FRS-Net: An Efficient Ship Detection Network for Thin-Cloud and Fog-Covered High-Resolution Optical Satellite Imagery

Zhiqi Zhang, Huigang Zheng, Jinshan Cao, Xiaoxiao Feng, and Guangqi Xie

***Abstract*—According to statistics, more than 50% of optical satellite images are covered by clouds or fog. Moreover, the cloud cover rate is much higher in large bodies of water and nearby areas than inland areas due to the high amount of water evaporation and condensation. Therefore, ship detection from optical remote sensing images based on water surface analysis is more susceptible to cloud and fog interference, which affects the detection accuracy. In the time-sensitive application field of remote sensing images, due to the massive parameters in the large-scale network model, the detection speed is slow, and a lightweight detection model is commonly used. However, it is difficult for the lightweight detection model to achieve both high efficiency and accuracy for ship detection in a cloud-covered environment. To solve these problems, this manuscript proposes a lightweight algorithm called the fog remote sensing ship detection network (FRS-Net) suitable for ship detection from remote sensing images in thin-cloud and fog-covered environments. FRS-Net is developed based on the deep learning algorithm and can effectively improve ship detection accuracy under thin-cloud and fog-covered environment. First, for the allocation strategy of anchor boxes, by using the K-means clustering algorithm, FRS-Net simplifies the number of anchor boxes by utilizing the shape characteristics of the ship, which improves the time efficiency and the detection accuracy. Second, the FRS-Net network can meet the detection accuracy requirement and has a fast inference speed. The FRS-Net network is mainly composed of a backbone extraction network, a feature fusion network, and a prediction network. Experimental results on the ship detection in optical remote sensing image dataset demonstrate the detection accuracy and computational efficiency of FRS-Net. The recognition mean average precision achieved 43.20% for ship detection under thin-cloud and fog-covered environment, with an efficiency of up to 424 frames per second. FRS-Net has the potential to be applied in future scenarios such as embedded processing and onboard processing, where computing capabilities are strictly limited and the timeliness requirement is high.**

***Index Terms*—Cloud cover, deep learning, high-resolution optical satellite imagery, object detection.**

Manuscript received 16 July 2022; revised 26 September 2022 and 8 November 2022; accepted 4 December 2022. Date of publication 7 December 2022; date of current version 15 December 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61901307, in part by the Open Research Fund of State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University under Grant 20E01, and in part by the Scientific Research Foundation for Doctoral Program of Hubei University of Technology under Grants BSQD2020054 and BSQD2020055. *(Zhiqi Zhang and Huigang Zheng contributed equally to this work.) (Corresponding author: Jinshan Cao.)*

Zhiqi Zhang is with the School of Computer Science, Hubei University of Technology, Wuhan 430068, China, and also with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan 430079, China (e-mail: zzq540@hbut.edu.cn).

Huigang Zheng, Jinshan Cao, and Xiaoxiao Feng are with the School of Computer Science, Hubei University of Technology, Wuhan 430068, China (e-mail: 102011015@hbut.edu.cn; caojs@hbut.edu.cn; 20220026@hbut.edu.cn).

Guangqi Xie is with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan 430079, China (e-mail: xiegqrs@whu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2022.3227322

I. Introduction

SHIP detection technology based on optical remote sensing images is widely used in river monitoring [1], [2], [3], [4], port management [5], [6], [7], [8], and illegal border crossing detection [9], [10]. Since ships are typical moving targets, such tasks have high requirements on timeliness, and the traditional satellite remote sensing application mode that focuses on the postprocessing of high-quality images is difficult to meet the needs. Fortunately, in order to quickly obtain important information and improve satellite remote sensing service capabilities, researchers have carried out exploration and research on onboard processing methods for remote sensing satellites and have achieved certain results [11], [12], [13], [14], [15], [16], [17], [18]. However, due to the limitations of the space environment, such as volume, weight, power consumption, and radiation resistance, the performance of the onboard processing hardware is much lower than that of a common computer, but its timeliness requirements are higher. Future onboard applications must consider the timeliness of algorithms as a key factor. On the other hand, unlike the high-quality optical satellite images that are generally distributed after being selected during ground processing, the average cloud coverage in the first-hand images obtained by the satellite exceeds 50% [19], [20], and a large number of potential targets are blocked by different degrees of clouds or fog. This situation interferes with the accuracy of target detection algorithms in practical applications. Especially due to the evaporation of water vapor and condensation, water areas are more prone to cloud cover than inland areas. This phenomenon increases the difficulty of target detection based on optical remote sensing images [21], [22], [23]. For the ship detection algorithm based on optical satellite images, when the target is covered by thick clouds, it cannot be detected at all, but under the conditions of thin clouds and fog, the main features of the target still exist and can still be detected by algorithms. For this situation, the typical solution usually contains two steps. First, the interference, such as thin clouds and fog, is removed, and then the detection is carried out on the basis of removal.

Generally, acceptable results can be obtained [24], [25], [26]. However, due to the complexity of the process, this solution is time-consuming and has insufficient overall timeliness, making it difficult to apply to real-time applications; on the other hand, it is easy to cause damage to small targets and reduce detection accuracy. For future onboard applications with high timeliness, more efficient and accurate methods need to be explored.

In previous studies, several complex traditional algorithms were used for ship detection, such as Shuai et al.'s, which obtained ship target candidate regions based on visual saliency detection and Fourier transform first and then proceeded with ship target detection [27]. Qi et al. proposed an unsupervised ship detection method to detect ships through candidate boxes and uniform filtering [28]. Yang et al. proposed a ship detection method based on structured forest edge detection. The scale-invariant feature transform feature operator and support vector machine are used to extract and filter ship targets and finally achieve target detection [29]. However, this kind of method usually requires a lot of computation and high time consumption, and it lacks robustness and stability. A lot of missed detections and false detections occur when images are affected by noise, such as clouds or fog [30], [31].

Accordingly, various deep learning methods have been used for target detection tasks based on remote sensing images. In general, the target detection algorithm based on deep learning can be divided into one-stage detection and two-stage detection. Compared with the one-stage target detection algorithm, the two-stage target detection algorithm appeared earlier. Including region-based convolutional neural networks (R-CNN) [32], fast R-CNN [33], faster R-CNN [34], and cascade R-CNN [35], they have better performance and can more accurately identify the target in complex environments. The strategy of most two-stage algorithms is extracting candidate regions first and then performing target detection, which brings higher accuracy to the model, but with the consumption of parameters and computability, such algorithms are not suitable for time-sensitive applications. In contrast, the one-stage algorithm directly predicts the location and category of the target by extracting features. Compared with the two-stage detection algorithm, models adopting this detection strategy have fewer parameters and faster computation, such as the You Only Look Once series [36], [37], [38], single shot multi-box detector (SSD) [39], RetinaNet [40], and CenterNet [41]. However, in order to apply such algorithms to achieve high accuracy and quick response in environments where computability is strictly limited, further optimization and improvement are still needed.

Unlike the images in computer vision (CV), such as Microsoft Common Objects in Context (COCO) [42] and Pascal Visual Object Classes (VOC) [43], targets in satellite remote sensing images are usually very small because remote sensing images are taken from a distance, usually hundreds or even thousands of kilometers from the ground. For example, in a remote sensing image of 1000×1000 pixels, a ship target only occupies 50×50 pixels [44], [45]. In terms of algorithms, the mainstream target detection algorithms based on deep learning come from the field of CV. However, the field of CV pays more attention to larger objects with a closer distance because they are more important to the observer, whereas in the remote sensing field, the important targets are all small and far away. Especially when these small targets are densely distributed, it will bring a great challenge to the detection algorithm. Due to the unique characteristics of remote sensing images, it is a challenging task to transfer target detection algorithms based on deep learning from CV to remote sensing. When applying the deep learning algorithm to the ship detection task of remote sensing images covered by clouds and fog, the major issues can be generally summarized as low detection efficiency, easy to be disturbed by interference, and low accuracy, especially in the case of dense target distribution.

To address the above-mentioned problems, this manuscript proposes a lightweight ship detection algorithm, named fog remote sensing ship detection network (FRS-Net), suitable for ship detection from remote sensing images in thin-cloud and fog-covered environment. FRS-Net is developed based on the deep learning algorithm and offers both high accuracy and efficiency. First, we propose a novel anchor box matching strategy that uses the K-means clustering algorithm [46] for the target shape to simplify the number of anchor boxes and improve the detection efficiency. Second, in order to solve the problem that the existing algorithms are difficult to achieve a balance between accuracy and efficiency in the task of ship detection in thin-cloud and fog-covered environments, this manuscript constructs the FRS-Net network, which consists of a backbone extraction network, a feature fusion network, and a prediction network. Among them, the backbone extraction network is the Cross Stage Partial Connections Darknet-Tiny (CSPDarknet53-Tiny), which has a small number of parameters and computation and is suitable for time-sensitive application requirements. The feature extraction network follows the idea of the feature pyramid network (FPN) to improve the robustness of the algorithm in complex environments by fusing high-level semantic information with detailed information. In the prediction network, we choose an appropriate prediction scale to alleviate the accuracy drop in dense ship distribution scenarios in thin-cloud and fog-covered environments. To verify the effectiveness of the proposed FRS-Net, we construct a dataset called ship detection in optical remote sensing images (SDIOR) based on the public DIOR [47] dataset and actual images from Google Earth, using the dark channel prior algorithm [48] to simulate different degrees of thin clouds and fog. Finally, the FRS-Net algorithm is compared with other state-of-the-art algorithms. The accuracy-efficiency comparison of FRS-Net with state-of-the-art algorithms and works is shown in Fig. 1.

II. Related Work

A. Ship Detection Algorithms Based on Deep Learning

Considering that the target scale in remote sensing images is small and susceptible to interference, especially in the case of dense distribution, most detection algorithms focus on improving recognition accuracy and pay less attention to efficiency. Following this consideration, researchers usually enhance the generalization ability and robustness of the algorithm by building more complex network models. For example, Zhang et al.

