OPD-Net: Prow Detection Based on Feature Enhancement and Improved Regression Model in Optical Remote Sensing Imagery

Yanan You, *Member, IEEE*, Bohao Ran, Gang Meng, Zezhong Li, Fang Liu, and Zhixin Li

***Abstract*— Accurate prow detection (i.e., ship heading prediction) is important in many applications that rely on optical remote sensing imagery, such as track forecasting and maritime navigation. In recent years, many advanced methods based on deep convolution neural networks (DCNNs) have succeeded in detecting multidirectional ships. However, these methods are not effective at determining the prow orientation, primarily due to three limitations: weak adaptability to geometric transformations of ship targets, confusing the semantic information between the prow and other parts of ships, and the boundary discontinuity problem. To address these problems, we propose an omnidirectional prow detection network (OPD-Net) based on feature enhancement and an improved regression model. OPD-Net consists of a feature refinement network (FRN), a prow attention network (PAN), and a complex plane coordinates regression model (CPCRM). First, the FRN balances the low-level location information and high-level semantic information from multiscale feature maps and then fits various geometric transformations regarding ship targets through deformable blocks. Next, the PAN, which is based on supervised learning, is used to enhance the ship prow feature as well as suppress background noise, which improves the accuracy of ship heading predictions. Finally, the CPCRM is designed to effectively solve the boundary discontinuity problem and correctly achieve prow detection in arbitrary orientations. Experiments on optical remote sensing image data sets demonstrate the robustness and superiority of our method for prow detection. Moreover, our approach is also competitive when used only for ship detection.**

***Index Terms*— Deep learning, feature enhancement, prow detection, regression model, remote sensing.**

I. Introduction

With the rapid advancement of remote sensing technology, optical remote sensing images [1] are now capable of accurately revealing and locating surface entities. Ship detection plays a major role in many optical remote sensing image applications, such as maritime navigation, cargo management, and military reconnaissance. Meanwhile, information about a ship's prow direction (i.e., its heading) is also highly important in applications such as course prediction and tracking.

In recent years, object detection methods based on deep convolutional neural networks (DCNNs) [2] have become an active research target. DCNN-based detectors, which have powerful feature extraction and representation abilities, can achieve automatic location and classification of targets. These detectors can be categorized into one- and two-stage methods. One-stage methods, such as you only look once (YOLO) [3] and the single shot multibox detector (SSD) [4], directly estimate object position from each divided grid in images. Therefore, one-stage methods are fast and efficient, enabling true real-time detection. Based on the region proposal algorithm, two-stage methods [5]–[8] first extract candidate objects in the input image and then perform classification and regression for each candidate. The two-stage methods consume more time than the one-stage methods, but they also achieve higher detection accuracies than do the one-stage methods.

Although many state-of-the-art detectors have obtained promising results on various natural image data sets [9]–[11], they still face challenges when applied to remote sensing images from conditions such as near-shore false alarms, large detection ranges, and multiscale objects. In response to these challenges, various methods have been proposed in recent years. Scene-Mask R-CNN [12] generates a land–sea mask during detection, thus suppressing most onshore false alarms. You *et al.* [13] proposed a complete processing system called the broad-area target search (BATS) system, which performs target detection in complicated scenes. To efficiently detect objects with different scales, HSF-net [14] uses a hierarchical selective filtering layer designed to map features of different scales to the same scale space. However, these methods have difficulty achieving satisfactory detection performances on multioriented ship targets due to the limitations of horizontal bounding boxes. For inclined ships, large redundant regions exist in the boxes, resulting in misalignments between the boxes and the targets. Moreover, the operation of the nonmaximum suppression (NMS) algorithm often causes missed detections when ships are docked closely.

Manuscript received March 5, 2020; revised May 27, 2020 and July 24, 2020; accepted July 28, 2020. Date of publication August 19, 2020; date of current version June 24, 2021. This work was supported in part by the Ministry of Education and China Mobile Communications Corporation (MoE-CMCC) Artificial Intelligence Project (MCM20190701), the National Key R&D Program of China under Grant 2019YFF0303300 and under Subject II No. 2019YFF0303302. *(Corresponding author: Bohao Ran.)*

Yanan You, Bohao Ran, Zezhong Li, and Fang Liu are with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: youyanan@bupt.edu.cn; bohaor@163.com; lizezhong@bupt.edu.cn; lindaliu@bupt.edu.cn).

Gang Meng and Zhixin Li are with the Beijing Institute of Remote Sensing Information, Beijing 100192, China (e-mail: menggangmark@126.com; li_zhixin@sina.com).

Digital Object Identifier 10.1109/TGRS.2020.3014195

Thus far, some detectors based on rotated bounding boxes have been applied to ship detection, such as the rotated region proposal network (R^2PN) [15], R^2CNN++ [16], and the rotation dense feature pyramid network (R-DFPN) [17]. Rotated bounding boxes introduce an angle attribute, allowing them to more closely surround targets and effectively solving the missed detections caused by NMS. However, the abovementioned methods neglect information about ship headings; the predicted angles in these methods represent only the inclined angle (less than 180°) of the rotated bounding box but not a specific prow orientation. Therefore, these methods cannot be effectively employed in applications that require knowledge of a ship's heading direction, such as track monitoring and prediction.

The prow prediction task is more difficult than ship detection because it determines which way a ship's head is pointed based on accurately target location and classification. Yang *et al.* [18] adopted a simple but effective method in which a new fully connected layer is imported to determine which side of the predicted box is the prow. However, this method is still limited by three shortcomings. The first is that it has weak adaptability to various geometric transformations of ship targets. In reality, the aspect ratios and sizes of ships vary greatly. In addition, even the same ship can present diverse orientations and scales because of different viewpoints and image resolutions. These factors lead to complicated geometric transformations of ships. However, in most DCNNs, which are based on standard convolution operations, adaptability to geometric transformations depends almost entirely on the results of artificial data augmentation. The second factor is that methods semantically confused prows with other parts of ships. The third problem is the boundary discontinuity problem, which is common in detectors that use rotated bounding boxes. This problem occurs because the angles of predicted boxes tend to deviate when the target orientation approaches an angular boundary. This problem derives from the angle-based regression model (ARM) in these detectors. Because the angle is periodic but the angular boundary is discontinuous, the regression loss cannot converge near the boundary, leading to inaccurate detection results. Consequently, these detectors are limited by the boundary discontinuity problem, and they cannot accurately detect ships heading in arbitrary directions.

To address the aforementioned issues for prow detection (i.e., ship heading determination), an omnidirectional prow detection network (OPD-Net) based on feature enhancement and an improved regression model is proposed in this article. This novel detection framework consists of three main parts: a feature refinement network (FRN), a prow attention network (PAN), and a complex plane coordinates regression model (CPCRM). First, the FRN integrates multilevel features from the feature pyramid to balance the location and semantics of targets; then, deformable convolution is used to perform geometric transformation modeling. By adding 2-D offsets to a regular sampling grid in a standard convolution, the deformable convolution learns augmented spatial sampling locations from the destination tasks without additional supervision. The feature extracted by the FRN contains more information about the various geometric transformations that targets may exhibit, which helps to improve the ship detection performance. Second, using a supervised attention algorithm, the PAN enhances the prow (i.e., the front part of a ship) feature while weakening background noise, which improves the prow detection accuracy and reduces false alarms. Finally, the CPCRM is designed to solve the boundary discontinuity problem: it directly predicts a ship's heading without requiring complicated computation. Rather than using angles, the coordinates in a complex plane applied in this regression model represent the prow direction. Regardless of how the prow direction varies in images, the coordinates generated by the CPCRM are always unique and continuous, thus it avoids the boundary discontinuity problem. The main contributions of this article are summarized as follows.

1) An FRN is proposed in OPD-Net that balances multilevel features and fits the various possible geometric transformations of ship targets.
2) We use the PAN in OPD-Net to enhance the prow feature and reduce the background noise, which improves the prow detection accuracy.
3) We design a novel regression model based on the coordinates in a complex plane, which solves the boundary discontinuity problem and achieves omnidirectional prow detection.
4) Comparative experiments conducted on a self-collected optical remote sensing data set and HRSC2016 data set [19] demonstrate that our approach outperforms other DCNN-based algorithms at prow detection. Furthermore, the results also show that our method is competitive with the state-of-the-art detectors even when used only for the ship detection task.

The remainder of this article is organized as follows. The related works are briefly introduced in Section II. Section III presents the details of the proposed method, including the FRN, PAN, and CPCRM models. The results of experiments on remote sensing images are reported in Section IV, and Section V provides conclusions.

II. Related Works

A. Multioriented Target Detection Networks

Detecting targets with multiple orientations and large aspect ratios, such as texts, vehicles, and ships, is still a challenge in the field of objection detection. Designed for horizontal or nearly horizontal generic object detection, the performance of classical DCNN-based detection methods usually degrades when applied to multioriented target detection. Aiming at this problem, some networks based on rotated bounding boxes have been proposed in recent years. Based on Faster R-CNN [8], the rotational region CNN (R^2CNN) [20] imports a new branch in the Fast R-CNN stage to predict inclined bounding boxes. Inclined NMS (INMS) is adopted as a postprocessing method that avoids missed detections of closely adjacent inclined targets caused by conventional NMS. On the other hand, R^2CNN still adopts horizontal proposals in the RPN stage, resulting in redundant regions for inclined targets, and it further induces inaccurate feature extraction by region of interest (RoI) pooling. To reduce the redundancy among region

proposals, R^{2}PN [15] uses rotated anchors with different angles, which surround target more accurately, to generate oriented proposals. Then, rotated RoI (RRoI) pooling [15] was proposed to extract the discriminative features with fewer redundant regions from such oriented proposals. In R-DFPN [17], a multiscale RoI align [21] layer is used to maintain the completeness of the information in the extracted features. In this method, two pooling sizes (3:16 and 16:3) are added to minimize the effects of the deformations caused by the interpolation methods. R^{2}CNN++ [16] proposed a multidimensional attention (MDA) module consisting of a pixel attention network and a channel attention network. This architecture weakens background interference and highlights the features of targets. It is worth noting that the predicted angles in the above methods represent only the inclined angle of the rotated bounding box in the images—not the ship heading direction. Consequently, these methods are not effective for determining the prow direction.

B. Prow Detection Methods

The direction in which a ship's prow is pointing is an important aspect of ship detection that contributes to track forecasting, maritime navigation, and other maritime applications. Some algorithms for machine learning and common digital image processing can be used to obtain the headings of ships. Li *et al.* [22] generated prow features in the transformed domain of polar coordinates, in which prows have an approximately trapezoidal shape, allowing them to be more easily detected. Then, these features are used in a classification operation based on a support vector machine (SVM) to detect prow candidates, and their directions are provided. In [23], line segments are utilized to detect the prow and find the main axis near the prow, which then serves as the pointing direction to deal with the rotation invariant problem. In contrast, Liu *et al.* [24] detected the prow from binary images to avoid disturbances in intensity diversity. However, these methods require accurate ship head contour extraction; thus, they are not appropriate for targets without obvious V-shaped prows or those that appear in complex backgrounds. In addition, these methods have shortcomings such as inefficiency and poor generalization. Compared with conventional methods, DCNN-based methods have the advantages of higher efficiency and good scalability. Yang *et al.* [18] improved the network structure of R^{2}CNN by adding a fully connected layer branch to achieve prow direction prediction. In their work, the four sides of the rotated bounding box are labeled in a counterclockwise direction, and then the berthing and sailing direction of the ship is predicted in the new branch. The experimental results showed that the sailing predictions always appear at the prow or stern, indicating that the network first learns to find the long side and then judges which end of the long side is the prow.

