection in complex jamming scenarios well in spaceborne optical images, a new complex background ship dataset (*CBSD*) and a novel ship detector for spaceborne optical images [antijamming and lightweight ship detector, (*ALSD*)] are proposed.

Specifically, the new dataset *CBSD* is collected from the satellites. It has 4826 ships with different scales, orientations,

Manuscript received March 1, 2022; revised April 26, 2022; accepted May 26, 2022. Date of publication June 2, 2022; date of current version June 10, 2022. *(Corresponding author: Xingxing Wei.)*

Huanqian Yan, Bo Li, and Hong Zhang are with the Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, China (e-mail: yanhq@buaa.edu.cn; boli@buaa.edu.cn; irvJK@buaa.edu.cn).

Xingxing Wei is with the Institute of Artificial Intelligence and Hangzhou Innovation Institute, Beihang University, Beijing 100191, China (e-mail: xxwei@buaa.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2022.3179612

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

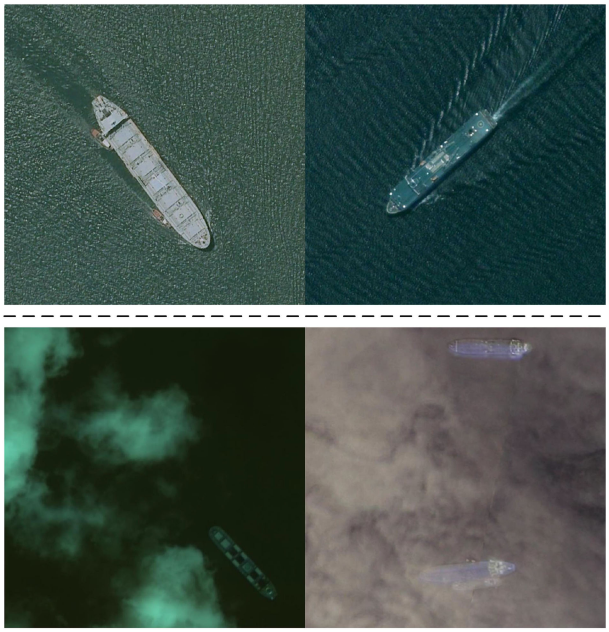


Fig. 1. Some satellite remote sensing images. Above the black dotted line are two images sampled from the public dataset *HRSC2016* [10]. Below the back dotted line are two images sampled from our collected dataset *CBSD*. The ships in these images have different scales and docking directions, but the background in the images sampled from *CBSD* is more complicated. *CBSD* is different from the previous public datasets, it is more realistic and challenging.

and shapes. Due to the complexity of practical application scenes, the collected images deliberately take into account the scenes under heavy cloud, low contrast, dense parked, moving ship wake, and other conditions. Some satellite images are shown in Fig. 1, which are sampled from the dataset *HRSC2016* and our collected dataset *CBSD*, respectively. Compared with *HRSC2016*, the proposed dataset *CBSD* is significantly more realistic and challenging. The novel detection framework mainly includes a feature refining module (FRM) and an antijamming module (AM). The FRM is based on SkyNet [21] and improves the deep features through multireceptive field feature extraction and feature refinement. The AM is a supervised network used to highlight the features of the object area. The proposed detector has a simple network structure, does not need a complex network connection, and has a good detection performance. It can be easily embedded into some edge computing devices such as DSP and FPGA. The main contributions of our work can be summarized as follows.

1) We introduce a new satellite optical ship dataset, which is different from existing public datasets. It is closer to the real application scenarios and includes various types of ship objects.
2) We propose a novel ship detector in spaceborne optical images. It includes a FRM and an AM. It has a high-inference speed and can deal with various complex backgrounds and recognizes objects accurately.
3) The proposed detector is lightweight and efficient, which has a few computations and parameters. It can be embedded into some resource-bounded mobile platforms, such as DSP 6678, without any hardware-specified optimized runtime libraries.
4) Extensive experiments have confirmed the effectiveness of our proposed detector. The results show that compared with the state-of-the-art lightweight detectors, the proposed method achieves better accuracy with less computation. It also demonstrates that the *CBSD* dataset is more applicable to detection scenarios and is quite challenging.

The rest of this article is organized as follows. In Section II, some related work about satellite optical ship datasets and some deep learning-based detectors are introduced. In Section III, a new satellite optical ship dataset is introduced. Section IV describes the details of the proposed detector, which includes a FRM, an AM, a rotated object detection head network, and some loss functions. In Section V, details of implementation and experiments are presented and discussed. Finally, Section VI concludes this article.

II. Related Work

A. Satellite Optical Ship Datasets

A lot of datasets have been released in recent years, which are used to design novel detection algorithms and evaluate the performance of the detectors. Some datasets are collected for multicategory detection that includes ship detection, and some datasets are specially made for ship detection. These datasets are nearly collected from Google Earth. Here, we mainly introduce the following public datasets containing ships.

HRRSD [14] is a dataset collected for alleviating the insufficiency of some publicly available remote sensing datasets. It contains a lot of ships at sea and in ports. The images in this dataset have high spatial resolution (approximately 0.15–1.2 m) and large scale. These images in the dataset are mainly from Google Earth, with a small number from BaiduMap.

DOTA [5] is a dataset with rotated bounding box annotations. It is collected from Google Earth with 15 different object categories. The image sizes of the dataset range from 800*800 to 4000*4000. There are 2806 images including object instances with different orientations and scales. Ship dataset is one of its subsets.

HRSC2016 [10] is a ship dataset collected from Google Earth used for ship detection in satellite remote sensing images. The ships in this dataset have different orientations, scales, and shapes. The number of images and ships is 1070 and 2976, respectively. There is also not a uniform image size for this dataset, and the image sizes range from 300 × 300 to 1500 × 900. 1000 × 600 is the size of most images.

FGSD [22] is a high-resolution remote sensing image dataset, which is collected from many large ports around the world with spatial resolution from 0.12 to 1.93 m. There are a total of 4736 pictures with a unified image size of 930*930. In addition to the horizontal boundary box annotation, the corresponding rotating boundary box annotation is also added to each ship instance.

DIOR [1] is a remote sensing dataset containing 20 categories with 23 463 images. All images have a unified image size of 800*800. The spatial resolutions of this dataset range from 0.5 to 30 m. All object instances in the dataset are manually annotated with horizontal bounding boxes. There are about 1200 images that maybe exist ships. Similar to most of the existing datasets, this dataset is also collected from Google Earth.

However, those remote sensing images above mentioned are different from the images directly acquired from satellites, and they have been processed and enhanced. The images from the satellite are more challenging. The main challenges of the public datasets focus on the difficulty of the recognition of the objects, mainly focusing on multicategory and multiscale. However, the images directly collected from satellites often have more complex background environments, and the difficulty of object detection in such images mainly focuses on the interference of complex backgrounds. Therefore, there still lack the corresponding datasets, which discourages the research of developing some more practical deep object detectors. A new dataset collected from satellites with a spatial resolution of 0.72 m is introduced in this article, which has large scale variation, large appearance variation, and complex backgrounds. The proposed dataset is used to fill this gap and can also be used for the verification and design of lightweight spaceborne object detection algorithms.

B. Deep Learning-Based Detector

Deep learning-based detection algorithms can also be roughly divided into two categories: anchor-based detectors and anchor-free detectors. Anchor boxes can be viewed as predefined sliding windows, which are adopted by many detectors like Faster R-CNN [23], Mask-RCNN [24], YOLO v2 [25], and RetinaNet [26], etc. Usually, anchor boxes are scattered on the feature map, the object regions are predicted by those anchor boxes and an extra offset regression network. Meanwhile, the object category can be also got by another regression network. Different from anchor-based detectors, anchor-free detectors are only based on regression. Through regression networks, locations and categories of objects can be predicted, like CornerNet [27] and CenterNet [28]. Due to CNN's remarkable achievements in natural object detection, researchers introduce it into satellite remote sensing image object detection.

A deep learning-based detector is designed for warship recognition [29]. To extract different sizes of ships, a multilayer feature network is proposed. To balance the different samples, different degrees of data expansion are adopted. The proposed method can recognize up to nine kinds of ships, but it can only give a horizontal bounding box, the detection performance is still not ideal. The horizontal bounding box is widely used in nature image detection, but it often introduces mismatches between the region of interest and objects in the remote sensing images [16]. Therefore, many oriented detectors are proposed to alleviate this problem.