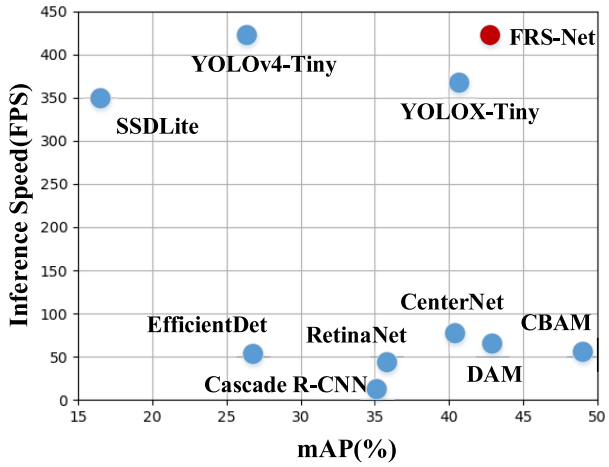

Fig. 1. Accuracy–efficiency comparison of FRS-Net with state-of-the-art algorithms and works.

proposed a ship detection method based on R-CNN, which improved the detection accuracy of small target ships by modifying the anchor box and by combining different resolution features [49]. Guo et al. proposed a fully deformable convolutional network for intensive ship detection tasks, designed an enhanced feature pyramid network to boost more accurate spatial and semantic information flow in the network, and added an adaptive balanced feature integrated module to enhance valuable features [50]. However, considering the high-timeliness requirements for ship detection tasks in practical application scenarios, such as target tracking and emergency rescue, as well as the application requirements for high-accuracy real-time processing under limited conditions in the future, it is necessary to enhance the importance of algorithm efficiency, which is regarded as important as accuracy.

B. Ship Detection Strategy in Foggy Environment

Optical remote sensing images are very susceptible to clouds and fog, resulting in degraded image quality. In this case, the traditional method usually does not consider the timeliness very much and generally regards the thin cloud and fog as noise, i.e., first seeks toremove it through an algorithm, and then performs subsequent processing. For example, Zou et al. constructed a task partition model based on a fully connected network, designed a more simplified and robust ship detection framework compared with manual feature extraction, and increased the robustness of the algorithm in complex environments [51]. You et al. proposed a ship detection algorithm named Scene Mask R-CNN that can effectively detect false candidate targets in nontarget areas and identify nearshore ships [52]. Chen et al. proposed a new ship detection method based on the combination of discrete wavelet transform and ship detection by building a dual generative adversarial network to complete defogging and sampling the region of interest for ship detection [53]. Song et al. proposed an improved parallel ISSD ship detection algorithm based on the SSD target detection network [45]. By introducing a new feature extraction layer generated by a saliency dataset and the extraction layer of the original network for feature fusion [54], the ship recognition rate under the occlusion of thin clouds and fog is effectively improved. This method effectively improves the ship recognition rate in an environment of thin-cloud and fog occlusion. Wang et al. proposed a ship detection algorithm under sea fog weather condition. The ship target is obtained by dividing the clear and foggy images, then using the dark channel prior algorithm for defogging, and finally detecting it using the YOLOv3 algorithm [55]. Chen et al. proposed a ship detection method based on an attention mechanism. By designing a novel lightweight dilated attention module (DAM) to extract the features of ship targets, the algorithm's detection capability in complex environments is improved [56]. However, on the one hand, the process of such algorithms is relatively complicated, and the pre-defogging operation significantly increases the processing time, which further reduces the overall processing efficiency; on the other hand, it is easy to damage the small target features in the process of dehazing and reduce the detection rate of real targets.

In summary, in the future practical application scenarios with high timeliness, for the optical satellite ship detection task covered by thin clouds and fog, it is necessary to consider the construction of suitable algorithms under the condition of limited computability. That is to say, while focusing on the efficiency of the algorithm, it is also important to take into account the detection accuracy and robustness, especially in the case of dense target distribution scenarios. In order to better apply to onboard processing of first-hand images with cloud coverage of up to 50% obtained on the satellites in the future.

III. Methods

Although the target detection algorithm based on deep learning has satisfactory performance in the VOC and COCO target detection tasks, it is difficult to directly apply it to the ship detection task from remote sensing images with thin-cloud and fog coverage. The main reason is that, unlike the natural image target detection task that mainly focuses on close-range targets, the ship targets in remote sensing images are small and narrow. Moreover, ship targets often disperse densely in small areas, such as ports and rivers. In addition, there exists a different degree of thin-cloud and fog coverage, which makes it difficult to achieve ideal results by directly using this kind of target detection algorithm. To improve the detection accuracy and inference speed of remote sensing image ships under the simultaneous interference of thin clouds and fog, this manuscript first proposes a more appropriate anchor box setting and allocation strategy according to the scale and shape characteristics of the ship. For the problem of insufficient robustness under interference conditions, the proposed FRS-Net constructs a feature fusion layer, which enables the shallow network to obtain more high-level semantic information and improves the algorithm's ability to identify targets in complex environments. Aiming at the problem that the detection effect of the densely distributed ship scene becomes worse when covered by thin clouds and fog, the detection accuracy of the algorithm is improved by selecting an appropriate prediction scale in the prediction network.

Algorithm 1: K-Means Clustering of Anchor Box Size.

Input: Training set: $\{x_i, x_2, x_m\}$, Cluster number: K

Output: Cluster division: $\{C_1, C_2, C_k\}$, Cluster centers: $\{\mu_1, \mu_2, \mu_k\}$

1. Randomly pick K samples from the training set as the initial cluster centers: $\mu_1, \mu_2, \ldots, \mu_k$;
2. Calculate the distance of sample x_i to every cluster centers μ_j, (j = 1, 2, .., k): $\text{Distance}(x_i, \mu_j) = 1 - \text{IoU}(x_i, \mu_j)$. Then put x_i into the nearest cluster set: $C_j = C_j \cup x_i$;
3. Calculate the new center of every cluster according to the samples in the cluster: $u_j = \frac{\sum_{x_i \epsilon C_j} x_i}{\text{len}(C_j)}$;
4. Repeat step 2 and 3, util the new cluster centers are stable;
5. Output final division $\{C_1, C_2, C_k\}$, and cluster centers $\{\mu_1, \mu_2, \mu_k\}$.

A. Anchor Box Matching Strategy for Ship Targets

Anchor box is a mechanism created in the field of target detection to replace the traditional image pyramid method to predict multiscale targets. It can set anchor boxes of different scales in the same feature output layer to detect the features of targets of different scales in the image. The basic idea is that by presetting different anchor box sizes, and then after the training of the neural network and parameter regression, the original fixed-size anchor box can generate prediction boxes of different sizes.

The size of the anchor box can be set according to the maximum possible width and height of the target, so the anchor box has direct impact on the performance of the target detection network to a certain extent. Therefore, choosing the appropriate anchor box size and number can not only help the network to converge better and infer faster but also improve the overall efficiency of ship detection.

The purpose of the anchor box design in the target detection algorithm is to detect different scale targets under the same scale feature, but the more anchor boxes there are, the more complex the network, so simply assigning more refined anchor boxes will reduce the network timeliness. Since the ship shapes are mostly narrow and long, this feature can simplify the number of anchor boxes and generally satisfy the algorithm's fitting of the ship boundary information.

To better balance efficiency and accuracy, this manuscript uses the K-means clustering algorithm to obtain the appropriate anchor box size setting and can obtain the preset anchor box that is conducive to ship detection according to the shape characteristics. The algorithm mainly has the following two steps: 1) calculate the distance of every target in the training set to the cluster center, and 2) calculate new cluster centers according to the clustering result. The detailed process is described in Algorithm 1.

InFig. 2, the anchor boxes assigned to the samples in the dataset are visualized. The horizontal and vertical axes represent the length and width in pixels, respectively. Fig. 2(a) shows the assignment of four anchor boxes. Fig. 2(b) shows the assignment of six anchor boxes. The yellow stars represent anchor boxes, and the points with different colors represent samples assigned to different anchor boxes. Specifically, by extracting the length and width of all target samples in the training set and all anchor boxes clustered by the K-means clustering algorithm, we perform intersection-over-union (IoU) calculations on all target samples and anchor boxes to obtain the anchor boxes assigned to each target sample point. Fig. 2(a) and (b) have the same samples, but due to the reduction in anchor boxes, the number of target samples matched by each anchor box increases.

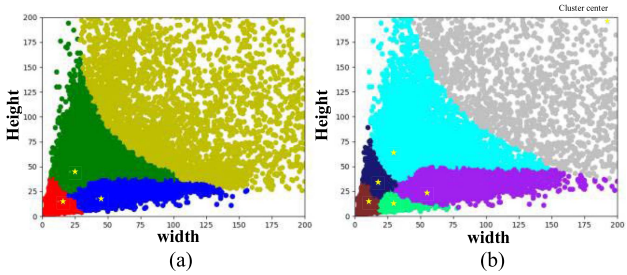

Fig. 2. Anchor box assignment with different settings. (a) Four anchor boxes strategy. (b) Six anchor boxes strategy.

TABLE I
FRS-NET WITH DIFFERENT ANCHOR PARAMETERS

Algorithm	FRS-Net		
Cluster	4	6	8
Feature Scale (52 × 52)	(12,15) (21,45)	(10,15) (29,13) (17,35)	(9,8) (11,19) (26,11) (16,34)
Feature Scale (26 × 26)	(41,18) (143,147)	(54,24) (30,64) (192,200)	(42,18) (25,61) (69,36) (201,212)

In order to evaluate the influence of the number of anchor boxes on efficiency and accuracy, we performed the K-means clustering algorithm on different settings of four, six, and eight anchor boxes. As shown in Table I, "Cluster" represents the number of anchor boxes, and "Feature Scale" represents different feature layers.

B. FRS-Net Feature Extraction and Fusion Network

In order to improve the robustness of the algorithm in complex environments, by establishing the connection between the shallow network and the deep network, FRS-Net fuses the low-level detailed information from the shallow network with the high-level semantic information from the deep network. Fig. 2 shows the detailed structure of FRS-Net.