C. Boundary Discontinuity Problem

In multioriented target detection networks [15]–[18], a five-element vector (x, y, w, h, θ) is usually used to represent a rotated bounding box [25] that encloses objects. To predict the rotated bounding box (predicted box), each box is regressed

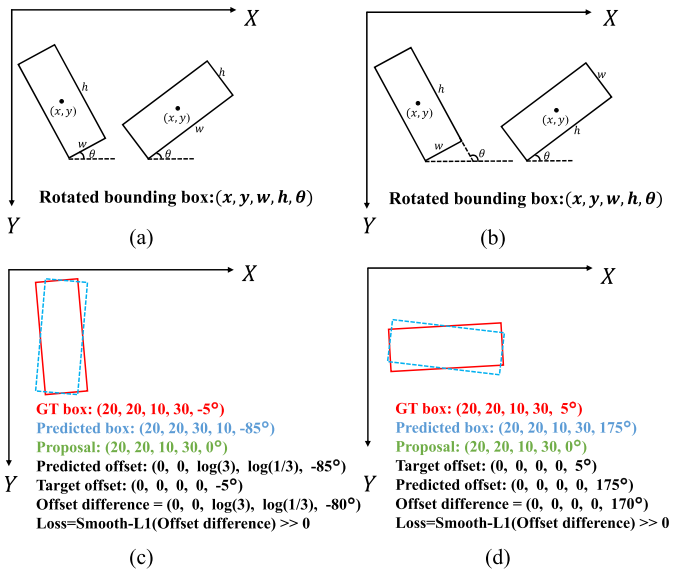


Fig. 1. Representation of rotated bounding boxes in detectors with ARM (a) R-DFPN and (b) R^{2}PN. In addition, the boundary discontinuity problem of these detectors when calculating the regression loss near the angular boundary (No proposals are drawn for easy viewing here) (c) R-DFPN and (d) R^{2}PN.

by the coordinates of the corresponding anchor (anchor box). Therefore, the ARM is defined as follows:

$$\begin{aligned} t_x &= (x_p - x_a)/w_a, \quad t_y = (y_p - y_a)/h_a \\ t_w &= \log(w_p/w_a), \quad t_h = \log(h_p/h_a) \\ t_\theta &= \theta_p - \theta_a \end{aligned} \tag{1}$$

$$\begin{aligned} t_x^* &= (x^* - x_a)/w_a, \quad t_y^* = (y^* - y_a)/h_a \\ t_w^* &= \log(w^*/w_a), \quad t_h^* = \log(h^*/h_a) \\ t_\theta^* &= \theta^* - \theta_a \end{aligned} \tag{2}$$

where (x, y), w, h, θ represents the box's center coordinates, width, height, and incline angle, respectively. The subscripts p, a and the asterisks denote the predicted box, anchor box, and ground-truth box, respectively. The regression loss function is defined as follows:

$$L_{\text{reg}}(t, t^*) = \text{smooth}_{L1}(t - t^*) \tag{3}$$

where t is the predicted offset, t^* is the target offset, and smooth_{L1} is the smooth L1 loss defined in [6]. However, this regression model results in the boundary discontinuity problem due to the periodicity of angles, that is, the regression loss calculation is inaccurate near the angular boundary.

For instance, in R-DFPN, R^{2}CNN++ and [18], θ is the angle at which the x-axis rotates counterclockwise to the first side of an encountered bounding box, and its range is $[-90°, 0°)$, while one side is denoted as w and the other side is h, as shown in Fig. 1(a). When the box direction approaches horizontal or vertical, w and h switch places, and θ changes abruptly at $-90°$ and $0°$. This leads to large differences between the target offset and the predicted offset; thus, the regression loss value becomes erroneously large. In Fig. 1(c), even though the predicted box and the ground-truth box have the same shapes and their directions are close, their w, h, and θ values differ substantially, resulting in a

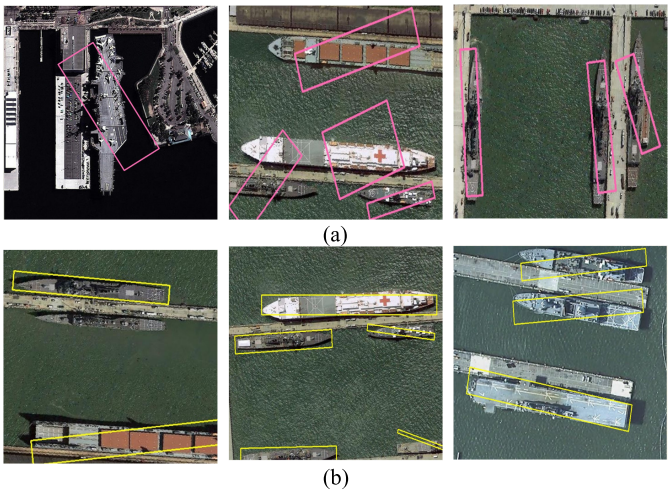


Fig. 2. Inaccurate detection results of detectors with ARM at angular boundary. (a) R-DFPN. (b) R^{2}PN.

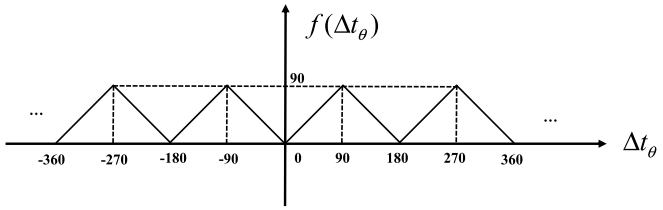


Fig. 3. Definition of an adjustment function $f(\Delta t_\theta)$. The angular offset Δt_θ is constrained by the adjustment function to avoid being too large near the boundary.

huge regression loss value. In R^{2}PN [15] and rotated region-based CNN (RR-CNN) [26], h represents the long side of the rotated bounding box, and θ is defined as the angle between the long side and the x-axis. The angle range is [0°, 180°), as shown in Fig. 1(b). Although this representation of the rotated bounding box avoids the alternation between w and h, the boundary discontinuity problem still exists due to the abrupt changes in θ at 0° and 180° [see Fig. 1(d)]. The boundary discontinuity problem degrades the detection results near the angular boundary in these networks with ARM, as shown in Fig. 2. In this figure, the angles and shapes of the boxes predicted by R-DFPN do not match the ships when they appear either horizontal or vertical in images. Similarly, the angles of the predicted boxes in R^{2}PN deviate when the ship is oriented horizontally.

Based on the definition of rotated bounding boxes shown in Fig. 1(b), we initially attempted to solve the boundary discontinuity problem by modifying the regression loss function. To avoid excessive regression loss near the boundary, the angular offset $\Delta t_\theta (\Delta t_\theta = t_\theta - t_\theta^*)$ is constrained according to the adjustment function $f(\Delta t_\theta)$ in the programming implementation, as shown in Fig. 3. Because the output angular offset Δt_θ of the network may be arbitrary and is not necessarily constrained to within [0°, 180°), the adjustment function is set to be periodic.

In this case, large angular offsets could be avoided, e.g., $f(\Delta t_\theta) = 2°$ when $t_\theta = 179°$ and $t_\theta^* = 1°$. However, the minimum value in this modified loss function is not unique but periodic, so the predicted angle offset t_θ may also be a

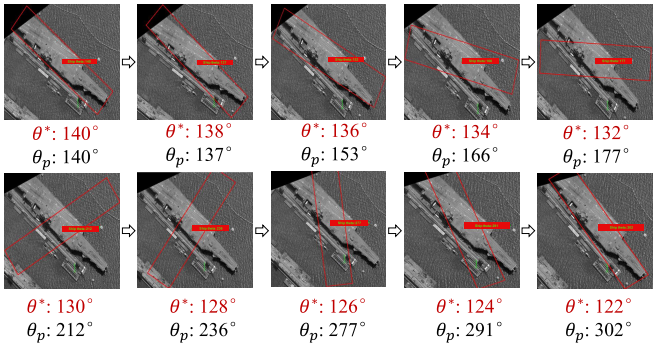


Fig. 4. After simply modifying the regression loss function, when the ground-truth angle θ^* varies from 140° to 122°, the predicted angle θ_p varies from 140° to 302° (i.e., 302° = 122° + 180°).

periodic value after training, e.g., $t_\theta = t_\theta^* \pm k \times 180°$ and $k \in (\ldots, -1, -2, 0, 1, 2 \ldots)$. This approach further induces a nonunique predicted angle $\theta_p (\theta_p = t_\theta + \theta_a)$. A 360° test performed on the same target after modifying the loss function, as shown in Fig. 4, shows that the predicted angle would have a progressive change of approximately 180° within a small angle range. When the ground-truth angle θ^* varies from 140° to 122°, the predicted angle θ_p varies from 140° to 302° (i.e., 302° = 122° + 180°). Under these circumstances, not only does the predicted angle exceeds the defined range [0°, 180°) but the detection results become inaccurate when the predicted angles vary between the periodic values. The reason for this phenomenon is that the predicted angle value is ambiguous and causes the network to fail when determining the correct angle of the target direction; thus, the predicted angles change between the periodic values. Therefore, the boundary discontinuity problem cannot be thoroughly solved by simply modifying the loss function. Hence, in this study, we design a novel regression model based on complex coordinates and discuss it in detail in Section III-C.

III. Methodology

A. Architecture of OPD-Net

In this section, we introduce an OPD-Net based on feature enhancement and an improved regression model. The overall framework of OPD-Net consists of three stages: feature extraction and enhancement, region proposal generation, and prow orientation detection, as shown in Fig. 5.

1) Feature Extraction and Enhancement: In the first stage, to better extract features from remote sensing images containing complicated objects and scenes, we adopt the deep ResNet-101 network [27] as a backbone. Next, to improve the detection performance, the extracted features are enhanced by two modules: the FRN and the PAN. The FRN integrates multiscale features to balance low-level location information and high-level semantic information. In addition, this module further enhances the geometric transformation features of targets. Subsequently, PAN is proposed to enhance the prow feature and weaken the background noise, which guides the network to better detect the prow direction.

2) Region Proposal Network: Based on the enhanced feature map, several proposals are generated by a sliding window

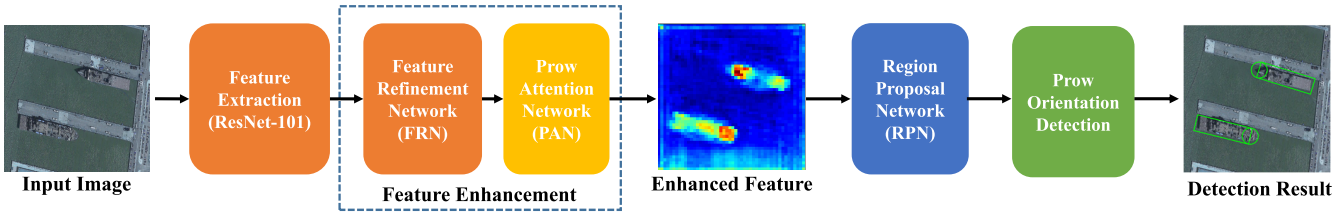


Fig. 5. Overall framework of OPD-Net, primarily including feature extraction and enhancement, region proposal network, and prow orientation detection.

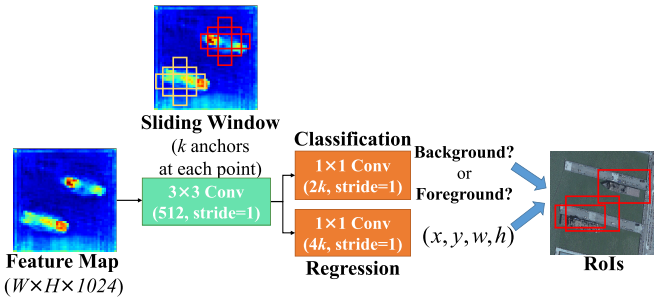


Fig. 6. Network structure of RPN.

