AR^{2}Det: An Accurate and Real-Time Rotational One-Stage Ship Detector in Remote Sensing Images

Yuqun Yang, *Student Member, IEEE*, Xu Tang, *Member, IEEE*, Yiu-Ming Cheung, *Fellow, IEEE*, Xiangrong Zhang, *Senior Member, IEEE*, Fang Liu, *Member, IEEE*, Jingjing Ma, *Member, IEEE*, and Licheng Jiao, *Fellow, IEEE*

***Abstract*— Ship detection plays a significant role in the high-resolution remote sensing (HRRS) community, but it is a challenging task due to the complex contents within HRRS images and the diverse orientation of ships. Recently, with the development of deep learning, the performance of the HRRS ship detection model has been improved greatly. Most of them employ deep networks and complicate anchor mechanism to get well ship detection results. Nevertheless, this kind of combination limits the detection efficiency. To address this problem, a new approach named accurate and real-time rotational ship detector (AR^{2}Det) is proposed in this article to detect ships without the anchor mechanism. Based on the extracted features by the feature extraction module (FEM) and the central information of ships, AR^{2}Det adopts two simple modules, ship detector (SDet) and center detector (CDet), to generate and improve the detection results, respectively. AR^{2}Det is efficient due to the simple postprocessing and the lightweight network. Also, AR^{2}Det performs satisfactorily due to the effective generation and enhancement strategy of bounding boxes. The extensive experiments are conducted on a public HRRS image ship detection dataset HRSC2016. The promising results show that our method outperforms the state-of-the-art approaches in terms of both accuracy and speed.**

***Index Terms*— Deep learning, high-resolution remote sensing (HRRS) image, ship detection.**

Manuscript received December 9, 2020; revised April 18, 2021; accepted June 21, 2021. Date of publication July 8, 2021; date of current version January 14, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61801351, Grant 61802190, Grant 61772400, and Grant 61672444; in part by the Key Research and Development Program of Shaanxi under Grant 2021GY-035; in part by the Natural Science Basic Research Program of Shaanxi under Grant 2021JM-139; in part by the Key Laboratory of National Defense Science and Technology Foundation Project under Grant 6142A010301; in part by the China Postdoctoral Science Foundation Funded Project under Grant 2017M620441; in part by the Hong Kong Scholars Program under Grant XJ2019037; in part by the Fundamental Research Funds for the Central Universities under Grant 30919011281 and Grant JSGP202101; in part by Hong Kong Baptist University under Grant RC-FNRA-IG/18-19/SCI/03 and Grant RC-IRCMs/18-19/SCI/01; in part by the ITF of Innovation and Technology Commission of the Government of Hong Kong under Project ITS/339/18; and in part by the Xidian University Artificial Intelligence School Innovation Fund Project under Grant YJS2115. *(Corresponding authors: Xu Tang; Yiu-Ming Cheung.)*

Yuqun Yang, Xiangrong Zhang, Jingjing Ma, and Licheng Jiao are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an 710071, China.

Xu Tang is with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an 710071, China, and also with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, SAR, China (e-mail: tangxu128@gmail.com).

Yiu-Ming Cheung is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, SAR, China (e-mail: ymc@comp.hkbu.edu.hk).

Fang Liu is with the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210023, China.

This article has supplementary downloadable material available at https://doi.org/10.1109/TGRS.2021.3092433, provided by the authors.

Digital Object Identifier 10.1109/TGRS.2021.3092433

I. Introduction

With the development of aerospace and sensor technologies, the resolution of remote sensing (RS) images obtained by diverse earth observation (EO) satellites has been improved dramatically. These high-resolution RS (HRRS) images can provide rich land-cover/land-use information for studying our planet. As a fundamental HRRS images' content understanding task, object detection always draws researchers' attention since it can display a lot of refined information within HRRS images and can be widely used in many realistic applications, such as geospatial object detection [1], vehicle detection [2], [3], and target recognition [4]. Nevertheless, the contents of HRRS are complex, and the objects within the HRRS images are diverse in type, huge in volume, and multiscale in size. It turns out that object detection is a nontrivial task in the RS community. Among various objects within the HRRS images, ships are a specific class, which are important in many fields, including marine traffic monitoring, ship rescue, territorial defense, fisheries management, and marine situational awareness [5]. Therefore, it is still desirable to develop an effective and efficient HRRS ship detection method. This article will concentrate on such a method.