In Fig. 3, the FRS-Net is mainly composed of the following three parts: the backbone feature extraction network, the feature fusion network, and the prediction network. To meet the needs of high-timeliness applications, FRS-Net adopts the backbone extraction network CSPdarknet53-Tiny, which consists of three

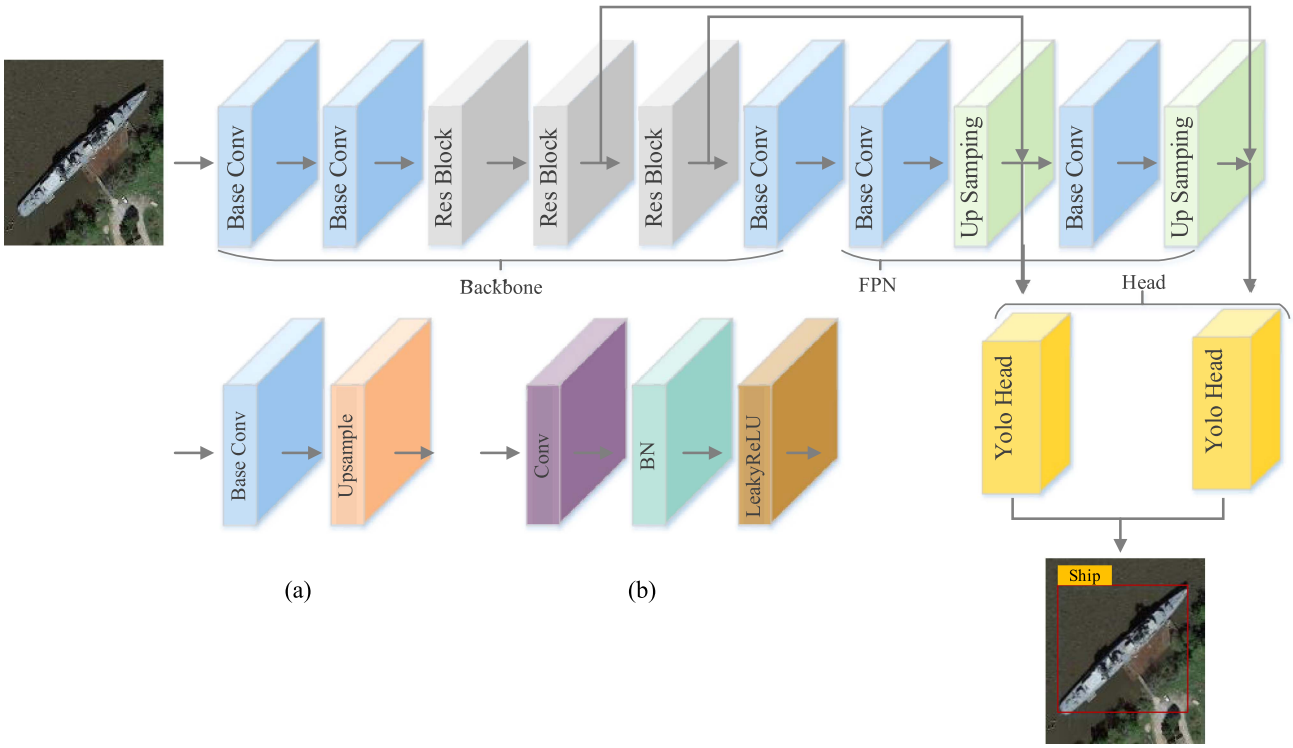

Fig. 3. Network structure of FRS-Net. (a) Upsampling. (b) Base Convolution.

base convolutions and two large residual blocks. The base convolution is composed of ordinary convolution, batch normalization (BN), and the leaky rectified linear unit (Leaky ReLU) activation function. The residual module can solve vanishing gradients and achieve good performance even with deeper networks.

The feature fusion network adopts the FPN algorithm. It is well known that shallow feature layers possess more low-level semantic information, including bounding box location information, but lack high-level semantic information. More detailed information is brought in through the shallow network, which is beneficial to the network to accurately locate the ship target in a complex environment. On this basis, the high-level semantic information of the deep network is transmitted to the shallow network through the FPN, further improving the network's ability to detect ships in thin-cloud and fog-covered environments.

In the training part, the sample image is resampled to 416×416. After "feature extraction" and "downsampling" of the backbone network, three feature maps with different feature scales, $C \times 13 \times 13$, $\frac{C}{2} \times 26 \times 26$, and $\frac{C}{4} \times 52 \times 52$, are obtained. In the feature fusion network, to fuse the parameters with the feature map $\frac{C}{2} \times 26 \times 26$, $C \times 13 \times 13$ needs to adjust the channel and feature map space size. After "base convolution" and "upsampling," the feature scale $C \times 13 \times 13$ is changed to $\frac{C}{4} \times 26 \times 26$, and the spatial size of the feature map becomes the same. Then, it is superimposed on the channel dimension to generate a feature map $\left(\frac{C}{4}+\frac{C}{2}\right) \times 26 \times 26$. At this time, the feature map has two functions. First, the channel is converted through the prediction head and then sent to the prediction network to predict the target bounding box, category, and confidence. Second, we continue to adjust the space size of the channel and feature map and perform parameter fusion with the feature map $\frac{C}{4} \times 52 \times 52$ to generate the feature map $\left(\frac{C}{16}+\frac{C}{4}\right) \times 52 \times 52$ and then send it to the prediction network to calculate feature information.

In the FRS-Net feature layer target detection allocation strategy, the feature output layer 52×52 is used to detect small targets, and the feature output layer 26×26 is used to detect nonsmall targets. To match the feature information of the feature output layer 26×26 with the feature information of the feature output layer 52×52, the "base convolution" and "upsampling" operations are added to the feature output layer with the feature scale of 26×26 to ensure information matches. When the feature information of the feature output layer 26×26 is parameterized with the feature information of the feature output layer 52×52, the shallow network can obtain high-level semantic information from the deep network to improve the robustness of the network model in thin-cloud and fog-covered scenarios.

C. FRS-Net Prediction Network

Small ship targets are often densely distributed in narrow areas, such as ports and rivers. On this basis, when there is thin-cloud and fog coverage, it will aggravate the missed and false detection of the target detection algorithm. The basic reason for this situation is that the extracted features of the convolutional network are inconsistent with those predicted.

In the prediction network, computing feature information requires dividing the input image into $S \times S$ grid units, and each

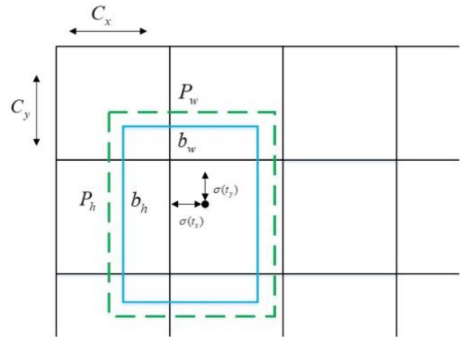

Fig. 4. Bounding box prediction.

grid unit is used to predict the object category and bounding box information. Fig. 3 shows the bounding box prediction.

In Fig. 4, (C_x, C_y) represent the coordinates of the upper-left corner of the cell where the target center point is located. (p_w, p_h) represent the width and height of the prior box, respectively. (b_w, b_h) represent the width and height of the prediction box, respectively. $\sigma(t_x)$ and $\sigma(t_y)$ represent the offset values predicted by FRS-Net. The boundary of the prediction box can be calculated by formulas (1)–(5), where t_w and t_h are also predicted by the convolutional network

$$b_x = \sigma\left(t_x\right) + C_x \tag{1}$$

$$b_y = \sigma\left(t_y\right) + C_y \tag{2}$$

$$b_w = P_w\, e^{t_w} \tag{3}$$

$$b_h = P_h\, e^{t_h} \tag{4}$$

$$\sigma\left(\mathrm{x}\right) = \frac{1}{1+e^{-x}}\,. \tag{5}$$

Since the prediction network only allows one grid to match one target, when the multiple-target feature information appears in the same grid, the grid center only retains the last target information. This results in inconsistencies between the extracted and predicted features. For ship detection in a port scene, there will be small ship objects of similar shape and size densely distributed together. In this case, the prediction layer of the small ship needs to select an appropriate prediction scale. A too-large prediction scale will complicate the network, and a too-small prediction scale will increase the rate of missed detection and false detection. Fig. 4 shows what happens when the prediction network chooses different prediction scales.

In Fig. 5, there are two boats distributed closely; the features are extracted by FPN and sent to the prediction network, and the features are converted into frame information. First, the target center point in the frame is activated by the activation function (sigmoid) and output $\sigma(t_x) \in (0, 1), \sigma(t_y) \in (0, 1)$. The length and width of each network are set to 1 in size. When the predicted feature layer is 26×26, the center point of the target needs to match the size of the feature map after normalization, which causes the center points of the two boats to be assigned to the same grid. The upper-left corner coordinate (C_x, C_y) of the grid can only match one target center point, so only one target is calculated in the prediction network, which is inconsistent with the extracted features. However, the two boats have the same width and height and belong to the same detection layer. Adding anchor boxes cannot solve this problem but makes the network inference slower. When the predicted feature layer is 52×52, as shown in Fig. 5(b), the target center point obtains the new grid, and the spatial distance between the two target center points is also increased and assigned to different grids. In the densely distributed remote sensing, ship detection scene is covered by thin clouds and fog. The FRS-Net algorithm selects the feature scale for the small target prediction layer in the prediction network as 52×52, which can alleviate the inconsistency between the extracted features and the predicted features, thereby improving the detection accuracy of ship targets in this scenario.

IV. Dataset and Other Details

A. Simulation Dataset (SDIOR)

To verify the effectiveness of the proposed FRS-Net, we construct a dataset called SDIOR based on the public DIOR dataset and images from Google Earth, using the dark channel fogging algorithm to simulate different degrees of thin clouds and fog.

The DIOR dataset is one of the largest, most diverse, and publicly available target detection datasets in the earth observation community. The size of the RGB images in the dataset is 800 × 800 pixels, and the ground sample distance is in the range of 0.5–30 m.

The images collected from the DIOR dataset and Google Earth are all clear ship images with good weather conditions. To be closer to the actual situation in the thin-cloud and fog-covered environment, this manuscript uses the dark channel prior algorithm to add some thin clouds and fog to the collected images. Different simulation strategies are applied for the training and test sets.

For the training set, in order to be as close to the actual situation as possible, the original training set was randomly divided into ten parts, and 1–9 levels of simulated fog were added to the original image (level 0 means no fog). The processed training set was used to train all the networks. For the test set, the original set was duplicated ten times and 1–9 levels of simulated fog were added to each. Ten test subsets were obtained, with a total number of ten times the original.

In total, we have collected 1302 images in the training set and 15 430 images in the test set, which contains 10 subsets of different levels of fog. Fig. 6(a) shows the original image, and Fig. 6(b)–(j) represent images with different degrees of fog with coefficients ranging from 0.45 to 0.05. Among them, the blue-dashed box is a fog map partially covered by fog.