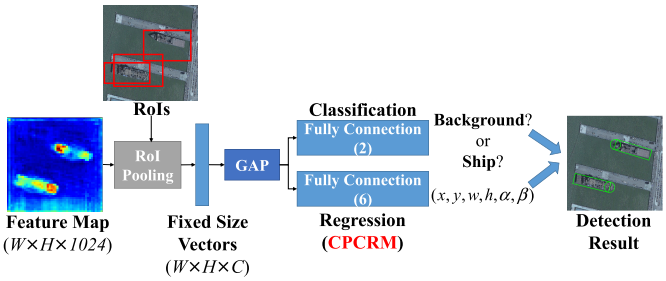


Fig. 7. Network structure in the detection stage.

in the region proposal network (RPN) [8], where the sliding window is implemented by a 3×3 convolution layer, as shown in Fig. 6. To match targets of different sizes and scales, a set of anchors is generated for each window. Compared with horizontal anchors, rotated anchors multiply the number of anchors, resulting in a larger computational load. (For example, for the same feature point, the number of horizontal anchors with three scales and three ratios is 9, while the number of rotated anchors with three scales, three ratios, and three angles is 27). Therefore, we use horizontal anchors in the RPN of OPD-Net. After the 3×3 convolution layer, the feature maps are fed into two 1×1 convolution layers for classification and regression. The regression layer outputs a regression vector corresponding to each anchor to estimate the coordinates of the proposals, and the classification layer determines whether the proposal consists of foreground or background.

3) Prow Orientation Detection: In the detection stage, as shown in Fig. 7, each RoI is extracted from the feature map and standardized into a fixed-size $H \times W \times C$ vector (we set $H = W = 7, C = 1024$, the same setting used in R^2CNN) by RoI pooling. In this stage, a global average pooling (GAP) [28] layer is used to replace the first two fully connected layers in R^2CNN; this reduces the spatial parameters to minimize overfitting. These fixed-size vectors are fed into a GAP layer and then into two fully connected layers to perform classification and regression of the rotated bounding boxes. In particular, a CPCRM is established during regression to solve the boundary discontinuity problem and directly predict the prow direction. The final detection results are obtained after the rotational NMS (R-NMS) operation [18].

B. Feature Refinement Network

1) Deformable Convolution: Different sensor viewpoints, different image resolutions, and different ship courses—even the same type of ship—cause the sizes, shapes, positions, and directions of ships to vary in remote sensing images. Thus, ship targets have numerous geometric transformation features. However, the adaptability of conventional CNNs to geometric transformations is almost entirely dependent on the diversity of input data during training, primarily because the standard convolution is established based on sampling in a regular grid; thus, these models have no ability to model the geometric transformations of targets. Consequently, the extracted features cannot adequately represent the target geometric information. To enhance the feature maps with balanced multidimensional information and the geometric transformation of targets, we design an FRN as one of the elements of OPD-Net.

The deformable convolution [29] greatly improves the geometric transformation modeling ability of the network. As a geometric self-adaptive method, deformable convolution imports 2-D offsets to a regular convolution grid and samples the intermediate feature maps from these flexible locations rather than from a fixed location, allowing free deformations of the sampling grid. Mathematically, a standard convolution can be defined as follows:

$$Y(p_0) = \sum_{p_0 \in Z} W(p_i) \cdot X(p_0 + p_i) \tag{4}$$

where Z is a regular grid over the input feature map X, p_i enumerates the locations in Z, p_0 denotes each location on the output feature map Y, and W denotes the weight coefficient matrix. The deformable convolution augments the regular grid Z with offsets $\{\Delta p_i | i = 1, \ldots, N(N = |Z|)\}$, which can be formulated as follows:

$$Y(p_0) = \sum_{p_i \in Z} W(p_i) \cdot X(p_0 + p_i + \Delta p_i) \tag{5}$$

where the offset Δp_i is the output from another convolution and is usually a noninteger value. Because the sample position $p_0 + p_i + \Delta p_i$ might be fractional and does not correspond to a regular pixel in the feature map, it should be calculated by bilinear interpolation

$$X(p) = \sum_{q} G(q, p) \cdot X(q) \tag{6}$$

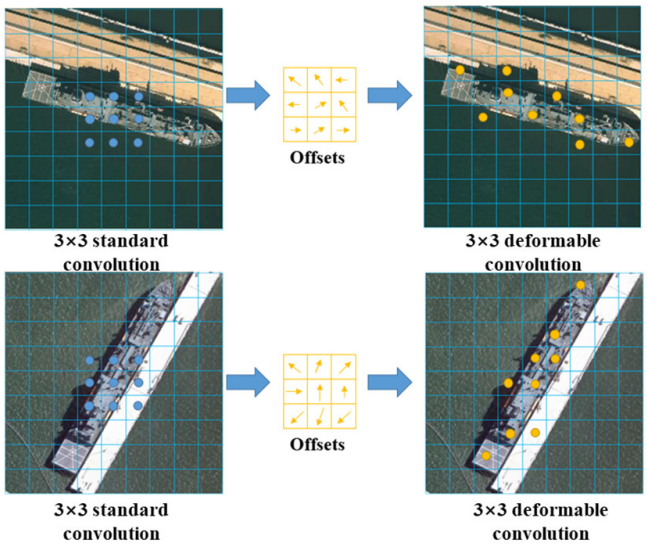


Fig. 8. Illustration of deformable convolution for ship targets.

where p denotes an arbitrary (fractional) location [$p = p_0 + p_i + \Delta p_i$ for (6)], q enumerates all the integral spatial locations around position p, and $G(\cdot)$ represents the bilinear interpolation kernel.

A schematic of deformable convolution is shown in Fig. 8. In this figure, the same type of ship presents different patterns in two images: in contrast, the standard convolution sample only at a fixed position. Deformable convolution enables a concentration of the sampling points on the target by using offsets, which better fit the geometric transformation features of the target.

2) Structure of FRN: The FRN structure is displayed in Fig. 9. In the backbone network, the low-level feature map reflects relatively accurate target locations but contains fewer semantics of the target. In contrast, the high-level feature map contains richer semantics but the locations of targets are more ambiguous. To balance the position information of the low-level features with the semantic information of the high-level features, the multilevel features are simply summed and averaged from the backbone network ResNet-101, where the last layer of each residual block is selected to form the feature maps $\{C_2, C_3, C_4, C_5\}$ in the bottom-up feedforward process. Considering the relative balance between the location and semantic information, C_4 is selected as the basic feature because the resolution of C_4 covers the most small targets. To sum them with C_4, C_3 and C_5 are transformed to C_3' and C_5', respectively, as follows:

$$C_3' = \text{Conv}_{1\times1}(\text{Down}(C_3))$$
$$C_5' = \text{Conv}_{1\times1}(\text{Up}(C_5)) \tag{7}$$

where $\text{Conv}_{1\times1}(\cdot)$ denotes a convolution with a 1×1 kernel and 1024 filters, and Down$(\cdot)$ and Up$(\cdot)$ represent the downsampling and upsampling operations through bilinear interpolation, respectively. After doing that, the number of channels and the scale of C_3' and C_5' are the same as those of C_4. By adding and then averaging the three feature maps, a balanced feature map F_1 is obtained

$$F_1 = \left(C_4 + C_3' + C_5'\right)/3. \tag{8}$$

Next, the balanced feature map F_1 is input into three deformable blocks composed of several 1×1 convolutions and 3×3 deformable convolutions. Each deformable block adopts a residual structure to enhance the features by summing them with the output of the previous structure; this approach also avoids the vanishing gradient problem from which networks with deep layers suffer. Finally, the feature map F_2 is the output of the FRN, and it contains adequate information regarding the geometric transformations of targets, which improves the accuracy of prow detection. That conclusion is well supported by the experiments presented in Section IV.

C. Prow Attention Network

The visualization of the feature map obtained by the FRN, as shown in Fig. 10, demonstrates that the response to ships has a nonuniform distribution. Specifically, the response to the prow is higher than that of the hull, which implies that a network constrained by the destination task will pay more attention to the front parts of ships during training and that the prow can be determined according to the higher response in the FRN feature map. However, the prow and other parts of some ships carry similar semantic information in images and can easily cause the network to predict the wrong heading. Moreover, the background around the targets also has a relatively high response, which can result in false alarms. One feasible approach is to guide the network to focus on the prows of ships and to suppress the background noise via supervised learning.

Inspired by the MDA network (MDA-Net) [16], the PAN is designed as a target feature enhancement element in OPD-Net. Its structure is depicted in Fig. 11. A three-channel saliency map is learned through an Inception [30] structure with multiple layers of different convolution kernels. Then, a three-channel saliency probability map is acquired after a softmax operation. Three channels correspond to the probability of background, hull, and prow, respectively (the probability value is [0, 1]). The two channels representing the hull and the prow are selected and then weighted and summed to estimate the prow attention mask [see Fig. 12(c)]. To adaptively obtain the weights between channels, a convolution combination of [1×1, 3×3, 1×3, 3×1] is implemented on the channel of the prow before the summation. Then, the enhanced feature map is acquired by multiplying the estimated mask with the input feature map, which removes most of the noise in the background and generates a high response to the prow [see Fig. 12(d)]. To train the PAN, first, a three-value map needs to be constructed based on the ground truth, i.e., setting the prow, hull, and background to 2, 1, and 0, respectively, as shown in Fig. 13. Then, the prow-to-ship ratio is set to 0.25 according to the best results in the experiments. Second, the three-value map is one-hot encoded to a three-channel binary map. Finally, the cross-entropy loss between the three-channel saliency probability map and the ground truth is utilized as the prow attention loss during training.

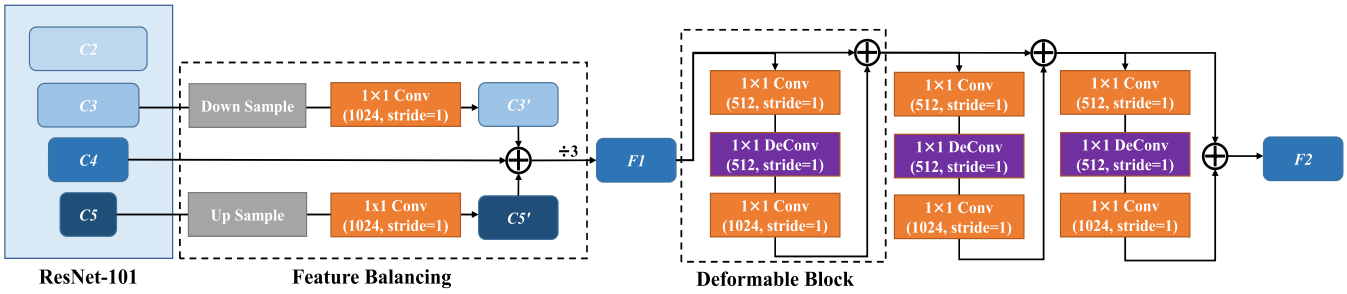


Fig. 9. Structure of FRN, including Feature Balancing and three Deformable Blocks.

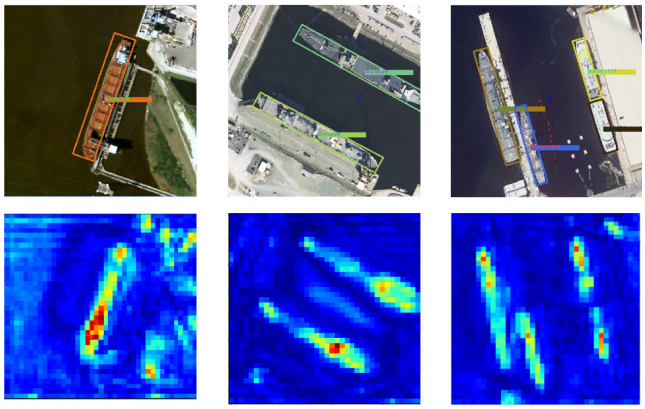


Fig. 10. Visualization of feature map after FRN, where the feature response to the prow is higher than that of the hull.