Generally, a basic and important step in the HRRS ship detection is the feature learning/extraction of the images. At the very beginning, handcrafted features are popular. The usual low-/middle-level visual features include scale-invariant feature transform (SIFT) [6], bag of words (BOW) [7], and histogram of oriented gradients (HOG) [8]. Based on these handcrafted features, many ship detection approaches were proposed [9]–[11]. However, their performance cannot meet what we expect because the extracted features are not able to fully describe the complex contents within the HRRS images. Recently, with the development of deep learning [12], [13], especially the deep convolutional neural network (DCNN)

[14]–[16], the representation capacity of the extracted features has been enhanced due to the hierarchical structure and learning manner of DCNN. With the help of these deep features, numerous HRRS ship detection methods were proposed and achieved the cracking performance [10], [11], [17]. Due to the excellent results, DCNN-based methods have dominated the HRRS ship detection community.

The existing DCNN-based ship detection methods can be divided into two categories [18], i.e., the two-stage methods and one-stage methods. For the two-stage methods, the first stage is to generate the region proposals, which aims to find some regions that may contain ships. To achieve this goal, many algorithms were proposed. For example, Zhang *et al.* [19] proposed the rotated region proposal network (R2PN) for the HRRS ship detection, in which the multiorientated proposal generation algorithm is developed based on the common region proposal network. Thus, the ships with the orientation angle information can be detected accurately. The second stage consists of the classification and regression, which focuses on further distinguishing ships and refining their location. For instance, in the paper [20], the fully connected layers are added on the top of the network to accomplish the ship classification and bounding box regression by the specific loss functions. The two-stage methods usually achieve high accuracy in ship detection. Nevertheless, they sacrifice the computation resources and increase the time costs. Thus, it is difficult for them to perform real-time ship detection [21].

To overcome the disadvantages mentioned above, one-stage ship detection methods are proposed, which generates the detection results without the refinement of the region proposal. For example, based on the successful you only look once (YOLO) algorithm [22], a one-stage HSSR ship detection method was introduced [23]. By adding the angle information, the method can detect multiorientated ships rapidly. Although the one-stage detection methods are efficient, their performance is not as good as the two-stage methods. The reasons behind this can be summarized as follows. First, since there is no refinement scheme such as the region proposal, the final detection performance is limited. Second, in the HRRS images, the numbers of the ships and the backgrounds are imbalanced. This would harm the performance of one-stage methods as they do not have the twice classification strategy for ships (which is a common operation in the two-stage methods). Third, it is difficult to select accurate bounding boxes from abundant detection results by the predicted scores since there is always a lot of noise in the predicted score.

To address these problems in one-stage methods, we will therefore develop a new ship detection method, which can obtain accurate results efficiently. We name it accurate and real-time rotational ship detector (AR^2Det). AR^2Det is a simple network with three submodules, including a feature extraction module (FEM), a ship detector (SDet), and a center detector (CDet). FEM is used to learn the basic features from HRRS images and enhance the discrimination of the features through fusing the multiscale information. SDet is developed to decide the positions and geometric attributes of the ships. CDet aims to adjust the predicted scores of SDet so that the final detection results can be obtained more accurately.

The main contributions of this article are summarized as follows.

1) A one-stage ship detection model AR^2Det is proposed, which can accurately detect ships from HRRS images with high efficiency. In the inference stage, the detection speed of AR^2Det (based on ResNet34 [16]) can reach up to 112 frames per second (FPS).
2) In SDet, we propose a relative coordinate scheme to describe the ships' locations by considering position information of feature pixels within the feature maps. Also, instead of using the intersection over union (IoU) as scores, we develop score labels to assess the quality of bounding boxes. The two proposed strategies can ensure detection accuracy and accelerate the training process (about eight times).
3) In CDet, to further reduce the computational costs, the scores of bounding boxes would be adjusted. Thus, many redundant bounding boxes can be ignored. Through this step, mean average precision (MAP) values of AR^2Det are increased from 81.79% to 89.57%. Meanwhile, the inference process can be accelerated by four times.