B. Dark Channel Fogging Algorithm

The dark channel fogging algorithm is based on the dark channel prior algorithm, and the fogging model used is as follows:

$$I\left(x\right) = J\left(x\right) t\left(x\right) + A\left(1 - t\left(x\right)\right) \tag{6}$$

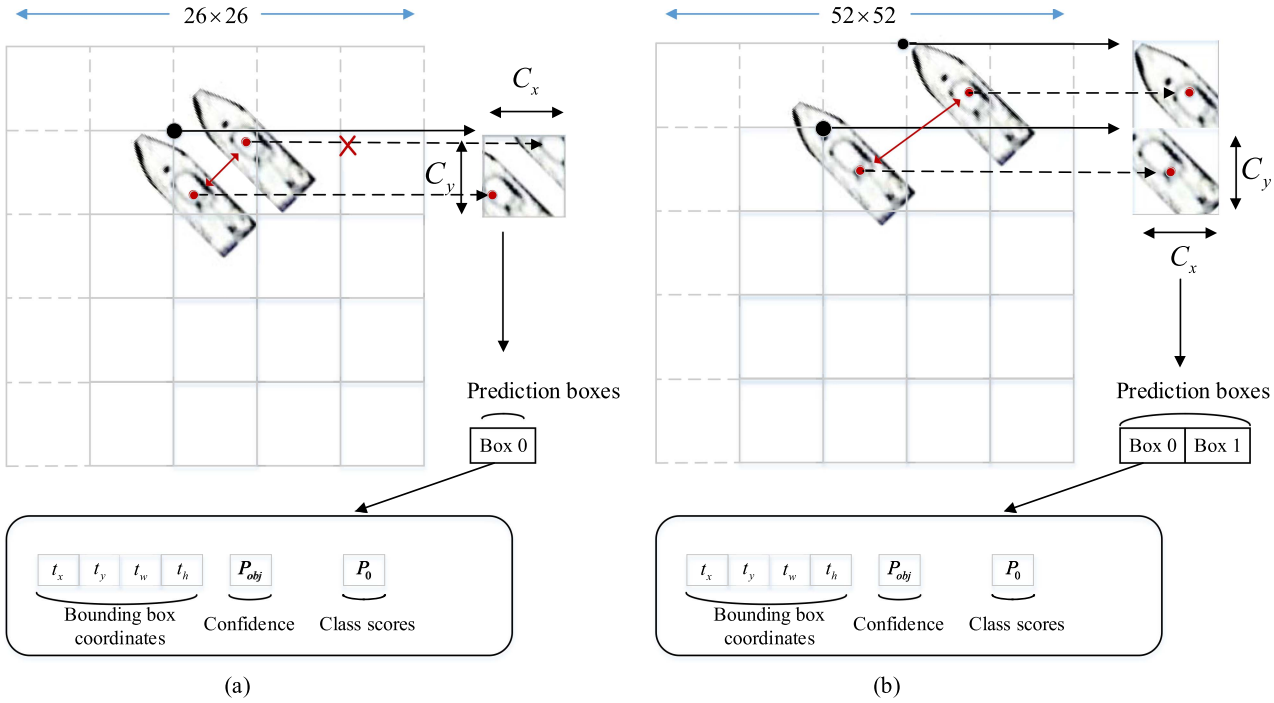

Fig. 5. Comparison of processing multiple targets by predictive networks with different feature scales. (a) 26 × 26 feature scale. (b) 52 × 52 feature scale.

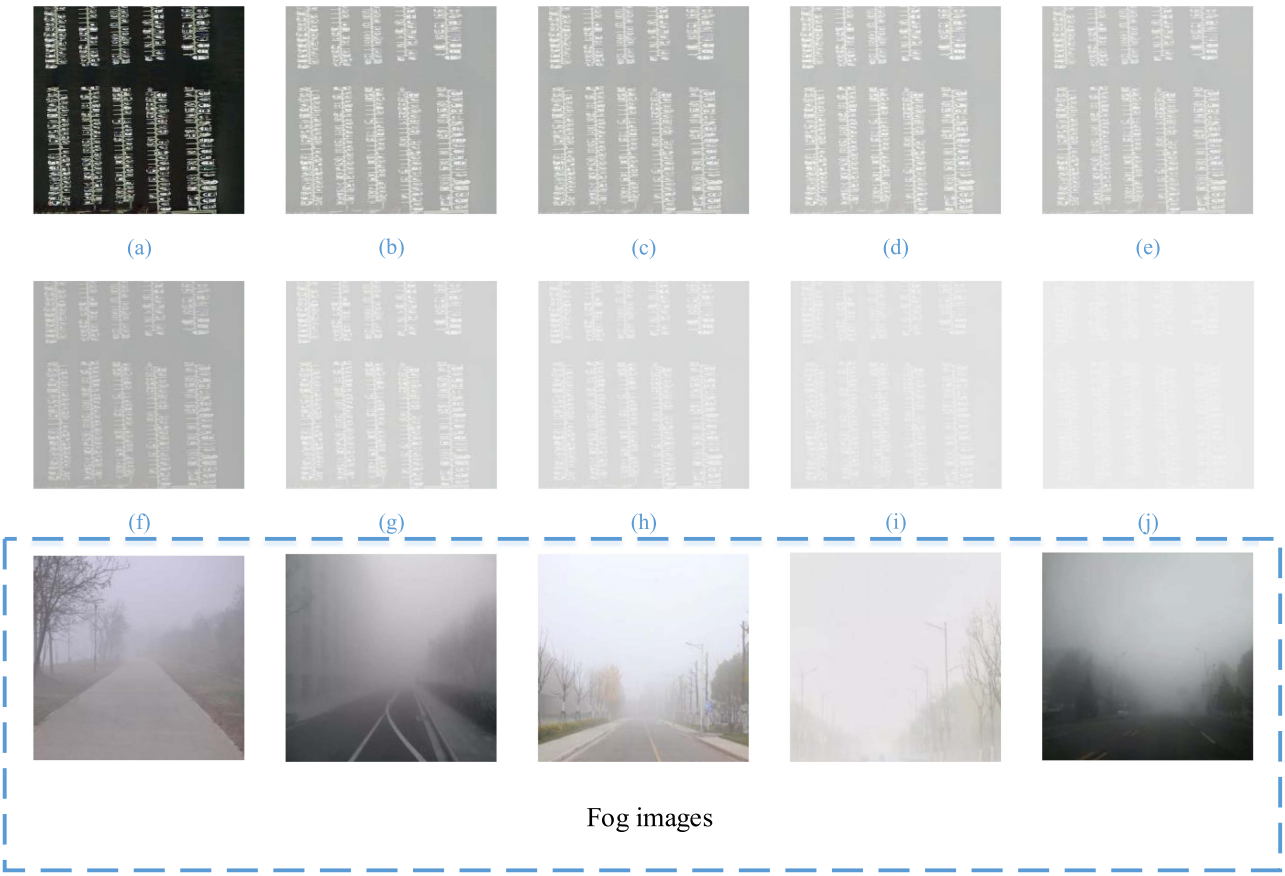

Fig. 6. Original image (a) and images with different fog thicknesses (b)–(j) in SDIOR dataset.

where $J(x)$ represents the clear image. $I(x)$ is the image with fog. $A(.)$ represents the atmospheric light component, and $t(x)$ is the image transmittance. To simulate the fog containing remote sensing images, this manuscript extracts the $t(x)$ and A values of the real fog image through the existing fog maps. These coefficients are used in the above-mentioned model to add fog to the original dataset image and quantitatively control the fog occlusion thickness through specific coefficients during the fogging process. The specific process is as follows:

$$t(x) = 1-\omega \cdot \min_{c\in(r,g,b)} \left(\min_{y\in\Omega(x)} \frac{I^c(y)}{A^c} \right) \tag{7}$$

where ω is set to 0.95. $\Omega(x)$ represents the local area block centered on "x." Transmissivity uses guided filtering to calculate the value.

After obtaining the $t(x)$ and A values of the existing real fog-containing map, the following equation is used to add fog to the original dataset image:

$$J(x) = \alpha \cdot t(x) \cdot (I(x) - A) + A \tag{8}$$

where α represents the thickness of fog that is set in the range of 0.45–0.05, with a step of 0.05, corresponding to the nine thickness levels of fog on the simulated dataset.

C. Evaluation Metrics

In this manuscript, the commonly used evaluation metric IoU is used as the threshold to measure whether the target detection prediction is correct; the average precision (AP) measured on this basis is used as the quantitative evaluation metric of detection accuracy; and the calculation delay (latency), network parameters (parameters), and frames per second (FPS) are used as the quantitative evaluation metric of detection efficiency. The IoU is calculated by the following formula:

$$\mathrm{IoU} = \frac{|\mathrm{B} \cap \mathrm{B_{GT}}|}{|\mathrm{B} \cup \mathrm{B_{GT}}|} \tag{9}$$

where B represents the size of prediction box and $\mathrm{B_{GT}}$ represents the size of ground truth box.

mAP is the average accuracy of all subsets to be detected in a model, so it is often used to evaluate the performance of the network. In detail, the AP is the area of the curve formed by the precision and recall (PR) calculations; this curve is usually called the PR curve and is formed by setting different thresholds for the calculation of precision and recall. It is calculated by the following formulas:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{10}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{11}$$

$$\mathrm{AP} = \int_0^1 P(r)\, dr \tag{12}$$

$$\mathrm{mAP} = \frac{1}{n} \sum_i^n \mathrm{AP}_i \tag{13}$$

TABLE II
EXPERIMENTAL ENVIRONMENT AND HYPERPARAMETERS

Configure	Information
Hardware	Operating System: Ubuntu 20.04.2.0
	GPU: NVIDA GeForce RTX 3060
Software	Python 3.8 + Pytorch1.9.1 cu111 +CUDA 11.6 + OpenCV 4.1.2
Parameter	Batch size: 8
	Learning rate: 0.001
	Number of iterations: 300
	Input size:(416,416)
	Scheduler: Cosine Annealing
	Mosaic: True
	Optimizer: Adam

where TP, FP, TN, and FN denote true positive, false positive, true negative, and false negative, respectively.

Meanwhile, in order to comprehensively measure the performance of the network, this manuscript adopts the AP values under different IoU threshold settings, including $\mathrm{mAP}_{0.5}$, $\mathrm{mAP}_{0.75}$, mAP_S, mAP_M, and mAP_L. Among them, $\mathrm{mAP}_{0.5}$ and $\mathrm{mAP}_{0.75}$ represent the detection accuracy when the IoU is set to 0.5 and 0.75, respectively. mAP_S represents the detection accuracy of the target size less than 32×32. mAP_M represents the detection accuracy of the target size between 32×32 and 96×96. mAP_L represents the detection accuracy of the target size greater than 96×96 [57], [58], [59].

Latency is the elapsed time between making a request and receiving a response and is used to measure the speed of algorithmic inference.

FPS refers to the number of output frames per second.

In addition, to comprehensively measure the detection accuracy and timeliness, this manuscript uses *Synthesis Performance (SP)* as the indicator of the comprehensive detection capability of the network, and the specific calculation is as follows:

$$\mathrm{SP} = \alpha \cdot \mathrm{mAP} + \beta \cdot \omega \cdot \mathrm{FPS} \tag{14}$$

$$\omega = \frac{1}{1+\ln(\mathrm{FPS})} \tag{15}$$

where FPS represents the speed of network inference, and mAP represents the accuracy of network detection. α and β represent the proportions of the two in the comprehensive performance index, which are set to 0.5 and 0.5, respectively.