RoI Transformer [16] is one of the most representative oriented detectors. It uses spatial transformation on RoIs to get the oriented bounding boxes of objects, and the transformation parameters are learnt with manually annotated bounding boxes. Similar to ROI transformer, DODet [30] is also designed for evading the problems of spatial and feature misalignments. AR ^{2} Det [31] is a one-stage ship detector, which consists of three submodules, including a feature extraction module, a ship detector, and a center detector. The feature extraction module is used to learn the basic features and enhance the discrimination of the features. A ship detector is developed to decide the position and geometric attributes of ships. The center detector aims to obtain more accurate detection results. Multiscale context and enhanced channel attention are designed for a lightweight oriented object detection algorithm [32]. It can detect some small and oriented objects well. Although the lightweight backbone is used, the proposed method still has a complicated structure, it can only run in real-time with specific high-performance graphics processors. Oriented RCNN [17] is a recently proposed method. Its core idea is to learn the high-quality oriented proposals of the object without any extra network modules. LO-Det [33] is a lightweight oriented detector. It uses channel separation-aggregation structures to simplify the deep model and can produce competitive results. However, it has many parameters, and it can only inference on embedded devices with good computing power. AOPG [34] abandons the horizontal boxes-related operations from the network architecture. It first produces coarse oriented boxes by coarse location module in an anchor-free manner and then refines them into high-quality oriented proposals. Because of the complex network structure, this algorithm cannot be used directly on edge devices.

Although many high-precision detectors have been proposed, they usually rely on high-performance graphics processing and are difficult to be directly deployed to edge devices like satellites. Therefore, an effective and practical ship detector is proposed for spaceborne optical images in this article. It is a one-stage detector without the cumbersome anchor design and can predict objects with oriented bounding boxes. The proposed detector has a lightweight structure and can be easily implemented without any third-party library dependencies. Moreover, our algorithm also can withstand a certain amount of jamming, and achieve high-precision detection in case of jamming.

III. Complex Spaceborne Optical Ship Dataset

Ship detection can be applied in maritime surveillance, traffic supervision, military operations, and other key links, playing an increasingly important role in many tasks. A new ship dataset is proposed in this article. The ship data are mainly collected from the far sea area and the near port area. By taking satellite images from different locations at different times and in different weather, we get satellite images covering large areas. Due to the huge resolution that cannot be dealt with by detectors well, we have cropped each image as small 800*800 pixel images. 800*800 is a common image size in mainstream public datasets. After cropping and selecting, we have collected 1658 high-resolution satellite images including about 4826 ships. Some images from the collected dataset are shown in Figs. 1 and 2.

As mentioned above, all satellite images are taken at different times, which makes the detectors' recognition task more difficult, requiring them to be able to detect ships at day time and night time. For specific types of ships, we collect ships including carriers, destroyers, cargo ships, etc. Unlike some common datasets, we do not give the concrete type of each ship, we classify all ships into the same category "ship." The main

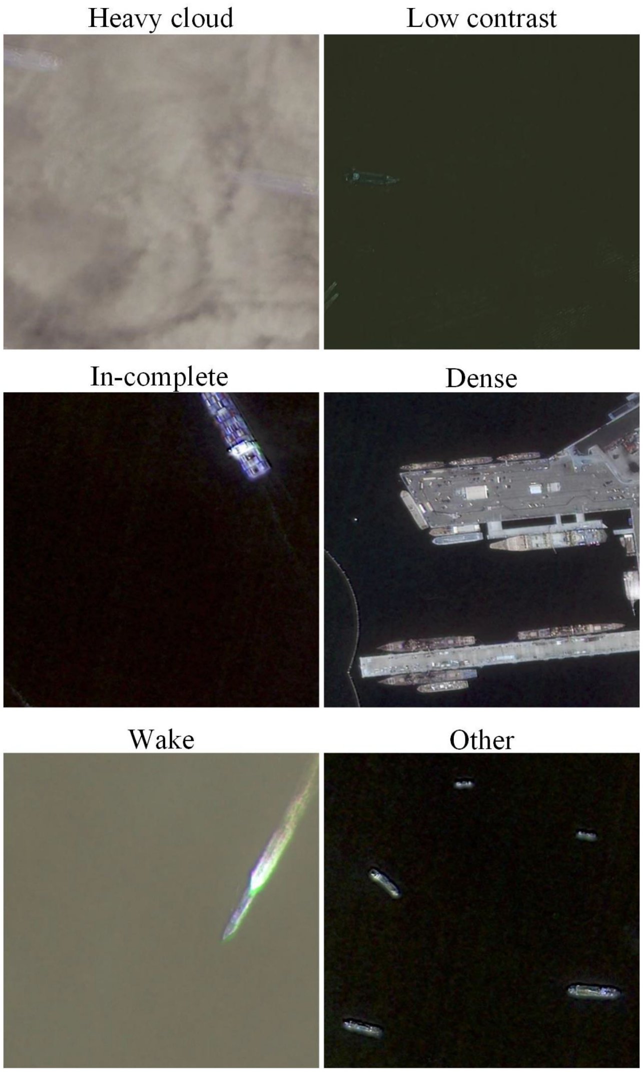


Fig. 2. Some remote sensing satellite images from the collected dataset *CBSD*. Due to different weather, different time periods of day, and different locations, we roughly divide these images into six scenes—heavy cloud scenes, low contrast scenes, incomplete ship scenes, dense scenes, wake scenes, and other scenes.

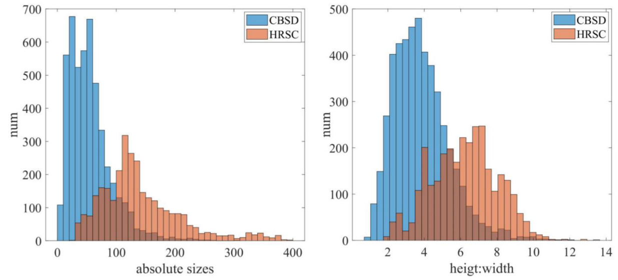


Fig. 3. Some attribute distributions of ships in the dataset *CBSD* and HRSC2016. On the left is the distribution of the aspect ratio of ships. The image on the right shows the scale distribution of ships.

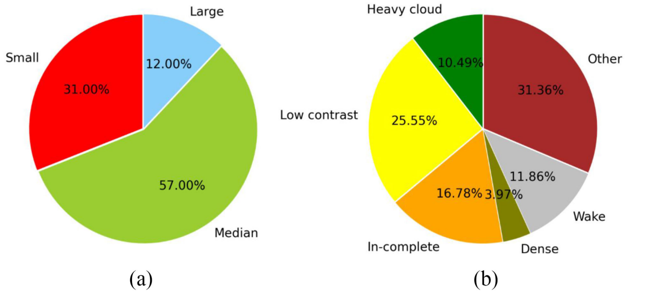


Fig. 4. (a) Ratio of ships with different scales. (b) Ratio of scenes with different complex backgrounds. As shown, the collected dataset *CBSD* has a large proportion of small ships and the images with complex scenes occupy a large proportion of the whole dataset.

reason why we set it is that the amount of ship data of some categories is very small, which is not conducive to the single category learning by deep networks. So the new dataset has a large apparent difference between ships.

The difference in ship appearance will increase the difficulty of detection. Here, we show the apparent difference in data from two aspects: scale variation and ratio variation. We use the absolute size (as_i) to measure the size of the ship i: $as_i = \sqrt{h_i \times w_i}$, where h_i and w_i is the height and width of the ship i. The statistical results of aspect ratio and absolute scale are recorded in Fig. 3. To show the uniqueness of the proposed dataset, the statistical results of the public dataset HRSC2016 are also shown in Fig. 3. Obviously, the proposed dataset has more ships, especially small ships, and the ships have large scale and ratio variations.

According to the COCO dataset setting [35], ships with pixel numbers less than 32*32 are classified as small ships, ships with pixel numbers between 32*32 and 96*96 are classified as medium ships, and ships with pixel numbers greater than 96*96 are regarded as large ships. The statistical results of large, median, and small ships are recoded in Fig. 4(a). As shown, the proportion of small ships is about 1/3. Using the same statistical setting, the proportion of the large ships in the public dataset HRSC2016 is more than 3/4, and there are almost no small ships. It is more challenging for identifying small objects in the collected dataset. The improvement of the detector for the small object is one of the effective means to improve detection performance for the new proposed dataset.