The rest of this article is organized as follows. The literature related to rotated object detection and ship detection is reviewed in Section II. In Section III, our AR^2Det with four submodels is introduced in detail. The experiments and their discussion are shown in Section IV. Finally, Section V draws a brief conclusion.

II. Related Work

Many successful object and ship detection methods have been proposed in recent years, which can be grouped into: 1) two-stage RS image object detection methods and 2) one-stage RS image object detection methods.

A. Two-Stage HRRS Image Object Detection Methods

Due to the high accuracy, two-stage object detection methods are popular in the RS community. As a classical model of two-stage methods, region convolutional neural network (R-CNN) [24] achieves cracking performance in many applications. Considering the specific characteristics of HRRS images, many variants of R-CNN have been proposed to deal with the HRRS image object detection. For example, Cao *et al.* [25] applied R-CNN to HRRS images directly. The positive results showed that R-CNN can address the HRRS image object detection task. Nevertheless, the reported results cannot achieve what we expect due to the complex contents within RS images. Also, due to the high time complexity of R-CNN, the detection process of HRRS images is time-consuming. To overcome this issue and take more properties of RS images during the detection, an HRRS object detection method was proposed in [26] based on the faster R-CNN [27]. This model combines the pretrained region proposed network (RPN) [27] and the sharing computation algorithm to find the diverse objects within the HRRS images robustly and rapidly. Considering the multiscale information of the objects, Wang *et al.* [28] introduced a multiscale block

fusion object detection for HRRS images. It first divides the large-scale HRRS images into blocks with different scales. Then, the detection results corresponding to multiscale blocks (obtained by faster R-CNN) are fused for the final results. The other variant of faster R-CNN [29] was designed using a suitable region of interest (ROI) scale of object detection to generate accurate results for multiple-scale objects. Apart from the complex contents and the multiscale information, the issue of small objects is another tough point in HRRS object detection tasks. To find them accurately, a network was presented in [30], which is named RS region-based convolutional neural network (R^2-CNN). Through integrating the global attention block and Tiny-Net [30], the small objects within HRRS images can be detected.

As a specific object in HRRS images, except for the general characteristics of HRRS image objects, ships have many other properties [31], such as arbitrary orientation and narrow shape. Therefore, rotational ship detection becomes an important and challenging task. To predict ships in HRRS images precisely, a number of two-stage methods have been proposed. For instance, rotational region CNN (R2CNN) was developed in [32], where the rotated bounding boxes can be obtained by adding the angle into the R-CNN stage. Liu *et al.* [20] proposed the rotated region-based CNN (RR-CNN) to explore the rotational ships. Through learning the features of rotated regions and adopting the rotated region of interest (RRoI) pooling layer [20], the ships with arbitrary orientations can be detected accurately. To further improve the ship detection performance, Zhang *et al.* [19] introduced the R2PN, in which not only the rotated RoI but also the rotated anchor boxes are developed to find and locate different ships. To reduce the computational complexity of R2PN, the region of interest transformer (RoI Transformer) was developed in [33]. Instead of using rotated anchor boxes, RoI Transformer adds the fully connected layer after the region proposal block to explore the rotated information using the horizontal anchor boxes. A similar work was published in [34], where four length ratios of the relative gliding offsets are regressed to assign the angle information to the horizontal anchor boxes.

B. One-Stage HRRS Image Object Detection Methods

Although two-stage methods can obtain accurate detection results for HRRS images, their time complexity is too high for many realistic applications. To reduce time costs, one-stage methods attract scholars' attention.

One-stage methods can locate and classify the object directly without region proposals, which greatly improves the efficiency of detection. The most popular one-stage method would be YOLO [22]. Based on YOLO, many successful methods have been developed for HRRS images recently. For example, a network named you only look twice (YOLT) was proposed in [35] for HRRS images. Through the data augmentation and the output size modification, the performance of YOLT for HRRS images is pleasurable. To eliminate the inference of the complex background within HRRS images, Hou *et al.* [36] developed the refined single-shot multibox detector (RSSD). RSSD consists of three blocks, including a single-shot multibox detector (SSD) [37], a refined network (RefinedNet), and a class-specific spatial template matching (STM) module. Combining them together, the object detection results are enhanced to a big degree. Wang *et al.* [38] introduced another model to reduce the influence of the complex background, named feature-merged single-shot detection (FMSSD). By aggregating the context information in the multiscale and single-scale feature learning, the objects can be detected accurately and rapidly.