D. Complex Plane Coordinates Regression Model

According to the analysis in Section II-C, the boundary discontinuity problem is probably caused by the ARM. In this model, the orientation of a rotated bounding box is represented by an angle. Considering that angles are periodic, the angular boundary is discontinuous. Therefore, it is necessary to establish a new regression model in which the direction of the rotated bounding boxes is defined to make it unique and to make the boundary continuous. For the above reasons, an improved regression model, namely, the CPCRM, is proposed. As illustrated in Fig. 14, first, a six-element tuple $(x, y, w, h, \alpha, \beta)$ is designed to represent the arbitrarily oriented rotated bounding box, where (x, y) are the coordinates of the center of the bounding box, w is the length of the long side, and h is the short side. (α, β) are the coordinates of the prow direction on the unit circle on the complex plane, which can be calculated as follows:

$$\alpha = \cos\theta, \quad \beta = \sin\theta \tag{9}$$

where θ is the angle between the prow direction and the x-axis. Starting from the positive x-axis, θ increases clockwise, and its range is $[-180°, 180°)$. Although θ is still discontinuous at the boundary (there is still a 360° abrupt change from $-180°$ to 180°), the coordinates (α, β) are continuous in any ship heading direction. In addition, the coordinates (α, β) are unique and aperiodic; therefore, there is no ambiguity during prow orientation detection. Therefore, the predicted coordinates (α_p, β_p) are first calculated on the complex plane

$$(\alpha_p, \beta_p) = (\alpha_a, \beta_a) + (t_\alpha, t_\beta) \tag{10}$$

where (t_α, t_β) represents the regression vector of the prow direction, and (α_a, β_a) represents the coordinates of the anchor direction on the complex plane. Because we use horizontal anchors (no rotation angle) in our network, the values of (α_a, β_a) are all set to $(-1, 0)$.

Consequently, the CPCRM of the rotated bounding box is defined as follows:

$$\begin{aligned} t_x &= (x_p - x_a)/w_a, \quad t_y = (y_p - y_a)/h_a \\ t_w &= \log(w_p/w_a), \quad t_h = \log(h_p/h_a) \\ t_\alpha &= \alpha_p - \alpha_a, \quad t_\beta = \beta_p - \beta_a \end{aligned} \tag{11}$$

$$\begin{aligned} t_x^* &= (x^* - x_a)/w_a, \quad t_y^* = (y^* - y_a)/h_a \\ t_w^* &= \log(w^*/w_a), \quad t_h^* = \log(h^*/h_a) \\ t_\alpha^* &= \alpha^* - \alpha_a, \quad t_\beta^* = \beta^* - \beta_a \end{aligned} \tag{12}$$

where x, y, w, h represent the center coordinates, width, and height, respectively, and (α, β) are the coordinates of the prow direction on the complex plane. The variables x_p, x_a, x^* represent the predicted box, anchor box, and ground-truth box, respectively (likewise for y, w, h, α, β). Now, the regression loss function (3) is continuous and aperiodic under CPCRM; thus, the boundary discontinuity problem is well solved. Compared with the ARM, CPCRM introduces only a few additional calculations and can be easily embedded in existing DCNN-based detection methods.

Considering that the angle is needed to calculate the skew intersection over union (IoU) between predicted boxes, the predicted coordinates (α_a, β_a) are converted into the predicted angle θ_p by the following piecewise function $F(\alpha, \beta)$:

$$\theta_p = F(\alpha_p, \beta_p) \times 360/2\pi \tag{13}$$

$$F(\alpha, \beta) = \begin{cases} \tan^{-1}(\beta/\alpha) & \alpha > 0, \beta > 0 \\ \tan^{-1}(\beta/\alpha) + \pi & \alpha < 0, \beta > 0 \\ \tan^{-1}(\beta/\alpha) - \pi & \alpha < 0, \beta < 0 \\ \tan^{-1}(\beta/\alpha) & \alpha > 0, \beta < 0 \\ 0 & a \geq 0, \beta = 0 \\ \pi & a < 0, \beta = 0 \\ \pi/2 & a = 0, \beta > 0 \\ -\pi/2 & a = 0, \beta < 0. \end{cases} \tag{14}$$

At this stage, each predicted box is represented as a five-element tuple $(x_p, y_p, w_p, h_p, \theta_p)$, and the range of θ_p is $[-180°, 180°)$. The final prediction results are acquired after R-NMS processing based on the skew IoU.

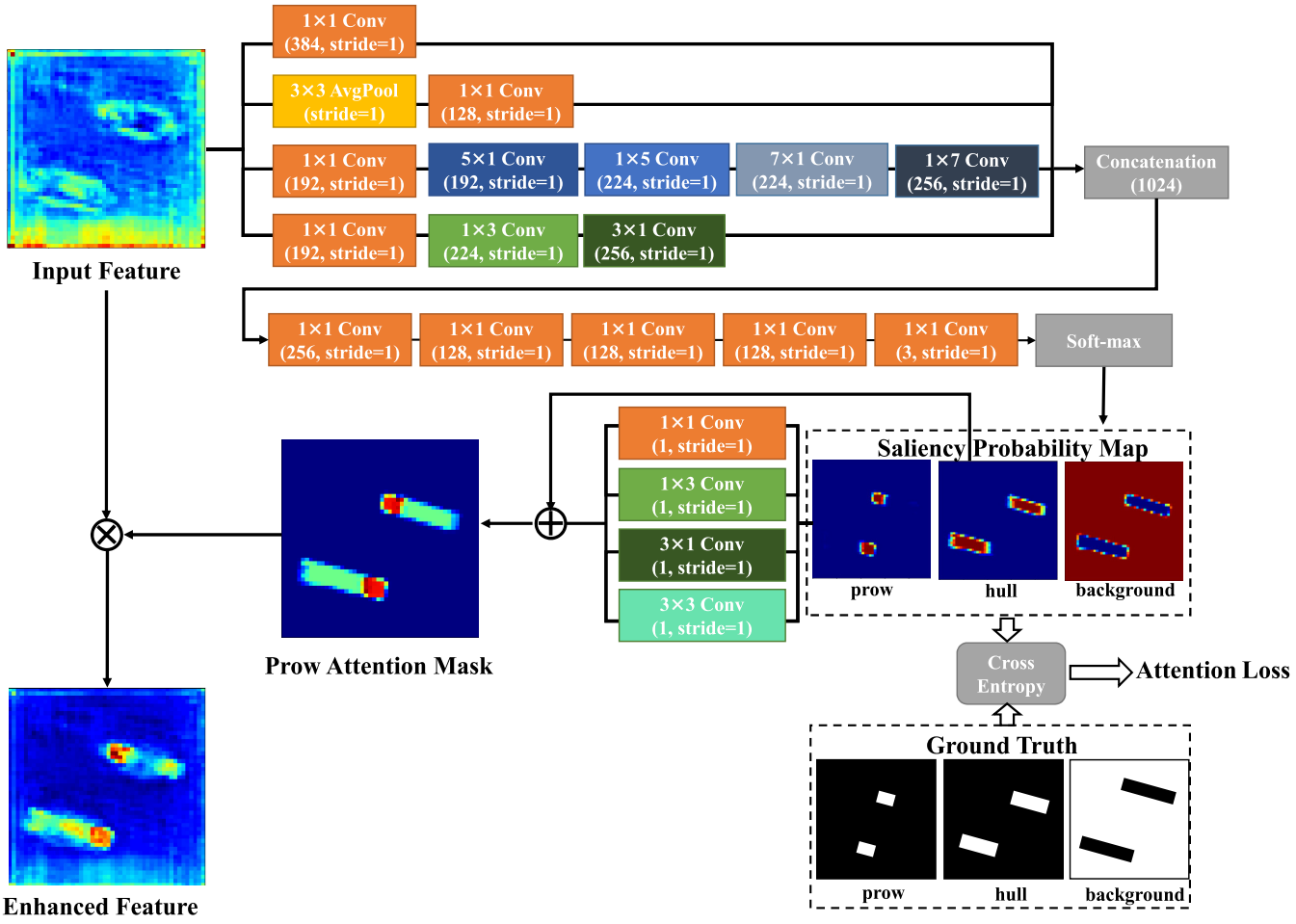


Fig. 11. Structure of the PAN. The PAN generates a prow attention mask that enhances the prow feature and weakens the background noise.

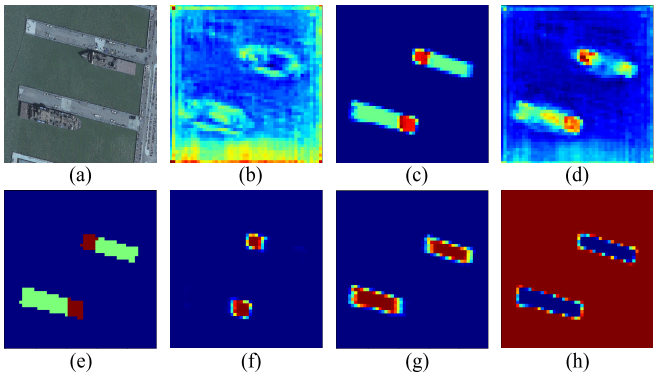


Fig. 12. Visualization of feature maps from PAN. (a) Input image. (b) Input feature map of PAN. (c) Prow attention mask. (d) Output feature map of PAN. (e) Ground truth. (f) Saliency map of the prow. (g) Saliency map the of hull. (h) Saliency map of the background.

For training, the samples are labeled using the following rules: take the top left corner of the prow as a starting point and draw a quadrilateral clockwise, as shown in Fig. 15. Next, the minimum enclosing rectangle of the quadrilateral is considered as the ground-truth box, while the line from the center point of corner points 3 and 4 to the center point of corner points 1 and 2 is considered as the prow direction.

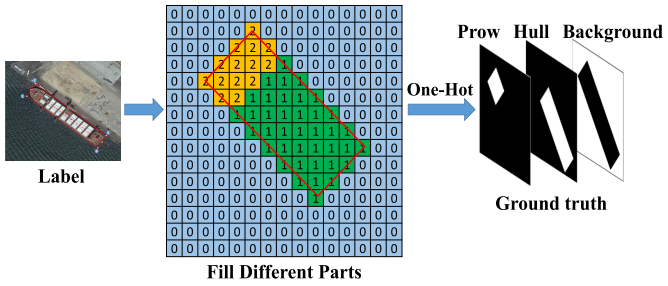


Fig. 13. Illustration of ground truth generation for the PAN.

E. Loss Function

In our method, a multitasking loss is employed to minimize the objective function, defined as follows:

$$L = \lambda_1 \frac{1}{N} \sum_i L_{\text{cls}}(p_i, l_i) + \lambda_2 \frac{1}{N} \sum_j L_{\text{reg}}\left(t_j, t_j^*\right) + \lambda_3 \frac{1}{i \times j \times k} \sum_i \sum_j \sum_k L_{\text{prow}}\left(u_{ijk}, u_{ijk}^*\right) \quad (15)$$

where N represents the mini-batch size of the proposals generated in RPN, and l_i represents object labels (l_i is 1 when

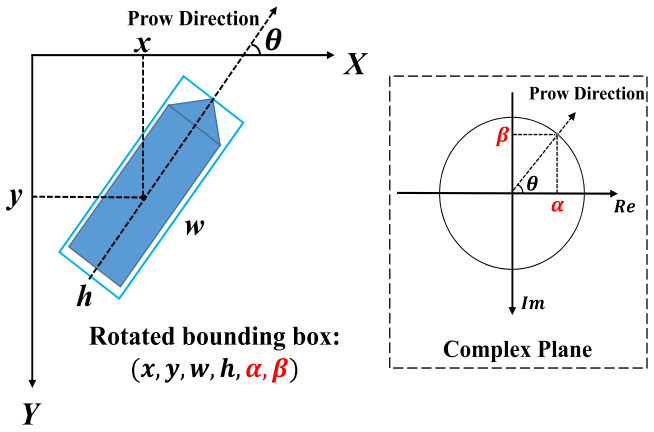


Fig. 14. New representation of rotated bounding boxes in our method, where the prow direction is represented by the coordinates on the unit circle of the complex plane.