Moreover, to more concretely describe the complexity of the images, we divide the images into different scenes roughly according to the background of the ship. We mainly divide the scene into six aspects: the heavy cloud scenes, the low contrast scenes, the dense scenes, the wake scenes, the incomplete scenes, and other scenes. There are often different cloud interference in heavy cloud scenes. Low contrast scenes are images with dark backgrounds and images with low illuminations. For incomplete ships, it is caused by cropping or satellite imaging limitations. For dense scenes, there are usually densely parked ships in the images. The wake scenes are the scenes that exist moving ships with large tailings. The distribution of different scene ships is shown in Fig. 4(b). Compared with the public datasets like HRSC2016, the image scenes of the proposed dataset are more diverse and complex. Although it may be difficult for one detector to take into account the ship detection in different complex scenes at the same time, it could help design effective detection

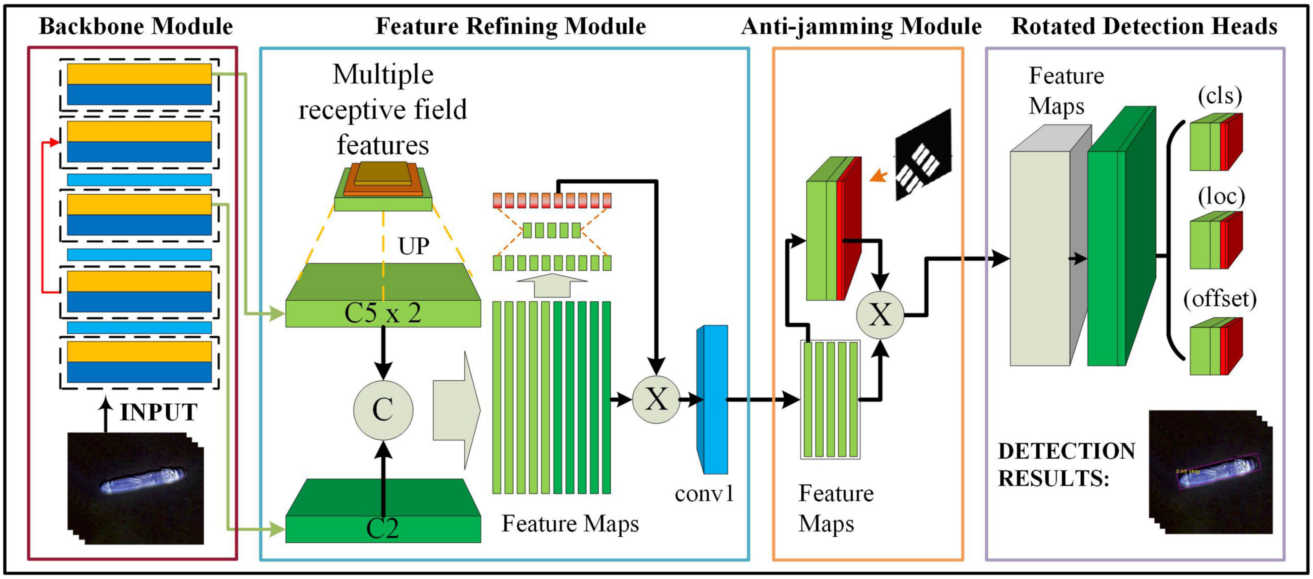


Fig. 5. Proposed ship detection framework. It mainly consists of a feature refining module and an antijamming module. The whole detector is lightweight and effective. It is robust to complex backgrounds and ship appearance variations, and it also can detect ships with oriented bounding boxes.

algorithms to alleviate the influence of complex backgrounds or jamming in practical applications.

The annotation of ships in the collected satellite image dataset is completed by four graduate students with experience in object detection tasks. They follow a uniform ship annotation specification to obtain consistent annotation information. They use the software tool[1] “roLabelImg,” which annotates the objects through θ-based oriented bounding boxes, namely (x_c, y_c, w, h, θ), where θ denotes the angle from the horizontal direction of the standard bounding box. Annotations are saved as XML files, which contain the position information and category information of each ship. It took about five days to annotate the entire dataset. Each image has at least one ship and up to 97 ships. All the ships have different scales and appearances. All the images in *CBSD* have different time periods, different illuminations, and different levels of jamming. It could help to develop object detection algorithms adapted to complex scenarios, which can be immune to jamming but can identify objects well.

IV. Proposed Method

Most state-of-the-art object detectors usually take a big and complex network as the backbone for high detection performance [7], [27], [36]. The mainstream big backbone networks include VGG [37], ResNet [38], GoogleNet [39], DenseNet [40], and so on. However, these backbone networks require huge computational overhead and are not suitable for practical deployment requirements. SkyNet [21] is designed to deliver AI capabilities on some resource-constrained edge devices. In terms of the tradeoff between detection performance and lightweight, it is chosen as the backbone for our detector.

In this section, we mainly introduce the design of the whole detector from four aspects: (FRM, AM, rotated detection heads (RDH), and the corresponding loss functions. The whole detection framework is shown in Fig. 5. In the following sections, we introduce each part step by step.

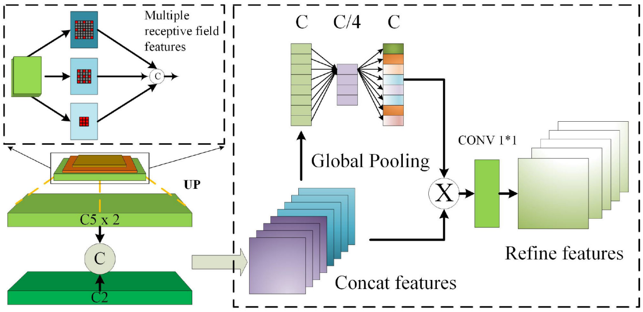
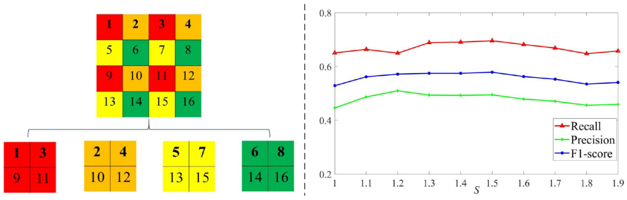


Fig. 6. Feature refining module. It mainly consists of a multireceptive field feature extraction network and a feature refinement network.

A. Feature Refining Module

The whole feature refining module is depicted in Fig. 6. To preserve more low-level features which are beneficial for localization in object detection, the outputs of the third (C2) and sixth (C5) layers of the backbone network are fused. To align these two feature maps, we use the nearest neighbour interpolation algorithm to dilate the size of high-level features as the size of low-level features. Instead of doing feature fusion directly, we convolved the features of the upper level with different dilation rates $r = 0, 1, 2$, and then concatenate these different receptive field feature maps with the low-level feature map. These operations are conducive to obtaining semantic information with different receptive fields, obtaining higher-level semantic

[1] [Online]. Available: https://github.com/cgvict/roLabelImg



(a) (b)

Fig. 7. (a) Feature reordering operation. (b) Detection performance changes with different S value settings.

information, and preserving the superficial feature details [41]. Additionally, we also use feature map bypassing and reordering operations as in the original SkyNet to enhance the ability of small object detection. The bypassing operation is similar to the skip-connection operation, it often crosses a pooling layer, so we use reordering to align the size of the original feature map and bypass one without losing valuable features. The mechanism of reordering operation is shown in Fig. 7(a), which converts a large feature map into several small ones. It is a reorganization operation of the deep features.

Due to the concatenation of low-level features and high-level features in the channel, the additional computational overhead is required if we directly use the fusion features in the next convolution operations. To reduce the computational complexity and make the deep features more effective, a refinement network is designed to reduce the channel of the concatenated features. Motived by the SENet [42], we first take the average pooling of the concatenated features as 1-D vectors. Then, we use a three-layer fully connected network to learn the weights of different channels, and the number of neurons in the fully connected network are 384, 24, and 384, respectively. The predicted weights are used for improving the deep features, and one layer with 1*1 convolution operations is used to reduce the feature dimensions. Compared with direct fusion features, the refined features can reduce the parameter numbers of the whole detector by about 13.9%. After FRM operation, the more efficient deep features can be got, and we use them in the next steps.

More detailed network structure information of backbone and FRM is recorded in Table I. The first six stages are the introduction of the backbone, the last stage is the description of the FRM. The final features extracted by the proposed FRM are recorded in the last line. The parameter numbers for each stage are counted in the last column. As shown, the whole FRM is lightweight. Additionally, the visualizations of deep features with refining module and without refining module are shown in Fig. 8 using EigenCAM algorithm [43]. Obviously, the features with the refining operation are more effective.

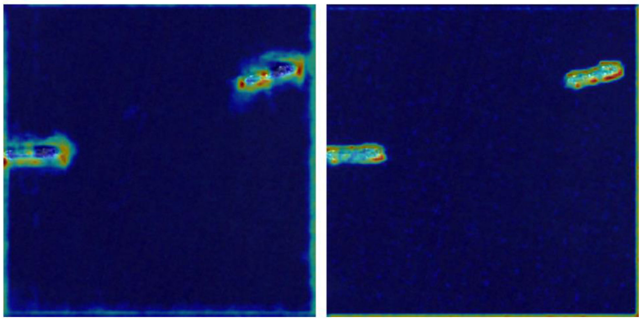
B. Antijamming Module

Remote sensing satellite images often face various forms of jamming in the object regions due to clouds, dust, weather, and other unknown reasons. Different jamming will cause occlusion to the object and the jamming with similar geometric shapes will induce the neural network to produce wrong recognition. Enhancing object cues and weakening nonobject information is an effective solution for mitigating the effects of jamming. The AM is designed for enhancing the objectness of objects under complex backgrounds. It can predict different weights for deep features. Usually, the features of the object region will have big weights and the features in other regions are going to have smaller weights.