For ships, many one-stage detection methods have been proposed in recent years, which push the ship detection toward the real-time stage [18]. For example, inspired by SSD, detector rotatable bounding box (DRBox) [39] was proposed. DRBox defines the rotatable bounding box to predict the poly directional ships, which improves the degree of overlapping between the bounding boxes and ships. To further improve the ship detection accuracy, a single-shot anchor refinement network (SSARN) was developed in [40]. It imitates the two-stage method, which regresses the anchor boxes twice and obtains superior results. Yang *et al.* [17] developed a refined rotation retinanet (R3Det) to address the HRRS object detection, in which a feature refinement module is introduced to improve the discrimination of the ship features.

III. Proposed Approach

The architecture of AR^{2}Det is shown in Fig. 1, which contains an FEM, an SDet, and a CDet. FEM aims to extract the discriminative features from HRRS images, SDet aims at generating the ship bounding boxes and their scores, and CDet focuses on obtaining the ship centers and their confidence. To train AR^{2}Det, four specific loss functions are formulated with the consideration of characteristics of HRRS images. In the inference phase, when the user inputs an HRRS, the trained AR^{2}Det is used to generate the ships' centers, centers' confidence, bounding boxes, and boxes' scores. Then, a postprocessing method, named score ad strategy, is developed to eliminate the redundant and inaccurate bounding boxes according to the centers and their confidence for the final detection results.

A. Feature Extraction Module

The visual features play a vital role in HRRS image object detection. How to obtain effective and discriminative features from HRRS images is always a challenging task. Due to the complex contents within HRRS images, both the low-level features (e.g., color, texture, and shape) and the deep-level features (e.g., context and semantics) should be considered during the representation exploration. Therefore, we adopt ResNet [16] and the feature pyramid network [41], [42] to complete the feature extraction in this article. In addition, to further enhance the discrimination of features, a simple yet useful fusion scheme is developed. The flowchart of FEM is shown in Fig. 2.

As discussed in [42], ResNet can be divided into four residual blocks according to its structure. Thus, when the HRRS image I is inputted into FEM, the features with different scales and various semantics can be learned by different residual

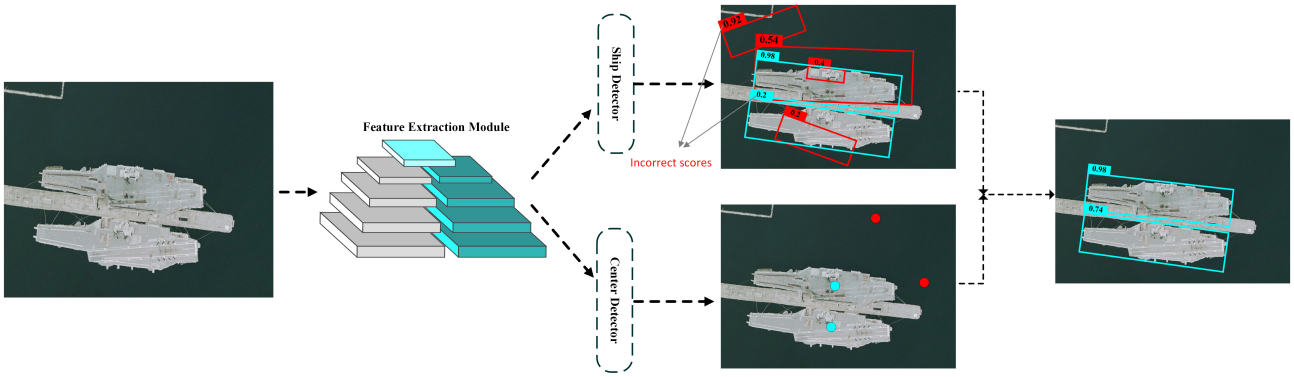

Fig. 1. Architecture of AR^{2}Det, which consists of an FEM, a CDet, and an SDet. The SDet is used to predict the bounding boxes and their scores. The CDet is used to predict the center of ships. The predicted centers are used for adjusting scores of bounding boxes to select the final bounding boxes.