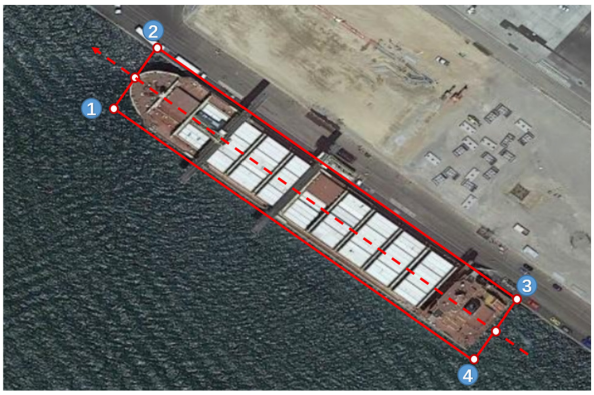


Fig. 15. Rule for labeling samples.

the proposal is positive and 0 when the proposal is negative). p_i represents the probability of various classes calculated by a softmax function, t_j represents the predicted offset vector $(t_x, t_y, t_w, t_h, t_\alpha, t_\beta)$, t_j^* represents the target offset vector of the ground truth $(t_x^*, t_y^*, t_w^*, t_h^*, t_\alpha^*, t_\beta^*)$, and (u_{ijk}, u_{ijk}^*) represents the prediction and ground truth of each pixel in the three-channel saliency probability map. $\lambda_1, \lambda_2, \lambda_3$ are hyperparameters for balancing the subtasks during training. In all the experiments in this article, we set $\lambda_1 = 2, \lambda_2 = 4, \lambda_3 = 1$. The functions L_{cls}, L_{reg}, and L_{prow} are defined as follows:

$$L_{\text{cls}}(p, l) = -l \log(p) \tag{16}$$

$$L_{\text{reg}}(t, t^*) = \text{smooth}_{L1}(t - t^*) \tag{17}$$

$$L_{\text{prow}}(u_{ijk}, u_{ijk}^*) = -u_{ijk}^* \log(u_{ijk}). \tag{18}$$

The smooth L1 loss is defined as follows:

$$\text{smooth}_{L1}(x) = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & \text{otherwise.} \end{cases} \tag{19}$$

F. Training

For training, the backbone ResNet-101 network [27] of OPD-Net is initialized by model pretrained on ImageNet [9]. To avoid exceeding the memory of the GPU, we set the training batch size to 1 in all the experiments. The total number of training iterations is 50k, with a learning rate of 0.0003 for the first 30k iterations, 0.00003 for the next 10k iterations, and 0.000003 for the last 10k iterations. We found that the model works well after approximately 50k iterations. The full set of training iterations (approximately 28 epochs) requires approximately 8 h on our device. In addition, we use Momentum [31] as an optimizer, setting the weight decay and momentum to 0.0001 and 0.9, respectively. To better fit ship targets with different aspect ratios and sizes, the size of the anchors is configured to {32, 64, 128, 256, 512}, the anchor ratios are set to {1:1, 1:3, 3:1, 1:5, 5:1, 1:7, 7:1, 1:9, 9:1}, and the anchor stride is 16. In the RPN stage in OPD-Net, 256 anchors are sampled as a mini-batch for training, and the ratio of positive to negative samples is set to 1:1. In the prow orientation detection stage, 128 proposals are sampled as a mini-batch, and the ratio of positive to negative samples is set to 1:1.

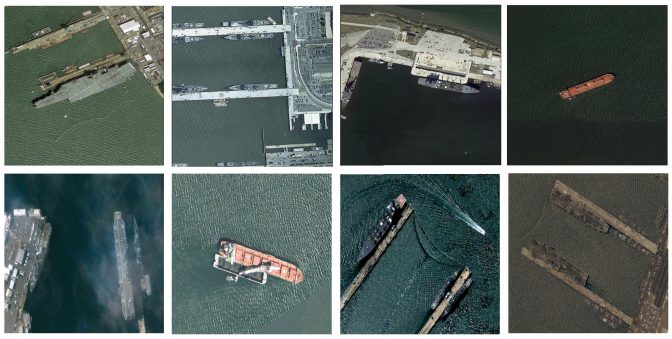


Fig. 16. Some samples from the two data sets. First row: our data set. Second row: HRSC2016 data set.

IV. Experiments

All the experiments were executed on a server equipped with an NVIDIA GeForce GTX 1080Ti GPU with 11-GB memory and executed on the TensorFlow deep learning framework [32].

A. Data Sets

1) Our Data Set: We collected our data set from Google Earth [33]. It consists of 70 optical remote sensing images sized from 4000 × 4000 pixels to 10000 × 10000 pixels. The image resolutions vary from 0.3 to 1.0 m. The scenes included in this data set mainly show civilian ports and naval bases. The targets in the images consist of various ship types with different sizes and headings, from cargo ships and aircraft carriers to small boats, as shown in Fig. 16. The original images were labeled according to the rule shown in Fig. 15. To ensure that each target in the high-resolution images is completely sampled at least once, we cropped the original images into 800 × 800-pixel subimages by taking the center points of targets as the centers of the images. We implemented rotation augmentation at angles of [90°, 180°, 270°] to increase the diversity of heading directions. Finally, we obtained a total of 3480 samples, of which 50% are for training and 50% are for testing.

Table I shows an overview of our data set. Statistically, there are 1740 images containing 3698 ships in the training set and 1740 images containing 3640 ships in the test set.

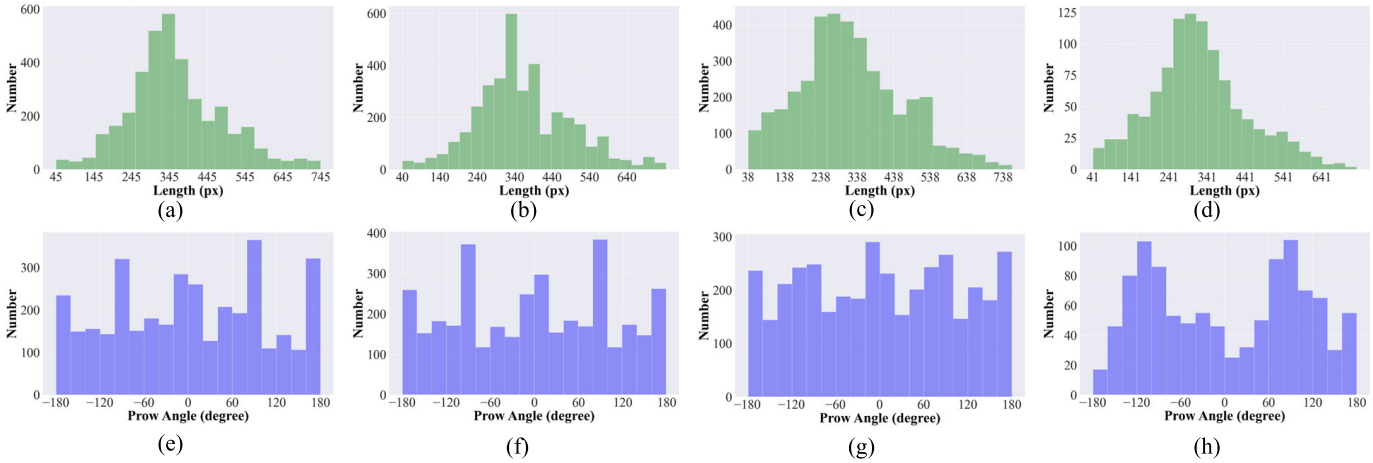


Fig. 17. Ship length distribution maps of the two data sets (a) training set of our data set, (b) test set of our data set, (c) training set of the HRSC2016 data set, and (d) test set of the HRSC2016 data set. In addition, the figure shows the prow angle distribution maps of the two data sets (e) training set of our data set, (f) test set of our data set, (g) training set of the HRSC2016 data set, and (h) test set of the HRSC2016 data set.

TABLE I
OVERVIEW OF TWO DATA SETS

Dataset	Data type	Image	Target
Our	Training	1740	3698
	Test	1740	3640
HRSC2016	Training	436×4	1207×4
	Test	444	1228

The lengths of the ships in the data set vary substantially: from 40 pixels to 745 pixels, as shown in Fig. 17(a) and (b). The direction distribution of prows in the data set is almost balanced, as shown in Fig. 17(e) and (f), and the data set can be used to test the detection performance of different methods in multiple directions.

2) HRSC2016 Data Set: The HRSC2016 data set [19] is an optical remote sensing image data set collected from Google Earth for ship detection purposes. It covers six famous harbors. The data set is divided into a training set, a validation set, and a test set containing 436 images (1207 samples), 181 images (541 samples), and 444 images (1228 samples), respectively. The image resolutions range between 0.4 and 2 m, and the image sizes range from 300 × 300 to 1500 × 900. In the HRSC2016 data set, the targets are annotated by bounding boxes and rotated bounding boxes. In addition, the data set provides ship prow position information for all ships with V-shaped prows. In the experiment, we convert the rotated bounding boxes into the annotated form shown in Fig. 15, depending on the positions of the ship prows. In this way, the HRSC2016 data set can be used for both ship detection and prow detection. To meet network input and memory limitations, all the images in the data set were resized to 800 × 800, and rotation augmentation at angles of [90°, 180°, 270°] was also adopted during training.

Similar to our data set, the lengths of targets in the HRSC2016 data set range from 40 to 700 pixels, although most are between 200 and 400 pixels, as shown in Fig. 17(c) and (d). The prow orientation is almost balanced in the HRSC2016 training set, while in the test set, the proportion of horizontally oriented targets (i.e., 0°, 180°, −180°) is small, as shown in Fig. 17(g) and (h).

B. Evaluation Indicators

We adopted average precision (AP) and angle mean square error (AMSE) metrics to evaluate the performances of different networks on the prow detection task. First, four indexes need to be calculated for AP, namely, true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs). A predicted box must satisfy two conditions to be considered as a TP: 1) the IoU between the predicted box and the ground-truth box must exceed an IoU threshold and 2) the angle difference of the prow direction between the predicted box and the ground-truth box must be smaller than the angle deviation threshold θ_{thresh} (we used the setting $\theta_{\text{thresh}} = 15°$ in our work). Predicted boxes that fail to meet these conditions are considered as FP. The FN represents the sum of actual regions not proposed, and the TN is the sum of the negative samples judged as negative, which is not required in our evaluation. Precision and recall are calculated as follows:

$$\begin{aligned} \text{Precision} &= \text{TP}/(\text{TP}+\text{FP}) \\ \text{Recall} &= \text{TP}/(\text{TP}+\text{FN}). \end{aligned} \tag{20}$$

Here, we set the IoU thresholds to 0.5 and 0.7: the 0.5 value is commonly used in object detection, and the 0.7 value is used to further evaluate the fitting ability of the rotated bounding boxes on ship targets. Because the aspect ratios of ship targets are generally very large, a slight deviation of the angle between the prediction and the ground truth will cause a sharp decline in the IoU. Accordingly, the IoU threshold of 0.7 is stricter for the direction and shape of the rotated bounding box.