TABLE I
BACKBONE NETWORK AND FEATURE REFINING MODULE CONSIST OF SEVEN STAGES

Stage		**Layer**	**Output size**	**Params**
	Input		160*128*3	
Stage 0	DW-Conv3	3*3, stride=1	160*128*3	0.273K
	PW-Conv1	1*1, stride=1	160*128*48	
	Max-Pooling	2*2, stride=2	80*64*48	
Stage 1	DW-Conv3	3*3, stride=1	80*64*48	5.328K
	PW-Conv1	1*1, stride=1	80*64*96	
	Max-Pooling	2*2, stride=2	40*32*96	
Stage 2 (C2)	DW-Conv3	3*3, stride=1	40*32*96	19.872K
	PW-Conv1	1*1, stride=1	40*32*192	
	FM Reordering		20*16*768	
	Max-Pooling	2*2, stride=2	20*16*192	
Stage 3	DW-Conv3	3*3, stride=1	20*16*192	76.608K
	PW-Conv1	1*1, stride=1	20*16*384	
Stage 4	DW-Conv3	3*3, stride=1	20*16*384	201.856K
	PW-Conv1	1*1, stride=1	20*16*512	
Stage 5 (C5)	FM Concatenate		20*16*(512+768)	407.808K
	DW-Conv3	3*3, stride=1	20*16*(512+768)	
	PW-Conv1	1*1, stride=1	20*16*256	
	Conv1	1*1, stride=1	20*16*256	
Stage 6	Dilation-Conv3	3*3, r=0,1,2	20*16*(64+64+64)	615.936K
	Nearest neighbor interpolation		40*32*192	
	Feature Fusion		40*32*(192+192)	
	Feature Refine		40*32*384	
	Conv1	1*1, stride=1	40*32*192	

DW-Conv3 is the 3×3 depth-wise convolutional layer. PW-Conv1 is the 1×1 point-wise convolutional layer. Here, an input scale of 160×128 is taken as an example to demonstrate the process of deep feature extraction.



(a) (b)

Fig. 8. Visualization of the deep features without the refining module (a) and with the refining module (b).

The AM is a four-layer convolution network without pooling operations, which is lightweight and flexible. As shown in Fig. 9, it consists of two depth separable convolution layers and two standard convolution layers. The channels of its four layers are 192, 128, 64, and 1, respectively. The parameter numbers of the AM are about 118.8 K. Due to its lightweight structure, it does not add too much additional computational overhead.

In the training phase, we use the ground-truth information to produce the corresponding mask for guiding the AM to update its parameters. The mask is a binary graph. The mask

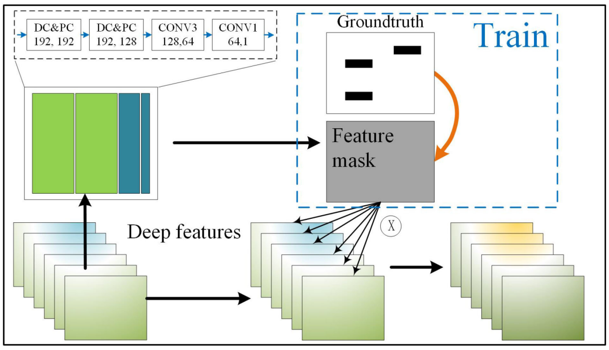


Fig. 9. Proposed AM. The input of the network are $h * w * c$ feature maps, and the output is a $h * w$ weight map. The weight maps are supervised by a ground-truth mask during training to learn the importance of different areas.

position corresponding to the object area is set to 1, and the value of the nonobject area is set to 0. To preserve more context information, we have expanded the object area by a factor of S. In experiments, different S values are tried with the input size 416*416, the detection results are shown in Fig. 7(b). It is found that $S = 1.5$ is a good setting. Therefore, the hyperparameter S is set to 1.5 in our proposed method.

The AM network predicts a mask map of the objects $RM \in R^{1\times H/s\times W/s}$, where W, H is the width and height of the input remote image. s is the stride of the deep model, it is four in the proposed detector. The channel of the prediction map is one because each image only needs one mask map to indicate the position information of the objects in the input image. The sigmoid function is used to scale the predicted value to the interval [0,1] at the end of the AM. For the loss function used for updating the antijamming network, we do not use a complex loss function, but a simple binary cross-entropy loss function

$$loss_{mask} = \frac{1}{N}\sum_{k=i}^{N} -[\widehat{p_i} \times log(p_i) + (1-\widehat{p_i}) \times log(1-p_i)] \tag{1}$$

where $\widehat{p_i}$ is the ground-truth pixel value, p_i is the prediction pixel value.

After training, the AM can predict the corresponding feature weights. The weights mean the probability that the current pixels have objects. So the deep features multiplying with the weights can enhance the object cues and weaken the nonobject information.

C. Rotated Detection Heads

As aforementioned, objects in remote sensing satellite images are usually rotated. Using horizontal boundary boxes to predict arbitrary direction objects and densely parked objects will lead to the deviation or mismatch between the predicted object position and the corresponding ground truth position. Compared with the horizontal boundary boxes, the rotated boundary boxes are more suitable for satellite remote sensing images. There are many rotated object detection algorithms, which usually regress an additional angle parameter to locate objects precisely [16]. Obviously, angle regression is sensitive. For large objects, a small angle deviation will result in a big positioning error. Moreover, the angle regression is separated from the regression of other attributes of the object, which is not conducive to learning and convergence of the network.

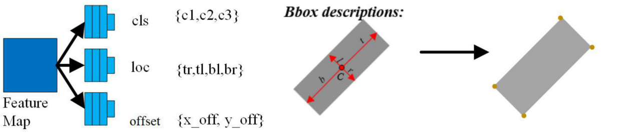


Fig. 10. Rotated detection head network. It has three different mission branches: *cls* head, *loc* head, and *offset* head.

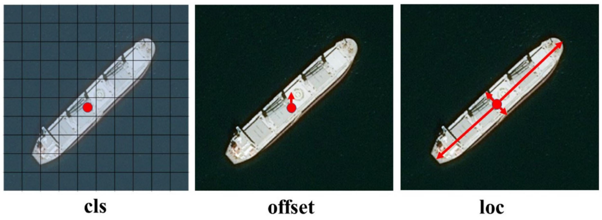


Fig. 11. Schematic diagram of the working principle of each detection head network.

In this article, we predict the oriented objects with key points. We do not regress the w, h, and angle θ at each feature pixel, rather than regress five points at each feature pixel. The center point and the position of the four vertices will locate the objects with oriented bounding boxes. The principle of position regression is shown in Fig. 10. It is simple and effective. For the detection heads, there are three branches: *cls* head, *loc* head, and *offset* head. The *cls* head is used to predict the categories of the objects. It has C channels, where C is the class number. The *loc* head is used to predict the positions of the four points. The *offset* head is used to compensate for the difference between the quantified floating center point and the integer center point. The total parameter numbers of these three detection head branches are about 1.25 M. Through these three heads, we can get accurate detection results. The working schematic diagram of different heads is shown in Fig. 11.

D. Loss Functions

Similar to most object detection frameworks, we use different convolution features to predict the categories of different objects. Through the recognition of object center points of different categories, it can be determined whether there are objects of the category in the input image. On the one hand, the center points can be used for predicting the categories of objects, and on the other hand, it is coordinated with the location prediction network module *loc*, which is more helpful to the learning and convergence of the detector. Specifically, the *cls* head outputs the the feature maps $OM \in R^{C\times H/s\times W/s}$ with C categories. The map values are regarded as the confidence of the objects. To make the maps smooth, each channel feature is normalized with a sigmoid function before prediction. During training, only the center points c are positive, and the other points are negative. To mitigate this balance issue, we use a variant focal loss [27] to train the *cls* head

$$loss_{cls} = -\frac{1}{N}\sum_{i}\begin{cases}(1-p_i)^{\alpha}log(p_i) & if \quad \widehat{p_i}=1\\ (1-\widehat{p_i})^{\beta}p_i^{\alpha}log(1-p_i) & otherwise\end{cases} \tag{2}$$

where $\widehat{p}$ and p refer to the ground-truth and the predicted heatmap values, i indexes the pixel locations on the feature map, N is the number of objects, α and β are the hyperparameters used for balancing the contribution of each feature pixel. Similar to [27], we set $\alpha = 2$ and $\beta = 4$ in our experiments.