D. Experimental Environment

The experimental environment and parameter settings of the algorithm are shown in Table II.

As shown in Table II, the hardware environment used in the experiment in this manuscript is the Ubuntu system and the RTX3060 graphics card. Testing in different hardware device environments will result in inconsistent algorithm detection speeds. The software environment is developed based on the PyTorch machine learning library and is equipped with the Compute Unified Device Architecture (CUDA) computing

TABLE III
ABLATION EXPERIMENTS ON FRS-NET

No.	Feature scaling	Anchor strategy			Latency (ms)	Parameter (M)	$mAP_{0.5}$ (%)	FPS	SP
		Anchor(4)	Anchor(6)	Anchor(8)					
1	×		√		2.35	6.5310	26.27	425	43.26
2	√		√		2.40	5.0375	35.22	416	47.18
3	√			√	2.49	5.398	24.02	401	40.67
4	√	√			2.36	5.0352	43.20	423	51.60

framework. Different software versions will lead to inconsistent algorithm precision in decimal places. On the hyperparameters of the algorithm, this manuscript sets the total number of iterations of the algorithm to 300, the learning rate adjustment method is the cosine annealing algorithm, and the optimizer is Adam. The setting of these parameters can speed up the convergence of the algorithm.

V. EXPERIMENT AND ANALYSIS

A. Ablation Experiments

To analyze the contribution of each part to the performance of the network, the ablation experiments are shown in Table III.

In Table III, "Feature scaling" refers to the network structure of different output feature scales. This factor can verify the benefits brought by fine prediction scales and prediction networks. "Anchor strategy" verifies the benefits of setting a different number of anchor boxes.

The first settings is the FRS-Net benchmark network, which the network structure is made by the output characteristic scales 26, 13 and the original anchor box are 6. The data results are $mAP_{0.5}$ (26.27%), FPS (425), parameter (6.5310M), and SP (43.26), respectively.

The second settings is the FRS-Net with six anchor boxes. Compared with the first settings, $mAP_{0.5}$ is increased by 8.95% (26.27% versus 35.22%), the FPS is decreased by 9 (425 versus 416), and SP is increased by 3.92 (43.26 versus 47.18). Experiments show that by reasonably adjusting the size of the feature network and the prediction network, the performance of the FRS-Net algorithm can be effectively improved.

The third settings is the FRS-Net with eight anchor boxes. Compared with the second settings, $mAP_{0.5}$ drops by 11.2% (35.22% versus 24.02%), FPS drops by 15 (416 versus 401), and SP drops by 6.51 (47.18 versus 40.67). Experiments verify that increasing the number of anchor boxes in each output feature layer while keeping the number of output feature scales unchanged does not have a positive impact on the network. It is speculated that the possible reason is that since the FRS-Net target detection algorithm is a lightweight model with a small number of parameters, the increase in anchor boxes makes the model difficult to converge, making the model performance worse.

The fourth settings is the FRS-Net with four anchor boxes. Compared with the second settings, $mAP_{0.5}$ is increased by 7.98% (35.22% versus 43.20%), FPS is increased by 7 (416 versus 423), and SP is increased by 4.42 (47.18 versus 51.60). Compared with the network model in the third settings, $mAP_{0.5}$ is increased by 19.18% (24.02% versus 43.20%), FPS is increased by 22 (401 versus 423), and SP is increased by 10.93 (40.67 versus 51.60). Experiments show that using ship characteristics to simplify the anchor box and selecting the appropriate anchor box size can not only improve the detection efficiency but also enhance the accuracy.

B. Comparison With State-of-the-Art Algorithms and Works

To verify the effectiveness of the proposed method, FRS-Net is compared with SSDLite, RetinaNet, CenterNet, cascade R-CNN, YOLOv4-Tiny, YOLOX-Tiny, EfficientDet, DAM [56], and CBAM [60] on the SDIOR dataset. The experimental results are shown in Table IV. Meanwhile, Fig. 7 shows the PR curves of each algorithm. Fig. 8 shows the receiver operating characteristic (ROC) curves. Fig. 9 shows the SP of each algorithm.

In Table IV, the Latency column contains three parts, i.e., net latency, which means the time of network inference, anchor latency, which means the time of anchor box inference, and total latency. Other indicators used to measure the performance of the model include $mAP_{0.5}$ (%), model parameter (M), FPS, e network inference time latency (ms), and SP.

Compared with the SSDLite, the $mAP_{0.5}$ of the FRS-Net is increased by 26.83% (16.37% versus 43.20%), network inference time increased by 0.17 ms (1.54 ms versus 1.71 ms), anchor box processing time decreased by 0.7 ms (1.35 ms versus 0.65 ms), FPS is increased by 75 (348 versus 423), and SP is increased by 18.03 (33.57 versus 51.60).

Compared with the RetinaNet, the $mAP_{0.5}$ of the FRS-Net is increased by 7.58% (35.62% versus 43.20%), FPS is increased by 382 (41 versus 423), and SP is increased by 29.46 (22.14 versus 51.60).

Compared with the CenterNet network, the $mAP_{0.5}$ of the FRS-Net is decreased by 0.01% (43.21% versus 43.20%), the FPS is increased by 352 (71 versus 423), and the SP is increased by 23.27 (28.33 versus 51.60).

Compared with the Cascade R-CNN, the FRS-Net improves the $mAP_{0.5}$ by 8.01% (35.19% versus 43.20%), the FPS is increased by 406 FPS (17 FPS versus 423 FPS), and the SP is increased by 31.82 (19.78 versus 51.60).

Compared with the YOLO4-Tiny, the $mAP_{0.5}$ of the FRS-Net is increased by 16.93% (26.27% versus 43.20%), the FPS is decreased by 2 (425 versus 423), and the SP is increased by 8.34 (43.26 versus 51.60).

Compared with the YOLOX-Tiny, the $mAP_{0.5}$ of the FRS-Net is increased by 2.11% (41.09% versus 43.20%), the FPS is increased by 57 (366 versus 423), and the SP is increased by 4.55 (47.05 versus 51.60).

TABLE IV
COMPARISON WITH OTHER STATE-OF-THE-ART MODELS AND WORKS WITH DIFFERENT THICKNESSES OF FOG

Model	Latency (ms)			Parameter (M)	$AP_{0.5}$ (0.45–0.05 represent different thicknesses of fog)										$mAP_{0.5}$ (%)	FPS	SP
	Net	Anchor	Total		Ori	0.45	0.4	0.35	0.3	0.25	0.2	0.15	0.1	0.05			
SSDLite	1.54	1.35	2.89	3.68	12.23	16.12	19.11	18.87	18.84	15.80	15.41	16.28	14.66	16.45	16.37	348	33.57
RetinaNet	23.28	0.54	23.81	36.33	35.45	35.63	35.65	35.65	35.67	35.70	35.67	35.67	35.65	35.49	35.62	41	22.14
CenterNet	11.12	2.78	13.90	32.66	44.07	44.06	44.08	44.08	44.12	44.12	44.12	44.08	44.02	35.37	43.21	71	28.33
Cascade R-CNN	33.86	23.36	57.22	69.10	36.07	41.02	40.51	39.10	38.63	40.20	35.26	32.94	29.36	18.81	35.19	17	19.78
YOLOv4-Tiny	1.65	0.70	2.35	6.53	30.15	27.94	27.64	27.48	27.07	26.53	26.12	25.05	23.87	20.82	26.27	425	43.26
YOLOX-Tiny	2.37	0.36	2.73	5.03	44.61	44.72	44.70	44.68	44.65	44.64	44.55	44.45	35.82	18.05	41.09	366	47.05
EfficientDet	18.08	0.37	18.44	6.55	17.30	27.92	28.01	28.05	28.26	28.34	28.07	27.79	27.31	25.07	26.61	54	18.72
DAM	11.67	0.84	12.51	65.36	38.42	43.58	43.63	43.33	42.87	42.09	41.33	39.90	37.52	31.43	40.41	79	27.56
CBAM	14.97	0.85	15.82	64.11	48.33	50.30	50.27	50.35	50.15	49.36	49.57	48.64	47.80	45.19	48.99	62	30.62
FRS-Net	1.71	0.65	2.36	5.04	57.81	50.75	50.69	50.61	42.73	42.71	42.50	34.40	34.11	25.70	43.20	423	51.60

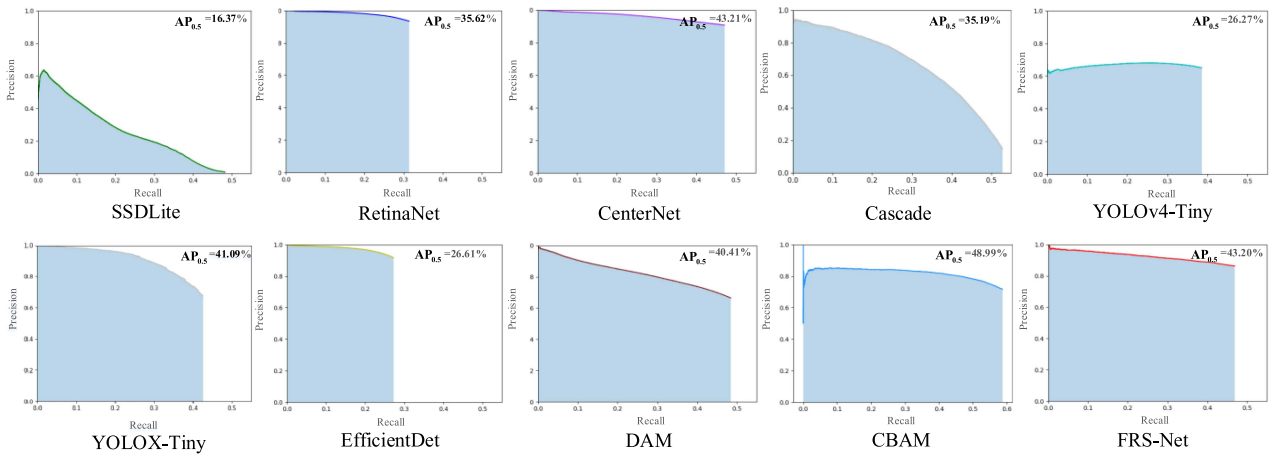

Fig. 7. PR curves of different detection algorithms on SDIOR.