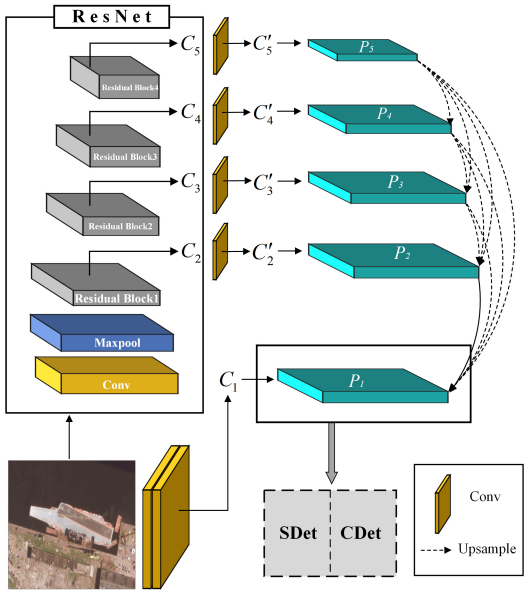

Fig. 2. Framework of features extraction module. It adopts the top-down dense connection to fusing the semantics with different scales and introduces the lateral connection manner to highlight the precise location of various semantic information.

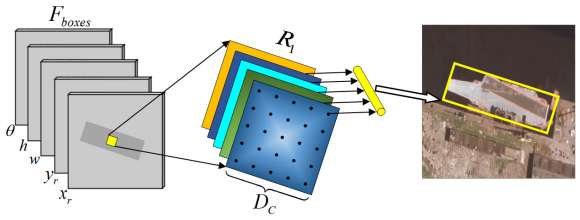

Fig. 3. Features F_{boxes} with five channels. For each ship, the feature pixels within the region R_1 are used to be the training data for predicting ships.

blocks. Here, we denote them C_2, C_3, C_4, and C_5. Then, the feature pyramid network is employed to fuse them for integrating the useful information from different aspects. The feature pyramid network consists of the top-down dense and lateral connections. The top-down dense connection focuses on fusing the semantics with different scales, and the lateral connection aims at highlighting the precise locations of the various semantic information.

Through the feature pyramid network, we can get the fused feature P_2. Although P_2 contains rich information of HRRS images and can be fed into the following modules for generating the detection results, its discrimination and effectiveness could be enhanced further. Taking the characteristics of HRRS images and the ship detection task into account, apart from the high-level semantics and the multiscale information, the low-level features are also important to predict the ships. For example, color and texture features can be used to distinguish the ships from the floating woods and trashes. Also, the shape features are good at finding ships from backgrounds. Therefore, it is necessary to add the low-level features to P_2. To this end, we first convolute the image I with two convolutional layers (the kernel sizes are 3×3 and the stride values are 2×2) to get C_1. Then, P_2 and C_1 are fused in accordance with the rules of the feature pyramid network to get the final visual features P_1 with the size of $W_p \times H_p$.

B. Ship Detector

As a specific location of ship geometry, the ship centers can provide unique information for ship detection. For example, a predicted bounding box has a high overlap with a ship, if its center is closer to the ship center [43]. Also, the complex backgrounds of HRRS images will influence the performance of ship detection negatively. To utilize the center information and reduce the influence of backgrounds, we design an SDet to detect ships within the HRRS image. After inputting the HRRS image into the AR^{2}Det, SDet only occupies the center region of each ship to eliminate the influence of backgrounds. Then, the information from the local (center regions) to the global (whole ships) is fused by continuously convoluting and pooling the HRRS image. Finally, the fused information will be transformed into bounding boxes and their scores by outputting layers.

Specifically, we can first obtain the features F_{boxes} with five channels by convoluting the features P_1 with the kernel size of 3×3 and the stride of 1×1. Then, for each ship, its center region R_1 (a square with the side length of d_c) is selected from F_{boxes}, and all the feature pixels with five channels that

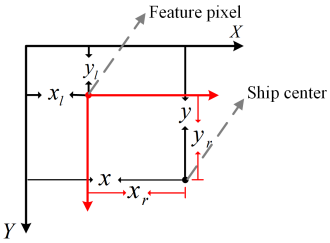

Fig. 4. Transformation of coordinates. The coordinates of ship are transformed from absolute coordinates (based on the top-left corner of the image) to the relative coordinates (based on the position of each feature pixel).