The AMSE is designed to evaluate the accuracy of the predicted prow direction, and its calculation is as follows:

$$\frac{1}{N}\sum_{i}(\theta_i - \theta_i^*)^2 \tag{21}$$

TABLE II

QUANTITATIVE COMPARISON OF R^2CNN IN DIFFERENT REGRESSION MODELS ON PROW DETECTION

Model	AP (IoU=0.5)	AP (IoU=0.7)	AMSE (IoU=0.5)
ARM	71.60	42.47	12.486
CPCRM	**84.06**	**61.02**	**6.761**

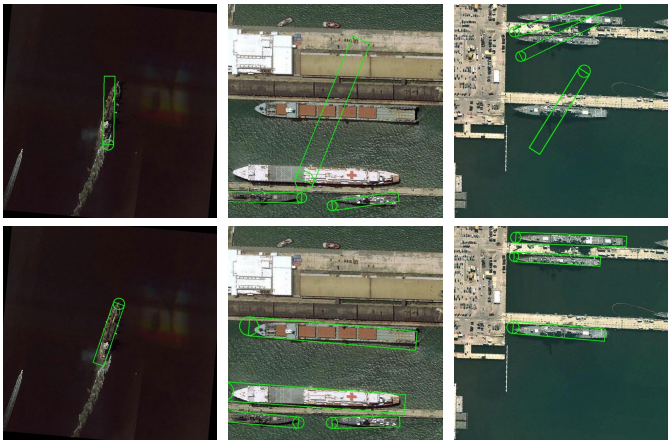


Fig. 18. Comparison results of R^2CNN in two different regression models. First row: ARM; Second row: CPCRM.

where N represents the total number of TPs, θ_i represents the prow direction angle of the ith TP, and θ_i^* represents the prow direction angle of the corresponding ground-truth box.

C. Experimental Analysis

1) Analysis of CPCRM: Due to the large regression loss at the angular boundary, the ARM leads to the boundary discontinuity problem. As described in Section II-C, an adjustment function is attempted to maintain the regression loss correctly, but the periodicity of this function induces ambiguities in the prediction of the detection network, which leads to inaccurate detection results within a small angle range. Hence, we proposed the CPCRM, which has nonperiodic and continuous characteristics, to solve the boundary discontinuity problem. To verify the effectiveness of this novel regression model, we tested the R^2CNN method with both those regression models (i.e., R^2CNN + ARM and R^2CNN + CPCRM). The angle range of the ARM is extended to $(-180°, 180°)$. Some of the detection results are shown in Fig. 18. The predicted box of ARM deviates from the proper direction when the target is close to horizontal. In contrast, the predicted boxes of CPCRM effectively fit the targets in those directions. Table II reports the quantitative comparison results and demonstrates that the CPCRM solves the boundary problems while also achieving a significant performance boost regarding direction detection.

2) Analysis of the FRN: The sizes of the ships in our data set vary widely; it contains not only aircraft carriers with lengths close to the size of the input images but also some small boats that are less than 50 pixels in length. In addition, targets of the same type present various sizes, shapes, and orientations, increasing the difficulty of prow detection. To verify the advantage of FRN for prow detection, we conducted experiments with different numbers of deformable blocks and tested whether to use feature balancing, where "R^2CNN+CPCRM" is set as Baseline_1. The experimental results are listed in Table III. Compared with Baseline_1, the feature balancing approach achieves incremental gains of 0.26% in AP (IoU = 0.5) and 2.11% in AP (IoU = 0.7), which demonstrates that feature balancing is a simple and effective way to alleviate the multiscale problem. The deformable block promotes the detection performance, and the best result is obtained when the block number is 3. The reason for this improvement is that the FRN can fit the geometric transformation of targets by using deformable convolutions. Even if ships of the same type appear new patterns in our test set, they will be detected.

TABLE III

ABLATION EXPERIMENTS ON FRN. KEY: FB = FEATURE BALANCING; DB = DEFORMABLE BLOCK

Method	AP (IoU=0.5)	AP (IoU=0.7)	AMSE (IoU=0.5)
Baseline_1	84.06	61.02	6.761
Baseline_1+FB	84.32	63.13	6.811
Baseline_1+FB+1DB	84.96	62.21	7.115
Baseline_1+FB+2DB	85.13	62.94	7.115
Baseline_1+FB+3DB	**85.22**	**64.72**	**6.703**
Baseline_1+FB+4DB	84.97	63.32	6.792

TABLE IV

ABLATION EXPERIMENTS ON PAN (IOU = 0.5). KEY: R = PROW-TO-SHIP RATIO IN THE PROW ATTENTION MASK

Method	Recall	PRECISION	AP	AMSE
Baseline_2	88.04	90.61	85.22	6.703
Baseline_2+PAN(R=0.10)	88.13	91.39	85.67	6.911
Baseline_2+PAN(R=0.25)	**88.81**	91.37	**86.52**	**6.599**
Baseline_2+PAN(R=0.50)	86.62	90.52	83.81	7.280
Baseline_2+PAN(R=0.75)	88.32	91.41	85.77	7.156
Baseline_2+PAN(R=0.90)	87.91	**91.53**	85.39	7.069

3) Analysis of PAN: As discussed in Section III-C, the PAN enhances the ship prow feature to better regress the heading direction, and it weakens the background noise by using the prow attention mask. This mask divides the ship into two parts: a prow and a hull. To find the best prow-to-ship ratio in the mask, we conducted several experiments with different ratios, and the model "R^2CNN+CPCRM+FRN" is set as Baseline_2. As shown in Table IV, the best prow-to-ship ratio is 0.25, which obtains a recall of 88.81% and a precision of 91.37%. Compared with Baseline_2, this ratio increases the recall and precision by 0.77% and 0.76%, respectively, and the AMSE decreases by 0.104. This result indicates that PAN reduces the angle deviation between the predicted boxes and the targets; thus, more ships can be detected. Meanwhile, false alarms due to background interference are suppressed, which is very advantageous for near-shore ship detection.

4) Analysis of Learning Rate: To choose an appropriate learning rate, we implemented experiments using different learning rates ranging from 0.0001 to 0.001. As Table V

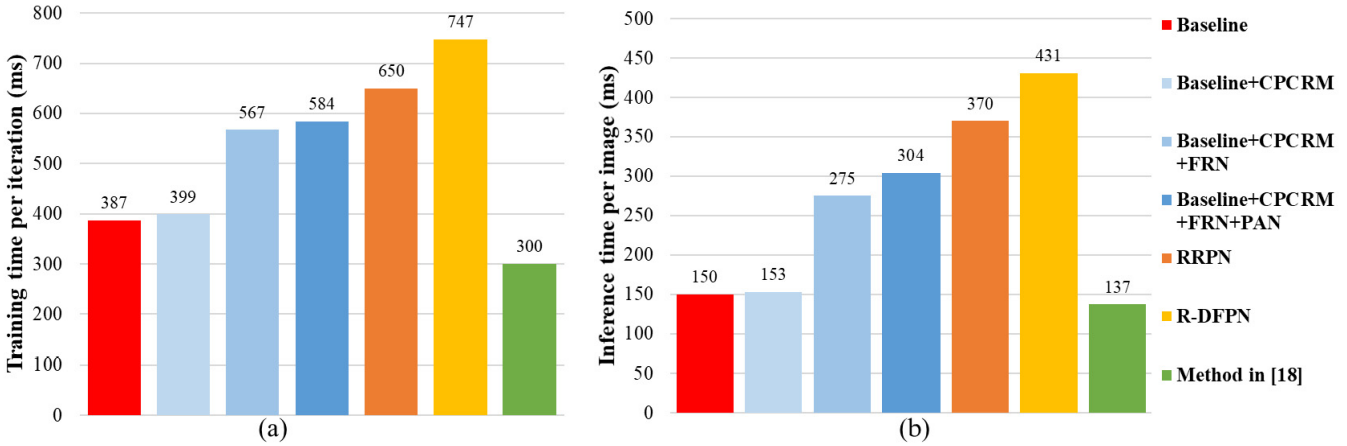


Fig. 19. Time consumption of different methods. (a) Training time per iteration (training batch size is 1). (b) Inference time per image (800 × 800 pixels).

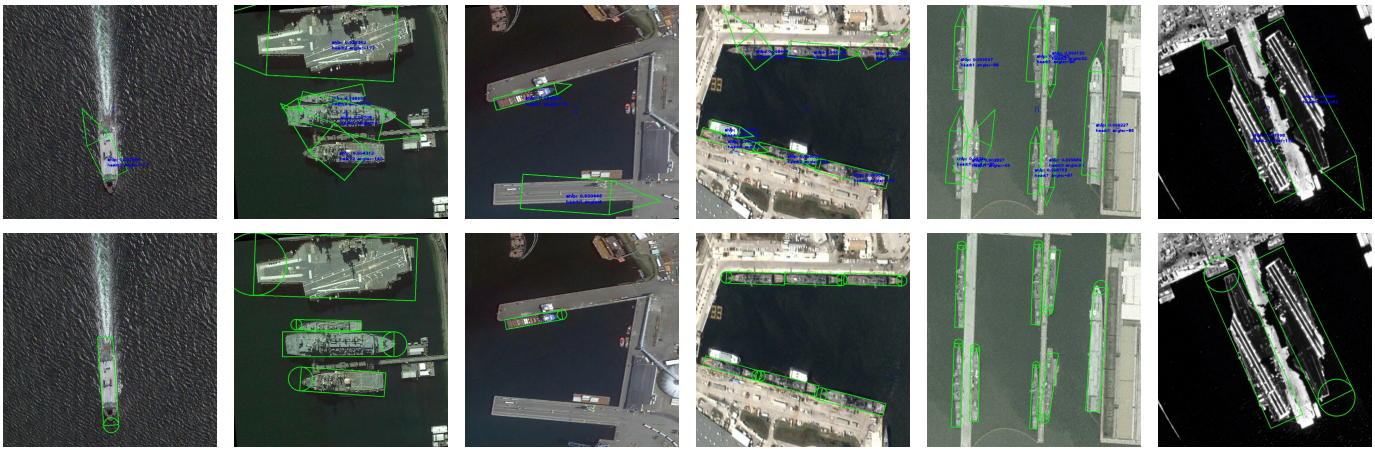


Fig. 20. Some prow detection results on two data sets of two methods. First three columns: images in our data set. Last three columns: images in HRSC2016 data set. First row: [18]. Second row: OPD-Net.

TABLE V
ABLATION EXPERIMENTS ON LEARNING RATE

Learning Rate	AP (IoU=0.5)	AP (IoU=0.7)	AMSE (IoU=0.5)
0.0001	82.59	53.37	8.378
0.0003	**86.52**	**63.69**	**6.599**
0.0005	86.31	62.74	6.710
0.001	80.49	51.13	7.995

shows, a learning rate of 0.0003 is a turning point for AP; the AP decreases when the learning rate is either below or above 0.0003. Therefore, in this article, the learning rate was set to 0.0003 to obtain better results.

5) Analysis of Loss Weights: In (15), the hyperparameters $\lambda_1, \lambda_2, \lambda_3$ are used to balance the classification, regression, and prow attention tasks, respectively. First, we set $\lambda_1 = 2$, $\lambda_2 = 4, \lambda_3 = 1$ as the default values (i.e., the same configuration as in R^{2}CNN++ [16]). Then, to verify the reasonableness of this setting, we conducted several experiments using different configurations of these three parameters, as shown in Table VI. The results show that our method achieves the highest AP when the default parameters are utilized; thus, the default settings are feasible.