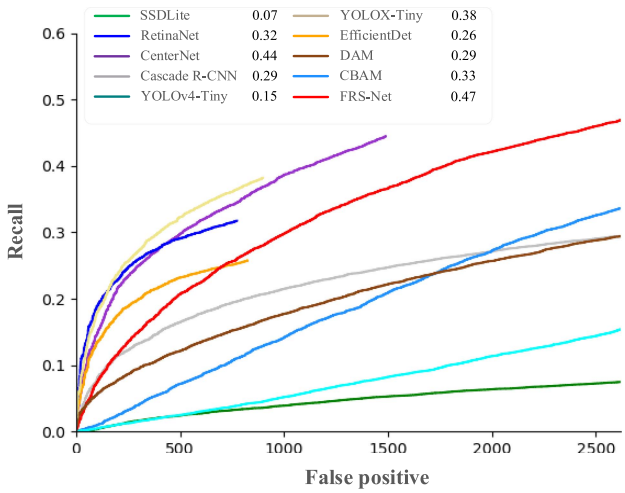

Fig. 8. ROC curves of different detection algorithms on SDIOR.

Compared with the EfficientDet, the $mAP_{0.5}$ of the FRS-Net is increased by 16.59% (26.61% versus 43.20%), the FPS is increased by 369 (54 versus 423), and the SP is increased by 32.88 (18.72 versus 51.60).

Compared with the DAM, the $mAP_{0.5}$ of the FRS-Net is increased by 2.79% (40.41% versus 43.20%), the FPS is increased by 344 (79 versus 423), and the SP is increased by 24.04 (27.56 versus 51.60).

Compared with the CBAM, $mAP_{0.5}$ of the FRS-Net is decreased by 5.79% (48.99% versus 43.20%), FPS is increased by 361 (62 versus 423), and SP is increased by 20.98 (30.62 versus 51.60).

In Tables IV and V, it can be clearly seen that CenterNet and CBAM with complex networks have higher detection accuracy than other networks under different IoU settings, but the detection speed is slower. Meanwhile, YOLOv4-Tiny and SSDLite have faster detection speed but lack performance in detection accuracy. FRS-Net balances the two well and has good comprehensive performance.

Figs. 7 and 8 show the PR curves and ROC curves of all algorithms at IoU = 0.5. It is worth noting that the area enclosed by the PR curves of each algorithm in Fig. 7 represents the $AP_{0.5}$ of the algorithm, and the ROC curve of each algorithm in Fig. 8 represents the recall value corresponding to the false positive value of the algorithm [61]. Some curves are shorter than others; this is due to the algorithm's inability to generate more predicted boxes above the confidence level of 0.5. It can be seen from the figures that FRS-Net is close to optimal in terms of target detection accuracy.

Fig. 9 shows the overall performance of the FRS-Net target detection algorithm and other algorithms. In Fig. 9, the horizontal axis represents different network detection models, and

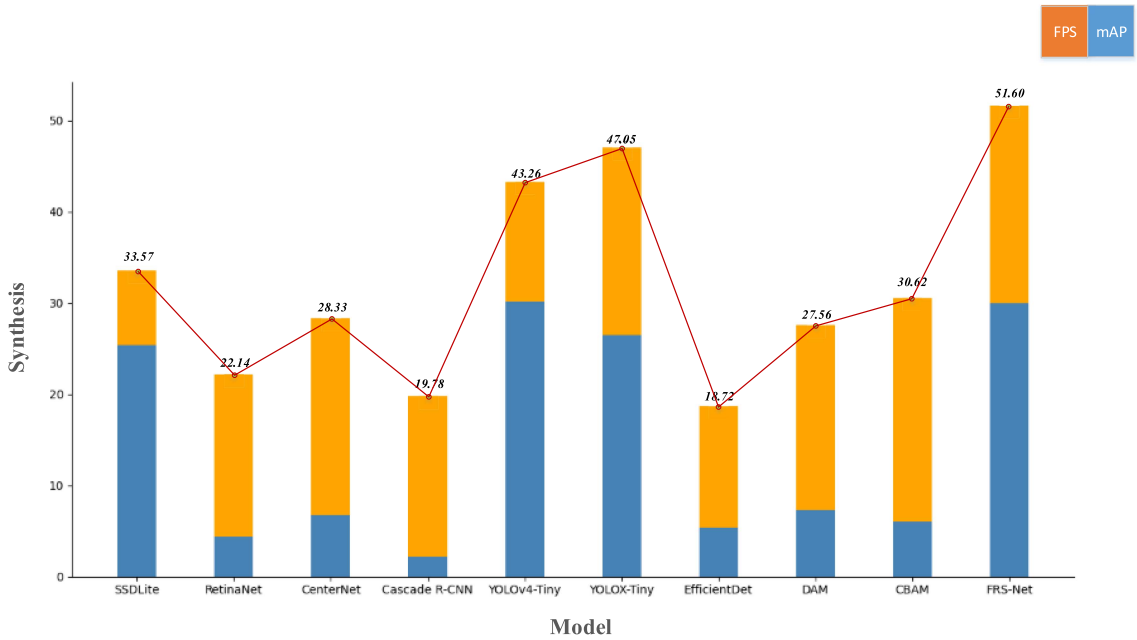

Fig. 9. Accuracy, efficiency, and SP comparison of different detection algorithms on SDIOR.

TABLE V
COMPARISON WITH OTHER STATE-OF-THE-ART MODELS AND WORKS ON DIFFERENT IOU SETTINGS

Model	Latency(ms)			Parameter (M)	mAP with different IoU					FPS	SP
	Net	Anchor	Total		$mAP_{0.5}$	$mAP_{0.75}$	mAP_S	mAP_M	mAP_L		
SSDLite	1.54	1.35	2.89	3.68	16.37	0.88	14.64	14.64	0.42	348	33.57
RetinaNet	23.28	0.54	23.81	36.33	35.62	22.52	7.43	70.33	13.35	41	22.14
CenterNet	11.12	2.78	13.90	32.66	43.21	36.49	30.42	61.89	12.85	71	28.33
Cascade R-CNN	33.86	23.36	57.22	69.10	35.19	25.17	20.45	49.91	6.23	17	19.78
YOLOv4-Tiny	1.65	0.70	2.35	6.53	26.27	21.38	22.19	26.07	4.85	425	43.26
YOLOX-Tiny	2.37	0.36	2.73	5.03	41.09	29.5	16.69	68.21	17.11	366	47.05
EfficientDet	18.08	0.37	18.44	6.55	26.61	18.6	8.16	55.07	15.26	54	18.72
DAM	11.67	0.84	12.51	65.36	40.41	32.5	29.43	47.87	6.23	79	27.56
CBAM	14.97	0.85	15.82	64.11	48.99	44.42	41.28	57.3	9.68	62	30.62
FRS-Net	1.71	0.65	2.36	5.04	43.20	33.1	33.92	44.78	5.93	423	51.60

the vertical axis represents the indicator SP used to measure the comprehensive performance of the network model. Among them, the comprehensive indicator SP consists of two parts, which are the $AP_{0.5}$ (%) and the FPS.

The conclusion that can be drawn from Tables IV and V and Figs. 7–9 is that in the remote sensing ship detection task covered by thin clouds and fog, the FRS-Net algorithm is superior to other algorithms in SP. Even though YOLOv4 outperforms FRS-Net in speed, it lacks accuracy; CenterNet and CBAM outperform FRS-Net in accuracy, but they lack detection speed.

In Fig. 10, the red rectangles are used to mark all missed and falsely detected ships. The first row represents the detection results of each algorithm in a clear environment. It can be seen that the SSDLite, DAM, and EfficientDet algorithms have many missed detections. There are many false detections in YOLOv4-Tiny. And FRS-Net, CenterNet, CBAM, and YOLOX-Tiny could detect all ships.

The second, third, and fourth rows represent fog environments with different thicknesses. All algorithms have missed detections. Compared with other algorithms, YOLOX-Tiny, CBAM, and FRS-Net have fewer missed detections and stronger antifog capabilities.

In Fig. 11, the red solid/dashed rectangles are used to mark the different detection results in the same area, further demonstrating the detection ability of each algorithm under dense distribution conditions. In the first row, comparing the dashed box area with the solid box area, EfficientDet, RetinaNet, and YOLOX-Tiny have many missed detections, and SSDLite, Cascade, YOLOv4-Tiny, and DAM have many false detections, whereas FRS-Net can accurately identify most targets. In the second, third, and fourth rows, FRS-Net also performs better in fog environments with different thicknesses.

Comparing the detection results of the above-mentioned algorithms in different scenarios, it can be concluded that FRS-Net can perform similarly to other advanced algorithms in remote sensing images without thin clouds and fog and has better performance than others in the environment covered by thin clouds and fog. Considering that it also has good execution efficiency, the comprehensive performance of FRS-Net is outstanding.

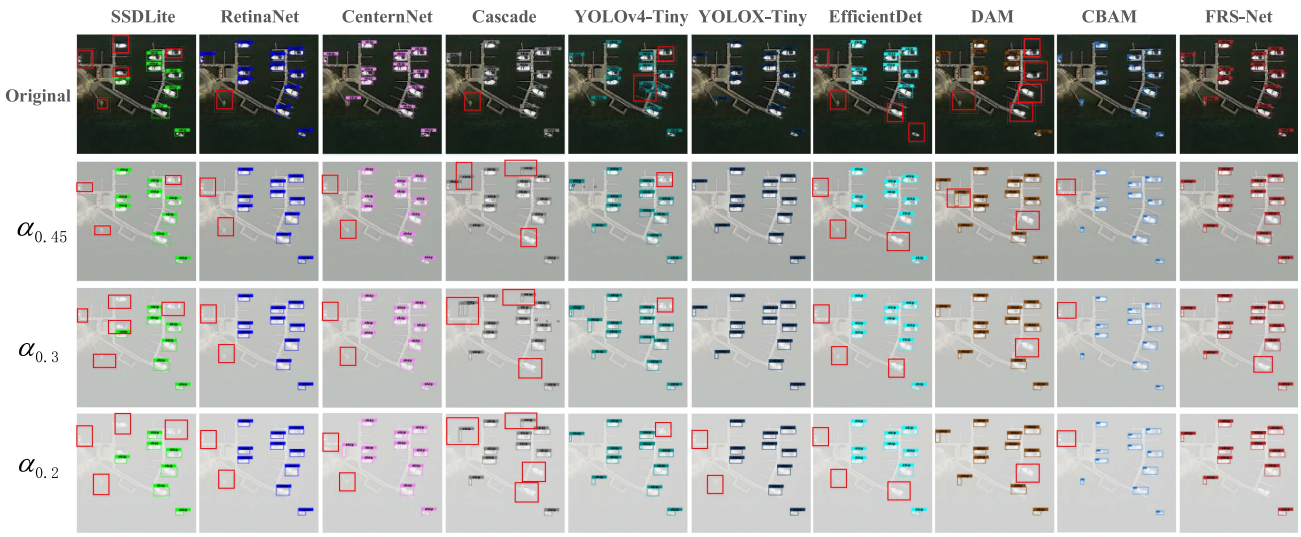

Fig. 10. Comparison of detection results of different algorithms with different fog thicknesses.