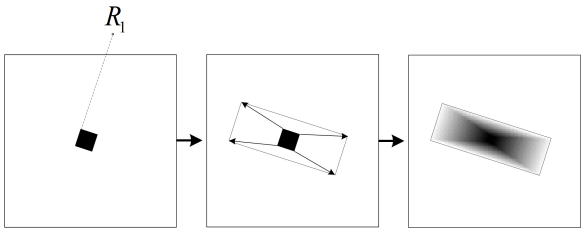

Fig. 5. Process of generating score label.

are located in the region R_1 are trained for predicting the bounding box of the ship. The example is shown in Fig. 3. The ground truth of bounding box (the center coordinates x and y, the long and short side w and h, and the rotation angle θ with the horizontal axis) can be obtained by the annotated HRRS image.

However, for the different feature pixels, the distances between them and the ship center are different. To utilize the position information of feature pixel to optimize relevant parameters adaptively, we predict the relative center coordinates of ship $[x_r, y_r]$ rather than the absolute center coordinates $[x, y]$ of ship, where $[x_r, y_r]$ is the coordinates of the ship center relative to the feature pixel location. We show the transformation of coordinate in Fig. 4. Therefore, for each of features pixel in region R_1, the ground truth of x_r and y_r can be computed by subtracting the feature pixel coordinates $[x_l, y_l]$ from the center coordinates $[x, y]$.

With the definitions mentioned above, the loss function of bounding boxes for an HRRS image can be formulated as follows:

$$\begin{aligned}
\mathcal{L}_{\text{boxes}} &= \frac{1}{N_{R_1}} \sum_i \mathbb{1}^{R_1} \left(\sum_j \text{smooth}_{L_1}(t_j) \right) \\
t_{x_s} &= \frac{\hat{x_{r_s}} - (x - x_l)}{d_c} \times (1 + |\sin\theta|) \\
t_{y_s} &= \frac{\hat{y_{r_s}} - (y - y_l)}{d_c} \times (1 + |\cos\theta|) \\
t_w &= \mathbb{L}(\hat{w}, w), \quad t_h = \mathbb{L}(\hat{h}, h) \\
t_\theta &= (\hat{\theta} - \theta) \times \frac{w}{h} \\
\text{smooth}_{L_1}(x) &= \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \\
\mathbb{L}(x, y) &= \begin{cases} \log\left(\dfrac{x}{y}\right), & x >= y \\ \log\left(\dfrac{y}{x}\right), & x < y \end{cases}
\end{aligned} \tag{1}$$

where N_{R_1} is number of features pixels that locate in region R_1, i is the index of a feature pixel, $\mathbb{1}^{R_1}$ denotes the feature pixel in the region R_1, j is the index of the five parameters (x_s, y_s, w, h, and θ), $\hat{x_{r_s}}$ and $\hat{y_{r_s}}$ are the predicted relative coordinates, x and y are the ground truth of ship center coordinates, x_l and y_l are the coordinates of feature pixel, d_c is the side length of region R_1, and $(\hat{w}, \hat{h}, \hat{\theta})$ and (w, h, θ) denote the predicted value and ground truth of width, height, and angle of ship, respectively.

According to the contents discussed above, we can get $W_p \times H_p$ bounding boxes. However, since only the center region is selected from F_{boxes} to train SDet, not all the bounding boxes are accurate. To evaluate their accuracies, we need to predict an extra parameter score for each bounding box. A common way to get score is computing IoU between the predicted bounding box and the corresponding ground truth [27], [44], [45]. Nevertheless, it is a time-consuming process, especially in the training stage. To handle this problem, a simple way is to estimate whether a feature pixel corresponding to a bounding box is in the region R_1 or not. In other words, if the feature pixel is located in the region R_1, the score of its corresponding bounding box equals 1; otherwise, the score is equal to 0. However, it loses the information about the width and height of the ship since the region R_1 is a square, and we cannot assume that all pixels without R_1 would generate incorrect bounding boxes (scores are 0) for ship detection. Therefore, we develop a simple strategy to define the score label (the set of all the scores) with size of $W_p \times H_p$ in this article.