Because the object position information will cause floating point to integer loss bias in the process of features mapping at different scales, detection algorithms usually learn this bias through an extra network module and then compensates for it in the inference phase. The *offset* head is used to learn this bias $FM \in R^{2\times H/s\times W/s}$. The *offset* between the scaled floating center point and quantified center point can be defined as

$$o = \left(\frac{\overline{c_x}}{s} - \left\lfloor\frac{\overline{c_x}}{s}\right\rfloor, \frac{\overline{c_y}}{s} - \left\lfloor\frac{\overline{c_y}}{s}\right\rfloor\right) \tag{3}$$

where $\overline{c} = (\overline{c_x}, \overline{c_y})$ is the ground-truth center point. The *offset* can be optimized with a smooth L_1 loss

$$loss_{offset} = \frac{1}{N}\sum_{i=1}^{N} Smooth_{L1}(o_i - \widehat{o_i}) \tag{4}$$

where $\widehat{o}$ refers to the ground-truth offset. The smooth L_1 loss can be expressed as

$$Smooth_{L1}(x) = \begin{cases}0.5\times x^2 & if|x|<1\\ |x|-0.5 & otherwise.\end{cases} \tag{5}$$

As aforementioned, we use the keypoint regression to predict object location information. Based on the position of the keypoints, we could calculate the bounding boxes through the minimum circumscribed rectangle algorithm. To increase the cooperation among keypoints and speed up the convergence rate of the detection head networks, we do not directly regress the four corners of the objects, but indirectly calculate the coordinates of the four corners by regression to the deviations of the center point of the four sides relative to the center points of the object. The regression corner learning is expressed on the right side of Fig. 10. This operation can increase the relationship between boundary points and central points, promote the accuracy of key points learning, and alleviate the problem of poor locating accuracy caused by weak features. The *loc* head will predict a box parameter map $LM \in R^{8\times H/s\times W/s}$. The reason there are eight channels is that there are 2×4 vectors. Smooth $L1$ loss is used to guide the regression

$$loss_{loc} = \frac{1}{N}\sum_{k=i}^{N} Smooth_{L1}(b_i - \widehat{b_i}) \tag{6}$$

where b and $\widehat{b}$ are the predicted and ground-truth box parameters, respectively.

Normalizing loss: Although $loss_{loc}$ can guide the detector to predict some objects accurately, it can not deal with some small objects well. Because the position deviation of small objects is easy to be ignored by the $loss_{loc}$ function, it can not have a positive effect on the detector. Small object imprecise location information could make the final detection results have small Precision values and small AP values. Here, we introduce another location loss function $loss_{norm}$ to train the detector except from the loss $loss_{loc}$. The corresponding quintuple representation of the object position (c_x, c_y, w, h, θ) is computed by transforming the coordinates of the prediction bounding boxes, so we can get the horizontal regression box B_i of the objects. The ground-truth horizontal bounding boxes $\widehat{B_i}$ can be acquired by minimum circumscribed rectangle algorithms. Finally, the $loss_{norm}$ loss function is defined as

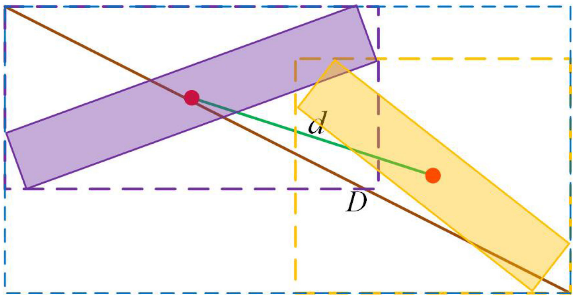


Fig. 12. Normalizing loss $loss_{norm}$ of two oriented regression boxes. It first calculates the distance loss of the corresponding horizontal regression boxes, then calculates the rotation angle bias loss, and evaluates the difference between the boxes by the product of the two kinds of losses.

$$Loss_{norm} = \frac{1}{N}\sum_{k=i}^{N}\left(\frac{d^2(B_i, \widehat{B_i})}{D^2}\right) \times \|\frac{\theta_i - \widehat{\theta_i}}{\pi}\| \tag{7}$$

where D^2 means the diagonal distance of the minimum circumscribed rectangle of two bounding boxes, d^2 means the distance of two center points of two horizontal bounding boxes, $\widehat{\theta_i}$ is the ground-truth angle of the object. The normalizing loss $loss_{norm}$ of two oriented regression boxes is described in Fig. 12. It does not ignore the deviation of some small objects, which is good for locating small objects.

Therefore, the whole loss function of the proposed detector is rough as follows:

$$\begin{aligned}Loss = \alpha \times loss_{cls} + \beta \times loss_{loc} + \gamma \times loss_{norm}\\ + \rho \times loss_{mask} + \kappa \times loss_{offset}\end{aligned} \tag{8}$$

where α, β, γ, ρ, and κ are the balance coefficients. As a rule of thumb, they are set to 1, 0.3, 0.7, 0.1, 1 in our experiments.

V. Experiments and Analysis

The new proposed dataset, *CBSD*, is used to quantitatively evaluate the performances of the proposed method in the experiments. To evaluate the effectiveness of the proposed method, it is compared with five mainstream methods: BBVector [7], RIDet [44], Oriented FCOS [45], RoI Transformer [16], and Oriented RCNN [17]. The evaluation metrics precision, recall, F1-score, and AP are used to measure the detection algorithms in the experiments.

A. *Competitive Algorithms*

For BBVector, it detects the center keypoints of the objects and regresses the box boundary aware vectors to capture the oriented bounding boxes. The framework is simple but efficient. For RIDet, it uses cascaded RetinaNet to achieve oriented object detection and designs a representation invariance loss to optimize the bounding box regression. For Oriented FCOS, it is an extensible version of the original FCOS method. FCOS is a single-stage object detection algorithm based on full convolution. It acquires the object position and category information by predicting each feature pixel, similar to semantic segmentation. In the experiments, in order to predict direction, an extra channel convolution layer is used to regress the direction angle θ. Following original FCOS, PolyIoULoss [46], [47] is used for training. As for RoI transformer, it learns spatial transformations on RoIs and uses these information to change the horizontal RoIs to oriented RoIs. For oriented RCNN, it proposes an oriented region proposal network. Since the prediction of rotation direction does not involve the additional network modules, it does not need additional computation.

To meet the requirements of practical scenarios, many researchers have been conducted to design lightweight object detectors. Replacing the initial complex backbone network of the detectors with a lightweight backbone network is a common way to make it lightweight, which is widely used and adopted. The focus of the proposed method is practical application, so the selected competitive algorithms need to have fewer parameters and computation. To make fair comparisons, we replace all backbone networks of comparison algorithms with MobileNet V2 network [13], SqueezeNet network [48], and ShuffleNet network [12]. MobileNet V2 uses depthwise separable convolutions to build lightweight deep neural networks. ShuffleNet utilizes two operations, pointwise group convolution and channel shuffle, to reduce computation costs. SqueezeNet uses squeeze convolution operations and expands layers to reduce parameters while maintaining competitive accuracy. Those models are well-known lightweight neural networks, widely used in object detection algorithms and adopted by various detection algorithm platforms.

In order to minimize the impact on the detection performance of the original detection algorithms, we retain the FPN operations [49] of all comparison detectors, and increased the number of iterations of each algorithm to 60 epochs. For MobileNet V2, 2nd, 4th, and 6th feature maps are selected. For ShuffleNet and SqueezeNet, 2nd and 4th feature maps are used for detection.

B. *Evaluation Metrics*

For each ship predicted by the detection algorithm, if the overlap rate between the area of the predicted ship position and the ground truth area is greater than 50%, it is considered that the prediction result is correct, otherwise, it is wrong. For evaluation metrics, recall, precision, F1-score, and average precision (AP) are used to evaluate detection results. Recall measures how many positive samples in the total sample are predicted to be correct. Precision measures how many of the predicted positive samples are positive. Since recall and precision rates have different evaluation concerns on detection results, F1-score evaluates detection performance by combining recall and precision, $F1 = (2 \times precision \times recall)/(precision + recall)$. AP is also an evaluation index that takes into account both recall and precision rates. Different from $F1$-score, AP value is a comprehensive evaluation index of precision under different recall values, similar to the area under the *precision-recall curve*. For all evaluation indices, the higher the values, the better the performance, and vice versa.

TABLE II
ABLATION STUDY ON NEW COLLECTED DATASET *CBSD*

Methods	R	P	F1	AP
RDH+SkyNet (Baseline)	0.555	0.477	0.513	0.481
Baseline+FRM	0.626	0.445	0.520	0.542
Baseline+FRM+AM	**0.698**	0.481	0.570	0.583
Baseline+FRM+AM+loss$_{norm}$(ALSD)	0.696	**0.495**	**0.579**	**0.588**

Recall (R) value, precision (P) value, F1-score (F1), and AP value are used to evaluate the detection performance. A larger metric value means a better detection result. The bold entities represent the best performances.