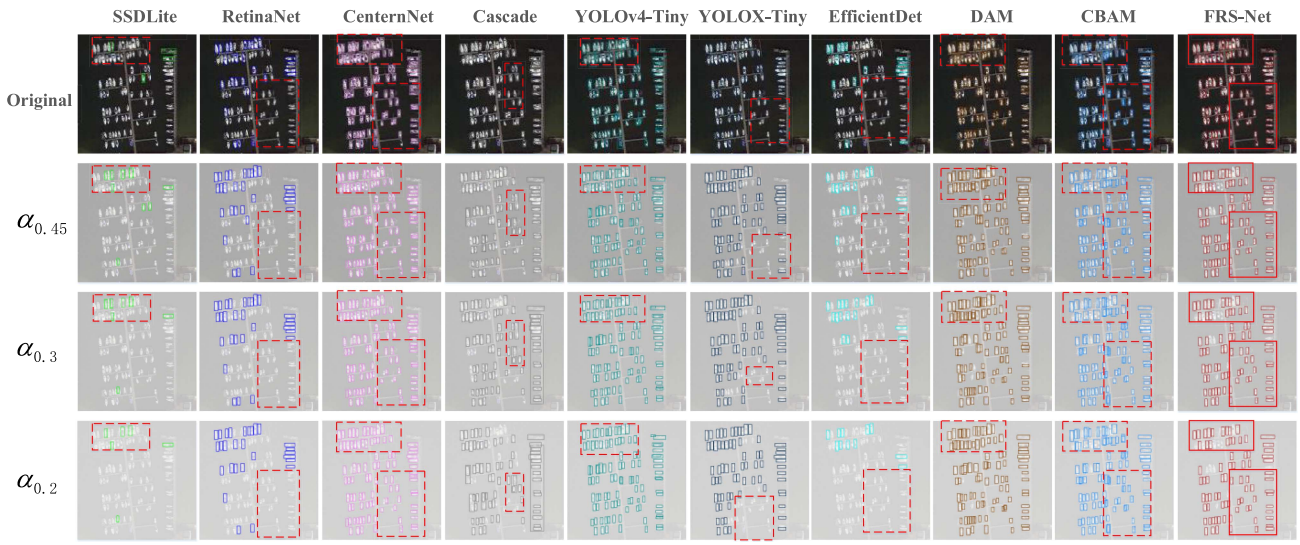

Fig. 11. Comparison of detection results of different algorithms under dense distribution conditions with different fog thicknesses.

VI. Discussion

In the traditional remote sensing data application mode, it is necessary to obtain high-quality image products for specific application scenarios through steps such as satellite data acquisition, raw data download, ground data processing and selection, and product distribution. Due to the complex processing steps, the timeliness is low, and it is difficult to meet the needs of observation, tracking, and rescue of moving targets, such as ships, through satellite data. In recent years, the research study on onboard applications has gradually become a hot topic. The main constraints faced by onboard applications include the following. 1) Due to the requirements of volume, weight, power consumption, and radiation resistance, the computability of the onboard computing device is insufficient; and 2) The relative position relationship between the satellite and ground station changes rapidly, accordingly, the satellite-to-ground data transmission window is very short, and the onboard computing must be completed within a short time after data acquisition, which requires high timeliness. Furthermore, for optical satellites, unlike high-quality image products on the ground, the data flowing into the onboard device is unselected first-hand image, generally with high cloud coverage. In this case, traditional methods rarely consider timeliness and generally treat interference such as clouds and fog as noise, that is, seeking to remove them first and then implementing various applications. However, this strategy not only increases the amount of extra computation but may also damage the features of small-scale targets. Relatively speaking, direct detection algorithms with stronger anti-interference ability are more important for high time-sensitive applications. For similar reasons, considering timeliness requirements, we choose the object detection algorithm over the instance segmentation algorithm.

Therefore, in order to meet the needs of future onboard applications, we design an efficient ship detection approach named FRS-Net from the perspective of balancing detection efficiency and accuracy while considering the adaptability to thin-cloud and fog coverage conditions. The experimental results show that the approach can well balance detection efficiency and accuracy in the field of ship detection in complex environments. In FRS-Net, the anchor box setting is optimized according to the shape characteristics of the ship target, which helps to balance the detection efficiency and accuracy well; the feature fusion network enhances the feature extraction ability of the model under the interference of thin clouds and fog; and the prediction network reduces the interference of thin clouds and fog on the ship detection ability under dense distribution. When the method needs to be applied to other domains, such as vehicles or airplanes, it is only necessary to adjust the anchor box setting and prediction network according to the shape and scale characteristics of the target.

However, it should be noted that compared with the cloud coverage of different degrees simulated in this manuscript, the actual optical image cloud coverage is more complicated. On the one hand, the natural cloud itself is not uniform and has some texture. On the other hand, the occlusion of the ship by clouds and fog is also more complicated. It is predictable that compared with simulated clouds and fog, the detection accuracy of algorithms will be further reduced in the real cloud and fog occlusion scene. To make the algorithm more suitable for the target of this scale in the future actual onboard application adaptation process, first, it is necessary to further optimize the hyperparameter settings of the algorithm according to the spatial resolution of the specific satellite. Then, the algorithm needs to be fine-tuned for actual complex cloud and fog coverage data to reduce the accuracy loss caused by complex interference. In addition, the algorithm needs to be deeply optimized by combining software and hardware on the onboard computing device to maximize the timeliness of the algorithm and cooperate with the onboard high-performance architecture to balance the onboard data flow-in rate and the processing speed to achieve real-time processing [18].

In terms of the actual processing efficiency in the future onboard environment, it should be noted that due to the large size of remote sensing images actually obtained by satellites, they need to be cut into small pieces for processing in practical applications. And in order to minimize the loss of information introduced in the cutting and splicing process, it is also necessary to leave a certain overlapping area between the slices. Even so, there is still a certain loss of efficiency and accuracy overall. It can be considered that the smaller the slice, the more serious the loss of efficiency and accuracy. Fortunately, efficient networks usually have faster inference speed and smaller parameters than larger networks, which means that under the same hardware conditions, efficient networks can infer faster while processing larger slice images, which can partially reduce the adverse effects caused by fragmented slicing. This advantage makes efficient networks more suitable for future practical onboard applications than larger networks. With the vigorous development of commercial aerospace, the trend of remote sensing satellites developing toward miniaturization, agility, and intelligence has become more and more obvious. In recent years, Chinese satellites, including the GF series, BJ series, Luojia series, etc., have begun to try to carry onboard computing devices to meet future needs. Based on the above-mentioned considerations, the approach presented in this manuscript has good potential for onboard applications, and we will carry out further work with the goal of actually deploying the approach on the Luojia3 satellite owned by Wuhan University in near future.

VII. Conclusion

This manuscript proposes a deep-learning-based ship detection method, FRS-Net, with higher detection accuracy, robustness, and timeliness to meet the application requirements of efficient ship detection under thin-cloud and fog coverage conditions. First, considering the characteristics of ship shape in remote sensing images, we propose that type and size of anchor boxes be optimized, which reduces the complexity and improves the performance of the algorithm. Second, the FRS-Net network is constructed, which is mainly composed of a backbone extraction network (CSPDarkNet53-Tiny), a feature fusion network, and a prediction network. Among them, the feature fusion network can fuse high-level semantic information with low-level semantic information so that the network can better extract and locate ship targets in the thin-cloud and fog-covered environment. On this basis, the prediction network selects an appropriate prediction scale, which can alleviate the problem of decreased detection accuracy of dense ship distribution in a thin-cloud and fog-covered environment. After experimental verification on the SDIOR simulation dataset, FRS-Net maintains a certain detection ability in the fog remote sensing ship task and has a fast inference speed. Compared with state-of-the-art methods, it can better meet the needs of practical application scenarios.

However, it is also necessary to realize that since FRS-Net has only two output feature scales, it also partially sacrifices the ability to adapt to more scale target detection while having a fast anchor box inference process. Additionally, limited by the lightweight backbone extraction network, there is still room for improvement in the accuracy of target extraction. In application scenarios that do not require high timeliness, the backbone can be replaced with a network model with stronger extraction capabilities. Our work can serve the application requirements of future onboard intelligent processing, and the timeliness of the method and the performance under low resource occupancy are considered. In the future, after the launch of the Luojia3 experimental satellite, FRS-Net will be further transplanted, verified, tested, and adjusted based on the real software and hardware environment and data.

Acknowledgment

Disclaimer: The authors would like to thank the anonymous reviewers and members of the editorial team for their comments and suggestions.

References

[1] N. Wawrzyniak, T. Hyla, and A. Popik, "Vessel detection and tracking method based on video surveillance," *Sensors*, vol. 19, 2019, Art. no. 5230.

[2] S. Zhang, R. Wu, K. Xu, J. Wang, and W. Sun, "R-CNN-based ship detection from high resolution remote sensing imagery," *Remote Sens.*, vol. 11, 2019, Art. no. 631.

[3] K. Zhang and H. Shen, "Multi-stage feature enhancement pyramid network for detecting objects in optical remote sensing images," *Remote Sens.*, vol. 14, 2022, Art. no. 579.

[4] W. Zhou and L. Liu, "An efficient ship detection and classification algorithm based on YOLOv4," *Int. Core J. Eng.*, vol. 8, pp. 163–173, 2022.

[5] H. Zou, S. He, Y. Wang, R. Li, F. Cheng, and X. Cao, "Ship detection based on medium-low resolution remote sensing data and super-resolved feature representation," *Remote Sens. Lett.*, vol. 13, pp. 323–333, 2022.

[6] D. Cui, L. Guo, and Y. Zhang, "Research on the development of ship target detection based on deep learning technology," in *Proc. Int. Conf. Frontier Comput.*, 2022, vol. 827, pp. 1730–1736.

[7] Z. Zakria, J. Deng, R. Kumar, M. S. Khokhar, J. Cai, and J. Kumar, "Multiscale and direction target detecting in remote sensing images via modified YOLO-v4," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 1039–1048, 2022.