TABLE VI
ABLATION EXPERIMENTS ON LOSS WEIGHTS. KEY: λ_1 = CLASSIFICATION LOSS WEIGHT; λ_2 = REGRESSION LOSS WEIGHT; λ_3 = PROW ATTENTION LOSS WEIGHT

$\lambda_1 : \lambda_2 : \lambda_3$	AP (IoU=0.5)	AP (IoU=0.7)	AMSE (IoU=0.5)
1:2:4	82.79	54.47	8.740
1:4:2	85.71	60.43	6.991
2:1:4	81.14	46.09	10.519
2:4:1	**86.52**	**63.69**	**6.599**
4:1:2	81.13	46.41	10.179
4:2:1	85.07	52.38	7.859
2:4:2	84.28	57.43	7.665
2:2:2	83.87	57.44	8.118

6) Analysis of Time Consumption: To evaluate the time consumption of OPD-Net, we recorded the training time and inference time for each iteration of OPD-Net with different modules. We also compared the proposed method with four state-of-the-art detectors: R^{2}CNN (Baseline), R^{2}PN, R-DFPN, and [18], as shown in Fig. 19. All the experiments were conducted on a computer equipped with a single NVIDIA GeForce GTX 1080Ti GPU with 11-GB memory. Due to GPU

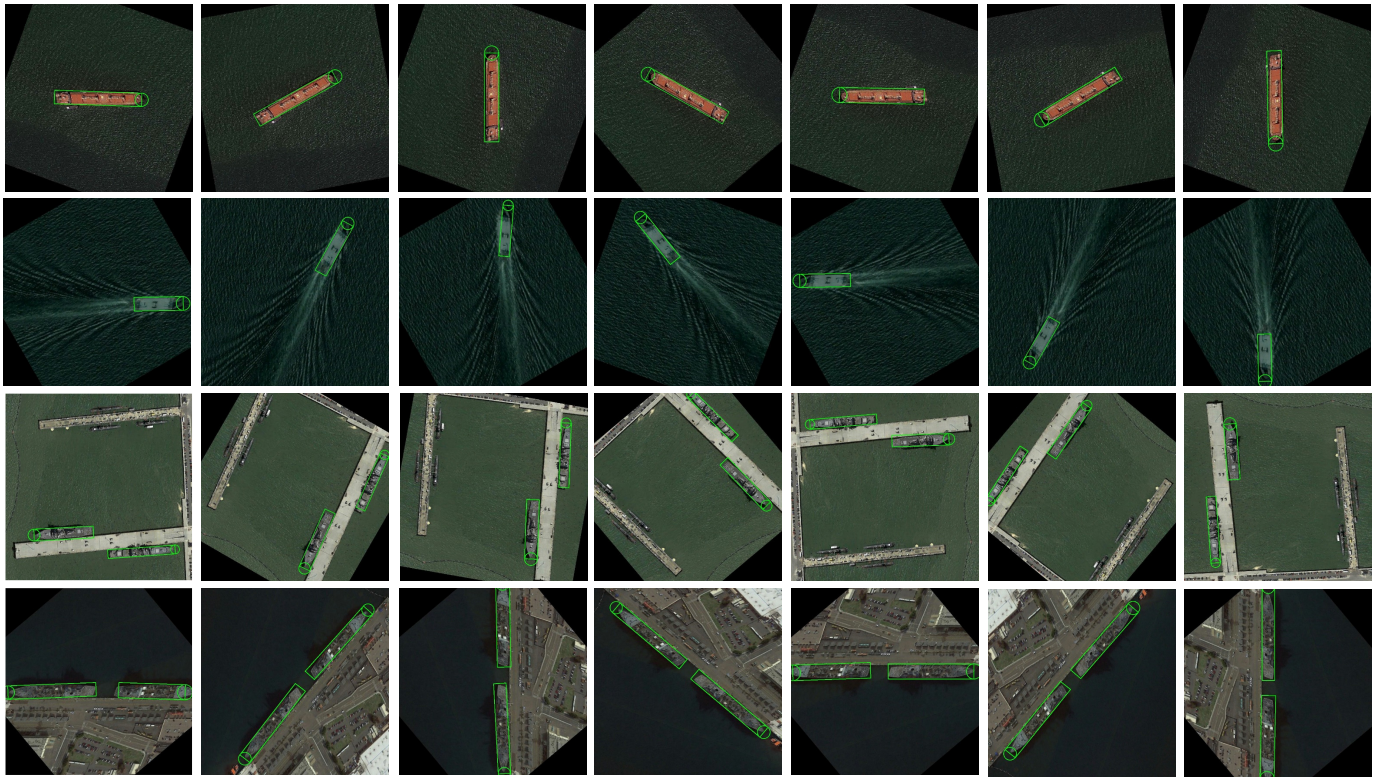


Fig. 21. Some prow detection results of the OPD-Net in 360° test.

memory limitations, the batch size was set to 1 during training. In Fig. 19, the CPCRM costs only approximately 12 and 3 ms longer than the baseline for training time and inference time, respectively. This result shows that the CRCPM improves the detection performance with only a small time consumption cost. It can also be seen that, due to the deformable convolutional operation and multilayer feature fusion, the FRN is the most time-consuming module in OPD-Net. However, the extra time is not enough to constitute an order of magnitude difference between OPD-Net and the Baseline model in terms of time consumption. Furthermore, the time cost of OPD-Net is lower than that of R^2PN and R-DPFN because they introduce higher computational complexity by using rotated anchors. In summary, the above results demonstrate the feasibility of our approach.

D. Comparison With Other Methods in Prow Detection

To quantitatively validate the effectiveness of the proposed method on the prow detection task (i.e., ship heading prediction), we compare the proposed OPD-Net with the DCNN-based prow detection method in [18] on both the HRSC2016 data set and our data set.

1) Parameter Setting: In [18], ResNet-101 is also used as the backbone network, and a dense feature pyramid connection [17] is adopted to fuse multilevel information. In addition, the sizes, strides, and ratios of anchors are configured to {32, 64, 128, 256, 512}, {4, 8, 16, 32, 64}, and {1:1, 1:3, 3:1, 1:5, 5:1, 1: 7, 7:1, 1:9, 9:1}, respectively. For the postprocessing, we set the confidence threshold to 0.5, and the IoU threshold (between two predicted boxes) to 0.1 for R-NMS.

TABLE VII
QUANTITATIVE COMPARISON OF DIFFERENT METHODS ON PROW DETECTION

Dataset	Methods	AP (IoU=0.5)	AP (IoU=0.7)	AMSE (IoU=0.5)
HRSC2016	[18]	76.73	50.93	6.952
	OPD-Net	**77.05**	**56.22**	**6.527**
Our	[18]	82.82	43.46	10.951
	OPD-Net	**86.52**	**63.69**	**6.599**

2) Experimental Results: We first evaluate the prow detection performance of our method and that of [18] on the HRSC2016 data set. The experimental results in Table VII show that although the AP (IoU = 0.5) of our method is only 0.32% higher than [18], our method achieves an increase of approximately 5.29% when the IoU threshold is 0.7. This proves that with a similar detection performance, the rotated bounding boxes in our method fit the pattern of ship targets more accurately.

We also report quantitative comparisons of the two methods on our data set, as shown in Table VII. Compared with the method in [18], OPD-Net achieves a 3.70% improvement in AP (IoU = 0.5) and shows notable improvement for AP (IoU = 0.7), for which our method achieves a remarkable improvement (20.23%), which again demonstrates that the rotated bounding boxes in our method have a higher matching accuracy. In addition, the lower AMSE achieved by OPD-Net shows that the predicted prow direction of our method is closer to the actual direction. Compared with the results on the HRSC2016 data set, there is an even larger gap in detection

TABLE VIII
OMNIDIRECTIONAL TEST OF THE PROPOSED METHOD ON PROW DETECTION

Test Image	AP (IoU=0.5)	AP (IoU=0.7)	AMSE (IoU=0.5)
100	90.36	77.05	6.624
100×360	88.19	72.34	6.434

performance between the method in [18] and OPD-Net on our data set. This is due to the fact that our data set contains more horizontally oriented ships, which are more likely to lead to the boundary discontinuity problem in [18], while OPD-Net solves this problem by using CPCRM. Some of prow detection results on the two data sets are shown in Fig. 20, where our method yields more accurate results when the targets are oriented horizontally or vertically, and it reduces prow misjudgments. This result proves that our approach is not limited by the target orientation and that it can be applied to a wider range of scenarios. In short, the above results reflect the superiority and robustness of our method in prow detection.

E. Omnidirectional Test

Some existing ship detection methods [15]–[17] have difficulty determining the prow orientation, due to the incomplete angle domain and the single-detection task. Moreover, coupled with the limitations of the boundary discontinuity problem, these methods cannot achieve arbitrary orientation detection. Our method uses the CPCRM, which extends the detectable range of the prow direction to 360° in images, giving each angle a unique and continuous representation and thereby achieving omnidirectional detection capability. To verify the stability of our method in any direction, we randomly selected 100 images from the test set of our data set, and rotated them throughout a 360-degree range at a step size of 1°. Table VIII lists the quantitative results, and Fig. 21 displays examples of the detection results. OPD-Net still achieves a good performance on these varied ship directions, which demonstrates its robustness for prow detection.

F. Comparison With Other Ship Detection Methods

In fact, ship detection is a simplified task (no ship prow detection) for our method. Thus, to verify that our method is competitive at the ship detection task alone, we compared it with three state-of-the-art multioriented target detectors: R^2CNN, R^2PN, and R-DFPN.

1) Parameter Setting: All these methods use ResNet-101 as their backbone network. It is worth noting that R^2PN and R-DPFN use rotated anchors, while both R^2CNN and our method use horizontal anchors. Considering the characteristics of ships, the anchor sizes and the anchor ratios of the four methods are configured to {32, 64, 128, 256, 512} and {1:1, 1:3, 3:1, 1:5, 5:1, 1:7, 7:1, 1:9, 9:1}, respectively. The anchor stride for R^2CNN and R^2PN is set to 16, whereas the anchor stride of R-DFPN is set to {4, 8, 16, 32, 64}. The angles of the rotated anchors are set to {−90°, −75°, −60°, −45°, −30°, −15°}. To ensure the fairness of the experiments, the other parameters are held strictly consistent during training and testing. In the postprocessing stage, we also set the confidence threshold to 0.5 and the IoU threshold for R-NMS to 0.1.

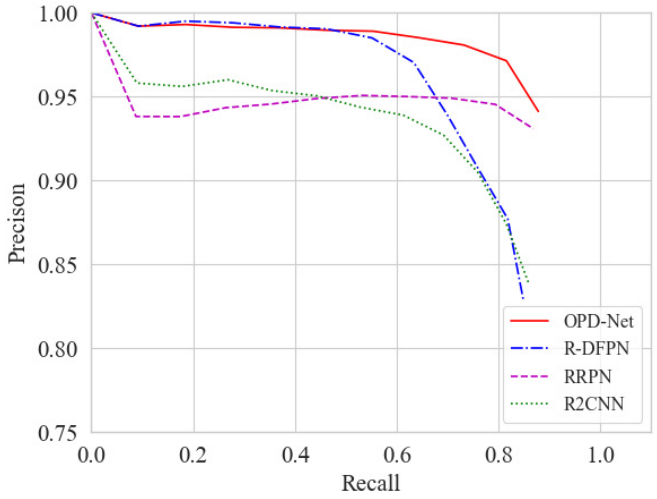


Fig. 22. Precision-recall curves of four methods for ship detection in our data set (IoU threshold = 0.5).

2) Precision-Recall Curve and AP: The recall and precision of detection vary with the confidence score threshold; therefore, we adopted precision-recall curves to evaluate the detection performances of the different methods. Fig. 22 plots the precision-recall curves of R^2PN, R-DFPN, R^2CNN, and OPD-Net on our data set when the IoU threshold is 0.5. In this figure, when the recall is below 0.8, the precision of OPD-Net is steadily maintained above 0.95 and it is consistently higher than the precision of R^2PN and R^2CNN. In contrast, the precision scores of R-DFPN and R^2CNN drop significantly—to below 0.9 when the recall is 0.8. In addition, OPD-Net achieves the best recall when the precision is greater than 0.95. In short, our method is more competitive for ship detection than are the other three methods.