C. *Implementation Details*

The dataset *CBSD* is randomly divided into 60% and 40% for training and testing, respectively. The same partitions are used for evaluating all algorithms in the experiments. We implement the proposed method on Pytorch 1.1.0. All algorithms in experiments use two NVIDIA TITAN V GPUs for training and testing. Because the training set is a bit small, we only train the proposed model for 60 epochs in the training phase. We use Adam optimizer [50] for training and 6e-4 as the initial learning rate and decay it by a factor of 0.5 at 25 and 50 epochs. The input size of the proposed model is set to 600*600 pixels, which leads to the batch size being only 4. For the comparison algorithms, we use the same settings as the proposed method.

D. *Ablation Studies*

We analyze the importance of each proposed component on our collected dataset *CBSD*. The impact of each component is listed in the Table II. In this section experiment, we do not use the large input size, but a smaller one, 416*416. It is well known that smaller input size tends to have faster training speed and inference speed, while larger input size tends to have higher performance indicators.

1) Effect of the Feature Refining Module: First, we use the proposed oriented heads and the original SkyNet to build a baseline method, which is named "RDH+SkyNet." There are not any improvement operations in this new detector, just a simple combination. Second, we add the FRM to the baseline method. The new combination is named "Baseline+FRM." The experimental results clearly show the advantages of the feature extraction refining module. The main improvement focuses on recall value and AP value. The "Baseline+FRM" are about 12.8% and 12.7% ahead of the baseline method "RDH+SkyNet" in recall and AP, respectively. This apparent advantage mainly comes from the improvement of the backbone network. The FRM can obtain more effective features and directly promote the improvement of network detection performance.

2) Effect of the Antijamming Module: There are some obvious jamming scenes and unclear object areas in the new dataset *CBSD*. The proposed AM is evaluated here. It refines the validity of the depth feature by suppressing the feature of the jamming region and highlighting the object region. Here, the AM is added to the method "Baseline+FRM," and the new method is named "Baseline+FRM+AM." As observed from the table, the module improves the final detection performance significantly. For recall value, the new method has been improved about 11.5%. It means that the problem of missing detection has been mitigated. For Precision value, the new method is also ahead about 8% than baseline. This means that detection results are more reliable. Additionally, the new method is ahead 9.6% and 7.6% of the method "Baseline+FRM" in indices F1-score and AP, respectively. Those significant improvements are consistent with the theoretical analysis. It can conclude that this module can effectively suppress the effects of jamming and reduce false detection and missing detection.

3) Effect of the Normalizing Loss: The proposed rotated distance loss function make some normalization for position information of the predicted object relative to the corresponding ground truth information, theoretically eliminating the criticism that the loss values are different due to the different object scales. Here, the normalizing loss is added to guide the training. This new method is named "Baseline+FRM+AM+loss$_{norm}$," and it also named *ALSD*. Through comparison, we can find that normalizing loss can slightly improve detection performance, mainly focusing on the improvement of precision values and F1-score values, because more accurate location information can effectively alleviate the problem of missing detection and wrong detection. The tighter the bounding boxes, the larger the overlap with the ground truth annotated rotated boxes, and the easier it is to be identified as a ship. The new loss function can improve the detection algorithm about 2.9%, 1.6%, and 0.9% in precision, F1-score, and AP, respectively.

TABLE III
DETECTION RESULTS ON THE DATASETS CBSD WITH SIX DETECTION METHODS BASED ON DIFFERENT LIGHTWEIGHT BACKBONE NETWORKS

Methods	Recall	Precision	F1-score	AP
RIDet$_{mbl}$	0.496	0.487	0.491	0.430
BBVector$_{mbl}$	0.395	0.319	0.353	0.298
Oriented FCOS$_{mbl}$	0.631	0.370	0.466	0.569
RoI Transformer$_{mbl}$	0.654	0.533	0.587	0.605
Oriented RCNN$_{mbl}$	0.632	0.551	0.589	0.603
RIDet$_{sh}$	0.651	0.554	0.598	0.610
BBVector$_{sh}$	0.328	0.318	0.323	0.280
Oriented FCOS$_{sh}$	0.480	0.554	0.514	0.414
RoI Transformer$_{sh}$	0.511	0.654	0.574	0.510
Oriented RCNN$_{sh}$	0.361	0.331	0.345	0.343
RIDet$_{sq}$	0.468	0.481	0.474	0.418
BBVector$_{sq}$	0.519	0.276	0.360	0.399
Oriented FCOS$_{sq}$	0.482	**0.768**	0.593	0.442
RoI Transformer$_{sq}$	0.657	0.593	0.623	0.611
Oriented RCNN$_{sq}$	0.591	0.543	0.566	0.523
ALSD (ours)	**0.705**	0.588	0.641	**0.636**
ALSD$_{sh}$ (ours)	0.511	0.676	0.582	0.511
ALSD$_{sq}$ (ours)	0.691	0.626	0.657	0.601
ALSD$_{mbl}$ (ours)	0.669	0.642	**0.655**	0.604

The bold entities represent the best performances.

E. Comparisons With the State-of-The-Art Methods

The detection results of the proposed *ALSD* and the other fifteen lightweight detectors are recorded in Table III. Those comparison algorithms include three single-stage detectors (BBVector, RIDet, oriented FCOS) and two double-stage detectors (ROI transformer and oriented RCNN). Note that all comparison algorithms are based on three different backbone networks. The best results under different evaluation indices are marked in bold. From a comprehensive point of view, the proposed *ALSD* has the dominant advantage in all evaluation indices. The proposed algorithm has an overwhelming advantage in recall value, which benefits from effective feature extraction and feature enhancement of object region. Some detection examples of the proposed method and two double-stage detectors RoI Transformer and Oriented RCNN are shown in Fig. 13. The reason the results of two double-stage detectors are chosen for the result presentation is that they have better detection performance in comparison algorithms. For those shown images, they are all sampled from the *CBSD* testing dataset, and the images are either low illumination or clouded. As shown, the proposed *ALSD* still deals well with ship detection in various scenarios, but the other two comparison algorithms have some false positives and false alarms. In the next paragraphs, we compare and analyze the proposed method with other competitive algorithms in detail.

Comparing the proposed *ALSD* with three single-stage detectors based on three different lightweight backbone networks, the proposed method has obvious advantages. Among those three comparison single detectors with different backbone settings, the best detection results are produced by RIDet with ShuffleNet (RIDet$_{sh}$). Compared to RIDet$_{sh}$, *ALSD* is 8.3%, 6.1%, 7.2%, 4.3% ahead of it in recall, precision, F1-score, and AP, respectively. In a series of comparison algorithms, oriented FCOS based on MobileNet V2 (oriented FCOS$_{mbl}$) is second only to RIDet$_{sh}$. Compared to oriented FCOS$_{mbl}$, *ALSD* is 11.7%, 58.9%, 37.6%, 11.8% ahead of it in recall, precision, F1-score, and AP, respectively. With the SqueezeNet setting, oriented FCOS (oriented FCOS$_{sq}$) has given a better precision value than the proposed *ALSD*, but oriented FCOS$_{sq}$ has a terrible result in AP value. For oriented FCOS, it regresses object attributes based on full convolution. The biggest difference between it and the proposed *ALSD* lies in the rotation angle prediction. It uses an extra network branch to predict the rotation angle value. Although it is simple, the angle value is sensitive, which limits the detection performance.

Comparing the proposed *ALSD* with two double-stage detectors based on three different lightweight backbone networks, three algorithms produce better detection results in a series of comparison detectors, which are RoI transformer based on MobileNet V2 (RoI transformer$_{mbl}$), RoI transformer based on SqueezeNet (RoI transformer$_{sq}$), oriented RCNN based on MobileNet V2 (oriented RCNN$_{mbl}$). The proposed *ALSD* is 7.8%, 10.3%, 9.2%, 5.1% ahead of RoI transformer$_{mbl}$ in recall, precision, F1-score, and AP, respectively. It also is 11.6%, 6.7%, 8.8%, 5.5% ahead of oriented RCNN$_{mbl}$ in recall, precision, F1-score, and AP. For RoI transformer$_{sq}$, it has a better precision