[8] S. Torre, C. Battini, and M. Burlando, "A new monitoring system of the wind actions on moored ships," in *Trends Maritime Technology Engineering*, vol. 1. Boca Raton, FL, USA: CRC Press, 2022, pp. 469–474.

[9] I. T. Joseph S, V. Karunakaran, T. Sujatha, S. B. Edwin Rai, and S. Velliangiri, "Investigation of deep learning methodologies in satellite image based ship detection," in *Proc. Int. Conf. Sustain. Comput. Data Commun. Syst.*, 2022, pp. 1734–1739.

[10] J. Wu, J. Li, and R. Li, "A fast maritime target identification algorithm for offshore ship detection," *Appl. Sci.*, vol. 12, 2022, Art. no. 4938.

[11] D. Li, X. Shen, J. Gong, J. Zhang, and J. Lu, "On construction of China's space information network," *Geomatics Inf. Sci. Wuhan Univ.*, vol. 40, pp. 711–715, 2015.

[12] D. R. Li, "Towards geo-spatial information science in big data era," *Acta Geodaetica Cartographica Sinica*, vol. 45, pp. 379–384, 2016.

[13] C. O. Davis, D. M. Horan, and M. R. Corson, "On-orbit calibration of the naval EarthMap observer (NEMO) coastal ocean imaging spectrometer (COIS)," in *Proc. Int. Symp. Optical Sci. Tech.*, San Diego, CA, USA, 30 Jul.4 Aug. 2000.

[14] S. J. Visser and A. S. Dawood, "Real-time natural disasters detection and monitoring from smart earth observation satellite," *J. Aerosp. Eng.*, vol. 17, pp. 10–19, 2004.

[15] A. G. Schmidt, G. Weisz, M. French, T. Flatley, and C. Y. Villalpando, "SpaceCubeX: A framework for evaluating hybrid multi-core CPU/FPGA/DSP architectures," in *Proc. IEEE Aerosp. Conf.*, 2017, pp. 1–10.

[16] G. Zhou, R. Zhang, N. Liu, J. Huang, and X. Zhou, "On-board orthorectification for images based on an FPGA," *Remote Sens.*, vol. 9, 2017, Art. no. 874.

[17] W. Mi, Z. Zhiqi, Z. Ying, D. Zhipeng, and L. Yingying, "Embedded GPU implementation of sensor correction for on-board real-time stream computing of high-resolution optical satellite imagery," *J. Real-Time Image Process.*, vol. 13, pp. 565–581, 2018.

[18] Z. Zhang et al., "Expandable on-board real-time edge computing architecture for Luojia3 intelligent remote sensing satellite," *Remote Sens.*, vol. 14, no. 15, 2022, Art. no. 3596.

[19] M. Lyu, L. Han, and S. Tian, "Cloud detection under varied surfaces and atmospheric conditions with MODIS imagery," *J. Remote Sens.*, vol. 20, no. 6, pp. 1371–1380, 2016.

[20] Climatology of Global Cloud and Surface Properties. Dec. 2010, Accessed: Mar. 28, 2020. [Online]. Available: https://isccp.giss.nasa.gov/analysis/climanal1.html

[21] S. Huang, D. Li, W. Zhao, and Y. Liu, "Haze removal algorithm for optical remote sensing image based on multi-scale model and histogram characteristic," *IEEE Access*, vol. 7, pp. 104179–104196, 2019.

[22] A. Makarau, R. Richter, R. Müller, and P. Reinartz, "Haze detection and removal in remotely sensed multispectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 9, pp. 5895–5905, Sep. 2014.

[23] Q. Liu, "Haze removal for a single visible remote sensing image," *Signal Process.*, vol. 137, pp. 33–43, 2017.

[24] J. Lv and D. Liu, "Ship target detection based on adverse meteorological conditions," in *Proc. IEEE Asia-Pacific Conf. Image Process., Electron. Comput.*, 2022, pp. 480–484.

[25] X. Qin, Z. Wang, and Y. Bai, "FFA-net: Feature fusion attention network for single image dehazing," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, pp. 11908–11915, 2020.

[26] W. Pikun, W. Ling, and Q. Jiangxin, "Unmanned aerial vehicles object detection based on image haze removal under sea fog conditions," *IET Image Process.*, vol. 16, pp. 2709–2721, 2022.

[27] T. Shuai, K. Sun, X. Wu, X. Zhang, and B. Shi, "A ship target automatic detection method for high-resolution remote sensing," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2016, pp. 1258–1261.

[28] S. Qi, J. Ma, J. Lin, Y. Li, and J. Tian, "Unsupervised ship detection based on saliency and S-HOG descriptor from optical satellite images," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 7, pp. 1451–1455, Jul. 2015.

[29] F. Yang, Q. Xu, F. Gao, and L. Hu, "Ship detection from optical satellite images based on visual search mechanism," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2015, pp. 3679–3682.

[30] T. Yulin, J. Shaohua, B. Gang, Z. Yonzhou, and L. Fan, "Wreckage target recognition in side-scan sonar images based on an improved faster R-CNN model," in *Proc. Int. Conf. Big Data Artif. Intell. Softw. Eng.*, 2020, pp. 348–354.

[31] J. Tang, C. Deng, G.-B. Huang, and B. Zhao, "Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1174–1185, Mar. 2015.

[32] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[33] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[34] 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.

[35] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 1823, 2018, pp. 6154–6162.

[36] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[37] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6517–6525.

[38] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Apr. 2018, *arXiv:1804.02767*. [Online]. Available: https://arxiv.org/abs/1804.02767

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

[40] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[41] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6569–6578.

[42] T. Lin et al., "Microsoft COCO: Common objects in context," May 2014, *arXiv:1405.0312*. [Online]. Available: https://arxiv.org/abs/1405.0312

[43] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, 2010.

[44] J.-B. Hou, X. Zhu, and X.-C. Yin, "Self-adaptive aspect ratio anchor for oriented object detection in remote sensing images," *Remote Sens.*, vol. 13, no. 7, p. 1318, Mar. 2022.

[45] X. Song, *Research on Ship Target Recognition Algorithm in Complex Background[D]*, Harbin Eng. Univ., Harbin, China, 2020, doi: 10.27060/d.cnki.ghbcu.2020.000501.

[46] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognit.*, vol. 36, pp. 451–461, 2003.

[47] 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. Photogramm. Remote Sens.*, vol. 159, pp. 296–307, 2020.

[48] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, Dec. 2011.

[49] S. Zhang, R. Wu, K. Xu, J. Wang, and W. Sun, "R-CNN-based ship detection from high resolution remote sensing imagery," *Remote Sens.*, vol. 11, 2019, Art. no. 631, doi: 10.3390/rs11060631.

[50] H. Guo, H. Bai, Y. Yuan, and W. Qin, "Fully deformable convolutional network for ship detection in remote sensing imagery," *Remote Sens.*, vol. 14, 2022, Art. no. 1850, doi: 10.3390/rs14081850.

[51] H. Lin, Z. Shi, and Z. Zou, "Fully convolutional network with task partitioning for inshore ship detection in optical remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1665–1669, Oct. 2017.
[52] Y. You, J. Cao, Y. Zhang, F. Liu, and W. Zhou, "Nearshore ship detection on high-resolution remote sensing image via scene-mask R-CNN," *IEEE Access*, vol. 7, pp. 128431–128444, 2019.
[53] L. Chen, W. Shi, C. Fan, L. Zou, and D. Deng, "A novel coarse-to-fine method of ship detection in optical remote sensing images based on a deep residual dense network," *Remote Sens.*, vol. 12, 2020, Art. no. 3115.
[54] Z. Ren, Y. Tang, Z. He, L. Tian, Y. Yang, and W. Zhang, "Ship detection in high-resolution optical remote sensing images aided by saliency information," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5623616.
[55] W. Fei, L. Meng, L. Xueqin, Q. Zhiliang, M. Benjun, and T. Yi, "Real-time detection of marine vessels under sea fog weather conditions based on YOLOv3 deep learning[J]," *Marine Sci.*, vol. 44, no. 8, pp. 197–204, 2020.
[56] L. Chen, W. Shi, and D. Deng, "Improved YOLOv3 based on attention mechanism for fast and accurate ship detection in optical remote sensing images," *Remote Sens.*, vol. 13, 2021, Art. no. 660.
[57] M. Shao et al., "Branch aware assignment for object detection[J]," *Vis. Comput.*, pp. 1–10, 2022.
[58] Y. Gao et al., "Decoupled IoU regression for object detection," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 5628–5636.
[59] B. Cheng, R. Girshick, P. Dollár, A. C. Berg, and A. Kirillov, "Boundary IoU: Improving object-centric image segmentation evaluation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 15334–15342.
[60] H. Fu, G. Song, and Y. Wang, "Improved YOLOv4 marine target detection combined with CBAM," *Symmetry*, vol. 13, p. 623, 2021, doi: 10.3390/sym13040623.
[61] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5525–5533.

Zhiqi Zhang received the B.Sc. degree in geographic information system from Huazhong Agricultural University, Wuhan, China, in 2006, the B.Eng. degree in computer science and technology from the Huazhong University of Science and Technology, Wuhan, China, in 2006, and the M.Eng. degree in computer technology and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2015 and 2018, respectively.

He is currently an Associate Professor with the School of Computer Science, Hubei University of Technology, Wuhan, China. His research interests include system architecture, algorithm optimization, artificial intelligence, and high-performance processing of remote sensing.

Huigang Zheng received the B.Sc. degree in computer science and technology from Wuhan Qingchuan University, Wuhan, China, in 2020.

He is currently a graduate student with the School of Computer Science, Hubei University of Technology, Wuhan, China. His research interests include high spatial resolution and hyperspectral remote sensing image processing and analysis.

Jinshan Cao received the Ph.D. degree in photogrammetry and remote sensing from the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China, in 2012.

He is currently an Associate Professor with the School of Computer Science, Hubei University of Technology, Wuhan, China. His research interests include geometric calibration, sensor orientation, and image registration of high-resolution satellite imagery.

Xiaoxiao Feng received the B.Sc. degree in surveying and mapping from Southeast University, Nanjing, China, in 2014, the M.Sc. degree in earth exploration and information technology from the China University of Geology, Wuhan, China, in 2017, and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2021.

She is currently a Lecturer with the School of Computer Science, Hubei University of Technology, Wuhan, China.

Guangqi Xie received the M.Sc. degree in remote sensing of resources and environment from the China University of Geosciences, Wuhan, China, in 2018. He is currently working toward the Ph.D. degree in photogrammetry and remote sensing from the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan, China.

His research interests include image matching and registration, pansharpening, and image superresolution.