Fig. 23 displays some of the detection results, and Table IX shows a quantitative comparison of the four methods on our data set. At the angular boundary, the other three methods obtain inaccurate bounding boxes and even miss several targets, thereby limiting the detection performance. OPD-Net achieves the highest AP value under both thresholds (88.24% and 70.70%). Compared with R^2PN, it has an increment of 3.39% in AP (IoU = 0.5) and an especially significant improvement of 25.12% in AP (IoU = 0.7). This evaluation shows that our method is not only more effective at ship detection but also better at fitting the directions and shapes of targets.

We also compared the proposed method with the other methods on the HRSC2016 data set. Table IX reports the APs of all the methods under the two IoU thresholds. Our method reaches the highest AP under both thresholds (85.15% and 68.60%). These results again demonstrate that OPD-Net achieves superior performance when used only for the ship detection task.

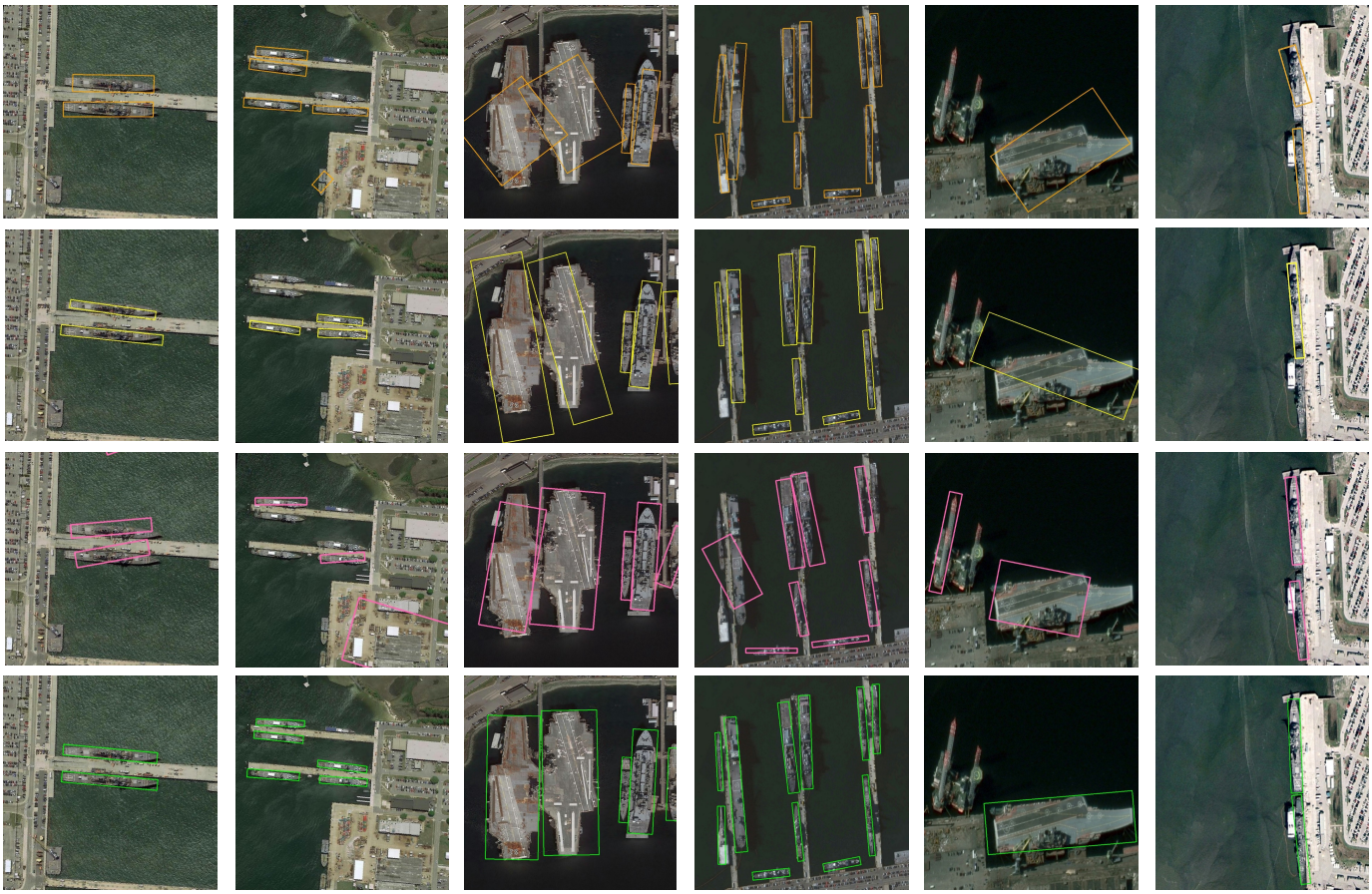


Fig. 23. Ship detection results on two data sets of four methods. First three columns: images in our data set. Last three columns: images in HRSC2016 data set. First row: R^2CNN. Second row: R^2PN. Third row: R-DFPN. Fourth row: OPD-Net.

TABLE IX
QUANTITATIVE COMPARISON OF DIFFERENT METHODS OF SHIP DETECTION

Dataset	Methods	AP (IoU=0.5)	AP (IoU=0.7)
Our	R^2CNN	81.87	43.33
	R^2PN	84.85	45.58
	R-DFPN	83.00	52.89
	OPD-Net	**88.24**	**70.70**
HRSC2016	R^2CNN	80.67	50.67
	R^2PN	82.75	50.64
	R-DFPN	80.93	49.19
	OPD-Net	**85.15**	**68.60**

V. CONCLUSION

In this article, we propose a novel network called OPD-Net that performs accurate prow detection (i.e., ship heading prediction) from optical remote sensing images. First, a FRN is built in OPD-Net to balance multilevel features and obtain information regarding the various geometric transformations of targets. Due to this more informative feature map, our network can adapt to multiple targets scales and direction variations. Second, we design a PAN to guide the network to focus on the ship's prow, which enhances the prow feature while weakening the background noise. Consequently, this module improves the accuracy of prow detection and reduces false alarms. Third, we establish a regression model that uses complex plane coordinates. Benefiting from the continuity and uniqueness of the complex coordinates in any direction, this novel regression model solves the inaccurate detections induced by the angular boundary discontinuity problem. Freed from the limitations of this problem, OPD-Net further achieves omnidirectional ship heading prediction. Finally, to verify the effectiveness of OPD-Net for prow detection, we conducted experiments on both the HRSC2016 data set and on a self-collected data set from Google Earth. The experiments show that the proposed method is more effective and robust than are other DCNN-based algorithms. In addition, on a simpler task (i.e., ship detection), our method is competitive with other state-of-the-art methods. The source code will be released on the Github.[1] Nevertheless, several aspects of our network could still be improved. In future work, we plan to further optimize the structure of the proposed network to improve its performance for ship heading prediction and to reduce its computational complexity.

REFERENCES

[1] G. M. Foody, *Remote Sensing Image Analysis: Including the Spatial Domain*. Norwell, MA, USA: Kluwer, 2004, ch. 3, pp. 37–49.
[2] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[1] https://github.com/Ranbohao/OPD-NET

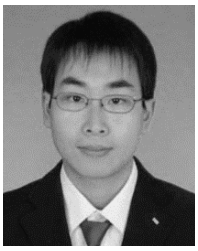
[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[4] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
[5] 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.*, Jun. 2014, pp. 580–587.
[6] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[7] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst*, 2015, pp. 91–99.
[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[10] 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, no. 2, pp. 303–338, Jun. 2010.
[11] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.
[12] 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.
[13] Y. You, Z. Li, B. Ran, J. Cao, S. Lv, and F. Liu, "Broad area target search system for ship detection via deep convolutional neural network," *Remote Sens.*, vol. 11, no. 17, p. 1965, Aug. 2019.
[14] Q. Li, L. Mou, Q. Liu, Y. Wang, and X. X. Zhu, "HSF-Net: Multiscale deep feature embedding for ship detection in optical remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 7147–7161, Dec. 2018.
[15] Z. Zhang, W. Guo, S. Zhu, and W. Yu, "Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 11, pp. 1745–1749, Nov. 2018.
[16] X. Yang *et al.*, "R2CNN++: Multi-dimensional attention based rotation invariant detector with robust anchor strategy," 2018, *arXiv:1811.07126*. [Online]. Available: https://arxiv.org/abs/1811.07126
[17] X. Yang *et al.*, "Automatic ship detection in remote sensing images from Google Earth of complex scenes based on multiscale rotation dense feature pyramid networks," *Remote Sens.*, vol. 10, no. 1, p. 132, Jan. 2018.
[18] X. Yang, H. Sun, X. Sun, M. Yan, Z. Guo, and K. Fu, "Position detection and direction prediction for arbitrary-oriented ships via multitask rotation region convolutional neural network," *IEEE Access*, vol. 6, pp. 50839–50849, 2018.
[19] Y. Jiang *et al.*, "R2CNN: Rotational region CNN for orientation robust scene text detection," 2017, *arXiv:1706.09579*. [Online]. Available: http://arxiv.org/abs/1706.09579
[20] Z. Liu, L. Yuan, L. Weng, and Y. Yang, "A high resolution optical satellite image dataset for ship recognition and some new baselines," in *Proc. 6th Int. Conf. Pattern Recognit. Appl. Methods*, 2017, pp. 324–331.
[21] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Jun. 2017, pp. 2961–2969.
[22] S. Li, Z. Zhou, B. Wang, and F. Wu, "A novel inshore ship detection via ship head classification and body boundary determination," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1920–1924, Dec. 2016.
[23] J. Lin, X. Yang, S. Xiao, Y. Yu, and C. Jia, "A line segment based inshore ship detection method," in *Proc. ICFCA*, 2012, pp. 261–269.
[24] G. Liu, Y. Zhang, X. Zheng, X. Sun, K. Fu, and H. Wang, "A new method on inshore ship detection in high-resolution satellite images using shape and context information," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 3, pp. 617–621, Mar. 2014.
[25] 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.
[26] Z. Liu, J. Hu, L. Weng, and Y. Yang, "Rotated region based CNN for ship detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 900–904.
[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[28] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*. [Online]. Available: https://arxiv.org/abs/1312.4400
[29] J. Dai *et al.*, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.
[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
[31] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. ICML*, 2013, pp. 1139–1147.
[32] (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: http://tensorflow.org/
[33] R. J. Lisle, "Google Earth: A new geological resource," *Geol. Today*, vol. 22, no. 1, pp. 29–32, Jan. 2006.

Yanan You (Member, IEEE) received the Ph.D. degree from the School of Electronic and Information Engineering, Beihang University, Beijing, China, in 2015.

He later worked as a Post-Doctoral Researcher with Beihang University from 2015 to 2017. Since September 2017, he has been with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, as a Lecturer. His research interests include remote sensing image processing, SAR interferometry processing, deep learning, big data technology, and so on.

Bohao Ran received the B.E. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2019, where he is pursuing the M.S. degree with the School of Artificial Intelligence.

His research interests include computer vision and remote sensing image processing, especially on ship detection.



Gang Meng received the Ph.D. degree from Beihang University, Beijing, China, in 2011.

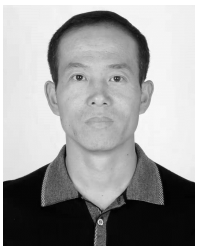
He works as an Engineer with the Beijing Institute of Remote Sensing Information, Beijing. His research interests include object detection and recognition in remote sensing images, pattern recognition, and machine learning.

Zezhong Li is pursuing the master's degree with the Beijing University of Posts and Telecommunications, Beijing, China.

His research interests include deep learning and ship detection on remote sensing optical images.

Fang Liu received the Ph.D. degree from Nankai University, Tianjin, China, in 1997.

She is an Associate Professor with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include broadband IP networks, network traffic monitoring, machine learning, and data mining.



Zhixin Li received the master's degree in signal and information processing from the Dalian University of Technology, Dalian, China, in 2006.

He is an Engineer with the Beijing Institute of Remote Sensing Information, Beijing, China.