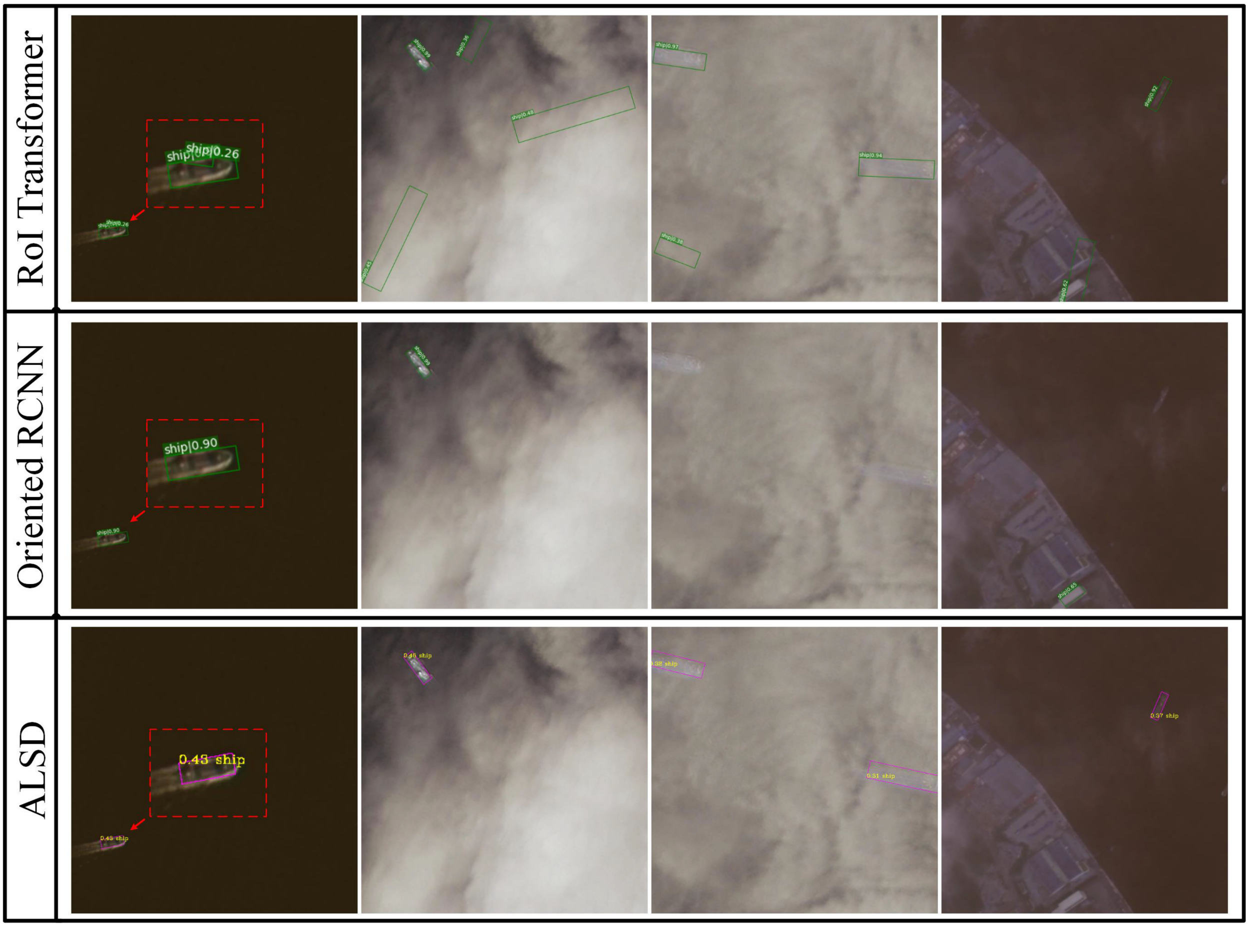


Fig. 13. Visual comparisons of RoI transformer$_{mbl}$, oriented RCNN$_{mbl}$, and *ALSD*. All images are from CBSD's testing dataset. The score threshold of these three methods for visualization is set to 0.2. Please enlarge the picture for showing it clearly.

index than *ALSD*, but the advantage is small. Except for the precision index, the proposed *ALSD* leads in indices recall, F1-score, and AP. With the lightweight setting, two double-stage methods are significantly superior to oriented FCOS and BBVector in all evaluation metrics in the experiments. Generally, double-stage detectors are better than single-stage detectors in detection performance, but the advantages do not seem to exist when compared with the proposed method. Two double-stage detectors have some missing and wrong detections. We think there are two possible reasons behind these poor detection results. One is that the feature extraction ability of the lightweight backbone network is limited, which is difficult to meet the double-stage tasks at the same time. The other one is that there are many challenges in testing data, such as large object variants and complex backgrounds.

In order to further demonstrate the superiority of the proposed detector, its variant methods "ALSD$_{mbl}$," "ALSD$_{sq}$," and "ALSD$_{sh}$," are also added in the comparative experiments. The variant methods are made by replacing SkyNet with MobileNet V2, SqueezeNet, and ShuffleNet. With the MobileNet V2 network, ALSD$_{mbl}$ has significant advantages than any comparison detectors based on the MobileNet V2 network. When all detectors use the SqueezeNet network as the backbone network, the proposed ALSD$_{sq}$ is excellent in general. More concretely, oriented FCOS$_{sq}$ has the best precision value, and RoI transformer$_{sq}$ has the best AP value. The proposed ALSD$_{sq}$ has the best recall and F1-score values. With the ShuffleNet backbone, RIDet$_{sh}$ have big advantages in recall and AP values. The proposed ALSD$_{sh}$ has the best precision value and has a similar performance to RoI transformer$_{sh}$.

Through a series of experiments and comparisons, we can conclude that the new dataset *CBSD* collected by us is somewhat challenging. It is different from other common data, and it is closer to the practical scenarios. The complex satellite images can easily cause deep detectors to miss and misidentify objects. Additionally, we can also find that the proposed *ALSD* has better detection performance and can predict a series of competitive detection results compared with the mainstream detection algorithms.

F. Detection on Dataset HRSC2016

The HRSC2016 is a widely used ship dataset including about 1070 images with the ship in various appearances. Here, it is used to show the effectiveness of the proposed method. 626 images are used as the training set, the rest 444 images are used as the testing set. Here, we select two double-stage detectors as comparison algorithms. The performance comparison results

TABLE IV
DETECTION RESULTS ON THE DATASETS HRSC2016

Method	Recall	Precision	F1-score	AP
RoI Transformer$_{sq}$	0.820	0.645	0.722	**0.779**
RoI Transformer$_{sh}$	0.559	0.518	0.538	0.497
RoI Transformer$_{mbl}$	0.588	0.374	0.457	0.488
Oriented RCNN$_{sq}$	0.804	0.565	0.664	0.763
Oriented RCNN$_{sh}$	0.609	0.544	0.575	0.572
Oriented RCNN$_{mbl}$	0.807	0.568	0.667	0.764
ALSD (ours)	0.817	0.618	0.703	0.718
ALSD$_{sq}$ (ours)	0.809	0.682	0.740	0.740
ALSD$_{sh}$ (ours)	0.822	**0.779**	**0.800**	0.768
ALSD$_{mbl}$ (ours)	**0.826**	0.711	0.764	0.761

The bold entities represent the best performances.

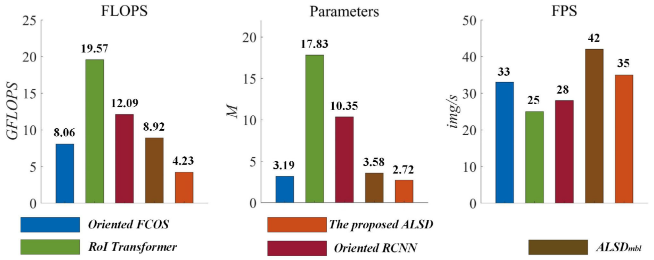


Fig. 14. Computational efficiency analysis of four detectors, oriented FCOS, RoI transformer, oriented RCNN, and *ALSD*. The input size of these algorithms are all 600*600 pixels.

between the proposed *ALSD* and comparison algorithms RoI transformer and oriented RCNN on HRSC2016 are shown in Table IV. To demonstrate the effectiveness of the algorithms, we combine each detector with different backbone networks. Through comparison, we can conclude that the detection performance of different algorithms has obvious fluctuation under different backbone settings. With the SqueezeNet setting, RoI transformer has the best results in indices recall and AP, but the proposed $ALSD_{sq}$ has better precision and F1-score values. Under the ShuffleNet setting, the proposed method has the best results in all indices. For the Mobilenet V2 setting, the proposed method also has the best results. Besides, we also record the *ALSD* with the SkyNet in the Table IV. Although *ALSD* with SkyNet does not produce the best detection results, it is still better than most competitive algorithms. In general, it can be concluded that the proposed *ALSD* can produce competitive detection results and is comparable to the mainstream detection algorithms.

G. Computational Efficiency Analysis

The execution efficiency of the algorithms is compared and analyzed in this section. We mainly record parameter numbers, FLOPs, and the inference speed of each algorithm. Here, all comparison algorithms are built on MobileNet V2. The corresponding results are shown in Fig. 14. The specific numerical information is recorded at the top of each bar chart. For inference speed, the proposed methods have some advantages, they have over 30 FPS with the input size 600*600. In general, the smaller the FLOPs value, the faster the inference speed. The proposed *ALSD* has 4.23 Gflops, but oriented FCOS, RoI transformer, and oriented RCNN have 8.06, 19.57, 12.09 Gflops. The proposed *ALSD* with MobileNet V2 has 8.92 Gflops, but it is still lighter than RoI transformer and oriented RCNN. Besides, $ALSD_{mbl}$ has the fastest inference speed and is ahead of *ALSD*, which may be due to MobileNet's unique optimization for the hardware platform.

Storage space is one of the factors restricting the successful deployment of the detection algorithms. For parameter numbers, parameter numbers of the proposed *ALSD* are about 85.3%, 15.3%, and 26.3% of the method oriented FCOS, ROI transformer, and oriented RCNN, respectively. The lower the number of parameters, the lower the storage requirements. The proposed *ALSD* with MobileNet V2 is also ahead of RoI transformer and oriented RCNN about 79.9% and 65.4%, respectively.

In addition, due to the simple architecture of the proposed detector, it can be easily implemented without any third-party library support. We have embedded it into the satellite with the DSP 6678 platform, and it can predict at the desired speed. To sum up, we can conclude that the proposed method is lightweight and practical.

VI. CONCLUSION

In this article, we have provided a new dataset for remote sensing satellite ship detection, which is practical and challenging. Additionally, we have proposed a novel object detector for remote sensing satellite images. An FRM has been designed to extract effective deep features, which can improve detection performance significantly. To deal with various jamming and complex backgrounds, a lightweight network module based on supervised learning was designed for highlighting the features of objects in the whole feature map and suppressing other region feature information. The proposed detection algorithm can deal with the ships in various scales and orientations well and is also robust to complex backgrounds. The extensive experiments have confirmed the validity of the proposed method. The proposed detection framework is different from other mainstream detection networks, it was lightweight and efficient. Finally, we hope that our newly introduced dataset *CBSD* will provide opportunities for researchers to develop novel and lightweight detection algorithms for remote sensing satellite images.

REFERENCES

[1] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object detection in optical remote sensing images: A survey and a new benchmark," *ISPRS J. Photogrammetry Remote Sens.*, vol. 159, pp. 296–307, 2020.

[2] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[3] P. Sun, G. Chen, and Y. Shang, "Adaptive saliency biased loss for object detection in aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 10, pp. 7154–7165, Oct. 2020.

[4] G. Cheng, Y. Si, H. Hong, X. Yao, and L. Guo, "Cross-scale feature fusion for object detection in optical remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 3, pp. 431–435, Mar. 2021.

[5] G.-S. Xia *et al.*, "Dota: A large-scale dataset for object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3974–3983.

[6] X. Yang *et al.*, "SCRDet: Towards more robust detection for small, cluttered and rotated objects," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8231–8240.

[7] J. Yi, P. Wu, B. Liu, Q. Huang, H. Qu, and D. Metaxas, "Oriented object detection in aerial images with box boundary-aware vectors," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2021, pp. 2149–2158.

[8] C. Xu, C. Li, Z. Cui, T. Zhang, and J. Yang, "Hierarchical semantic propagation for object detection in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 4353–4364, Jun. 2020.

[9] H. Yan, "Detection with fast feature pyramids and lightweight convolutional neural network: A practical aircraft detector for optical remote images," *J. Appl. Remote Sens.*, vol. 16, no. 2, 2022, Art. no. 024506.

[10] Z. Liu, H. Wang, L. Weng, and Y. Yang, "Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 8, pp. 1074–1078, Aug. 2016.

[11] A. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[12] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6848–6856.

[13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[14] Y. Zhang, Y. Yuan, Y. Feng, and X. Lu, "Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5535–5548, Aug. 2019.

[15] M. Zhou, Z. Zou, Z. Shi, W. Zeng, and J. Gui, "Local attention networks for occluded airplane detection in remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 3, pp. 381–385, Mar. 2020.

[16] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning ROI transformer for oriented object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2849–2858.

[17] X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han, "Oriented R-CNN for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3520–3529.

[18] Z. Huang, W. Li, X.-G. Xia, and R. Tao, "A general Gaussian heatmap label assignment for arbitrary-oriented object detection," *IEEE Trans. Image Process.*, vol. 31, pp. 1895–1910, 2022.

[19] Z. Huang, W. Li, X.-G. Xia, X. Wu, Z. Cai, and R. Tao, "A novel nonlocal-aware pyramid and multiscale multitask refinement detector for object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–20, 2022.

[20] Y. Li, H. Mao, R. Liu, X. Pei, L. Jiao, and R. Shang, "A lightweight keypoint-based oriented object detection of remote sensing images," *Remote Sens.*, vol. 13, no. 13, 2021, Art. no. 2459.

[21] X. Zhang *et al.*, "SkyNet: A champion model for DAC-SDC on low power object detection," 2019, *arXiv:1906.10327*.

[22] K. Chen, M. Wu, J. Liu, and C. Zhang, "FGSD: A dataset for fine-grained ship detection in high resolution satellite images," 2020, *arXiv:2003.06832*.

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

[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.

[25] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7263–7271.

[26] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[27] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," *Int. J. Comput. Vis.*, vol. 128, no. 3, pp. 642–656, 2020.

[28] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.

[29] P. Qin, Y. Cai, J. Liu, P. Fan, and M. Sun, "Multilayer feature extraction network for military ship detection from high-resolution optical remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 11058–11069, 2021.

[30] G. Cheng *et al.*, "Dual-aligned oriented detector," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–11, 2022.

[31] Y. Yang *et al.*, "AR ^{2} Det: An accurate and real-time rotational one-stage ship detector in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.

[32] Q. Ran, Q. Wang, B. Zhao, Y. Wu, S. Pu, and Z. Li, "Lightweight oriented object detection using multi-scale context and enhanced channel attention in remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 5786–5795, 2021.

[33] Z. Huang, W. Li, X. G. Xia, H. Wang, and R. Tao, "LO-Det: Lightweight oriented object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, no. 99, pp. 1–15, 2022.

[34] G. Cheng *et al.*, "Anchor-free oriented proposal generator for object detection," 2021, *arXiv:2110.01931*.

[35] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[36] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-yolov4: Scaling cross stage partial network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13029–13038.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[39] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[40] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.

[41] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.

[42] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, vol. 42, no. 8, pp. 2011–2023.

[43] J. Gildenblat *et al.*, "Pytorch library for cam methods," 2021. [Online]. Available: https://github.com/jacobgil/pytorch-grad-cam

[44] Q. Ming, L. Miao, Z. Zhou, X. Yang, and Y. Dong, "Optimization for arbitrary-oriented object detection via representation invariance loss," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[45] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9626–9635.

[46] D. Zhou *et al.*, "IoU loss for 2D/3D object detection," in *Proc. Int. Conf. 3D Vis.*, 2019, pp. 85–94.

[47] J. Han, J. Ding, J. Li, and G.-S. Xia, "Align deep features for oriented object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–11, 2022.

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

[49] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[50] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization 3rd Int," in *Proc Conf. Learn. Representations, San*, 2014.

Huanqian Yan received the bachelor's degree in computer science and technology from the Changchun University of Science and Technology, Changchun, China, in July 2015, and the master's degree in computer application and technology from Lanzhou University, Lanzhou, China, in July 2018. He is currently working toward the Ph.D. degree in computer application and technology with the School of Computer Science and Engineering, Beihang University, Beijing, China.

His current research interests include object detection, adversarial examples, and clustering analysis, etc.

Bo Li received the B.S. degree from Chongqing University, Chongqing, China, in 1986, the M.S. degree from Xian Jiaotong University, Xi'an, China, in 1989, and the Ph.D. degree from Beihang University, Beijing, China, in 1993, all in computer science.

He joined the School of Computer Science and Engineering, Beihang University. He has authored or coauthored more than 100 academic papers in diverse research fields, including intelligent perception, big data intelligence, remote sensing image fusion, and intelligent hardware.

Hong Zhang received the B.E. degree in computer science and technology in 2020 from ShenYuan Honors College, Beihang University, Beijing, China, where he is currently working toward the M.S. degree in computer science and technology with the School of Computer Science and Engineering.

His research interests include remote sensing image processing and object detection.

Xingxing Wei received the bachelor's degree in control science and engineering from the School of Automation and Electrical Engineering, Beihang University (BUAA), Beijing, China, in July 2010, and the Ph.D. degree in computer application and technology from the School of Computer Science and Technology, Tianjin University, Tianjin, China, in July 2015, advised by Prof. Xiaochun Cao.

He is currently an Associate Professor with the School of Computer Science and Engineering, BUAA. From 2017 to 2019, he was a Postdoc Researcher with the Department of Computer Science and Technology, Tsinghua University, Beijing, China, working with Prof. Jun Zhu. Prior to this, he was with Ant Financial, Hangzhou, China, as a Senior Algorithm Engineer. His research interests include remote sensing, adversarial examples, etc.