In detail, we first set the pixel value of score label within/without R_1 to be 1 and 0, and we record R_1 in the first box. Second, a subbox is generated by expanding the first box with the interval of one pixel, and the pixel value of score label within this subbox is added by 1. Third, repeat the second step until the size of the subbox is the same as the ground truth of the bounding box. Finally, the score label is normalized into [0, 1] as the final scores set. The graphic example of this procedure is shown in Fig. 5.

Therefore, the loss function of score for each HRRS image is formulated as

$$\begin{aligned}
\mathcal{L}_{\text{scores}} &= \frac{1}{N_{\mathcal{S}}} \sum_i (\mathcal{I}(s) \cdot \mathbb{L}(s, \hat{s}) - (1 - \mathcal{I}(s)) \cdot \log(1 - \hat{s})) \\
\mathcal{I}(x) &= \begin{cases} 1, & x >= 0 \\ 0, & x = 0 \end{cases}
\end{aligned} \tag{2}$$

where $N_{\mathcal{S}}$ denotes the pixels number of score label, i is the index of pixel, s and $\hat{s}$ denote the ground truth and predicted values of score, respectively, and $\mathbb{L}$ is the loss function mentioned above.

C. Center Detector

To predict the ship centers, the features F_{centers} with two channels are generated by convoluting the visual features P_1

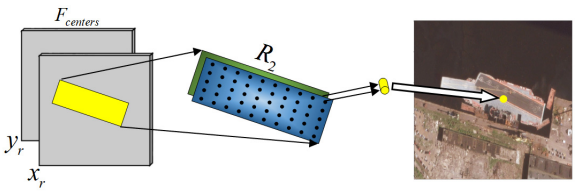

Fig. 6. Features F_{centers} with two channels. For each ship, the feature pixels within the region R_2 are used to be the training data for predicting the ship centers.

with the kernel size of 3×3 and the stride of 1×1. For each ship, we select a region R_2 (a rectangle that is surrounded by the ground truth of bounding box) from features F_{centers}, and all the features pixels with two channels of the region R_2 are trained for predicting the relative center coordinates x_r and y_r (shown in Fig. 6). Therefore, for each of features pixel in region R_2, the ground truth of x_r and y_r can be computed by subtracting the feature pixel coordinates $[x_l, y_l]$ from the ship center coordinates $[x, y]$.

The loss function of ship centers is defined as follows:

$$\begin{aligned}\mathcal{L}_{\text{centers}} &= \frac{1}{N_{R_2}} \sum_i \mathbb{1}^{R_2} \left(\sum_j \text{smooth}_{L_1}(t_j) \right) \\ t_{x_c} &= \frac{\hat{x_{r_c}} - (x - x_l)}{d_c} \times (1 + |\sin\theta|) \\ t_{y_c} &= \frac{\hat{y_{r_c}} - (y - y_l)}{d_c} \times (1 + |\cos\theta|) \end{aligned} \tag{3}$$

where N_{R_2} is the number of features pixels that locate in region R_2, i is the index of a feature pixel, $\mathbb{1}^{R_2}$ denotes the feature pixel in the region R_2, j is the index of the two parameters (x_c and y_c), $\hat{x_{r_c}}$ and $\hat{y_{r_c}}$ are the predicted relative coordinates, x and y are the ground truth of ship center coordinates, x_l and y_l are the coordinates of feature pixel, θ is the ground truth of angle of ship, d_c is the side length of region R_1, and smooth_{L_1} is the function mentioned above.

According to the mentioned above, we can get $W_p \times H_p$ $[x_r, y_r]$. To evaluate their accuracies, we predict a parameter confidence. The ground truth of confidence of each $[x_r, y_r]$ is 0/1, if the feature pixel that generates this $[x_r, y_r]$ is without/within the region R_2.

For learning confidence, we can adopt the standard cross entropy (CE) to be the loss function in general, and its definition is

$$-\frac{1}{n} \sum (c \cdot \log(\hat{c}) + (1 - c) \cdot \log(1 - \hat{c})) \tag{4}$$

where c and $\hat{c}$ are the ground-truth and predicted values of confidence and n equals the pixel number of features F_{centers}. However, the standard CE pays attention to ship and backgrounds equally, which is not suitable for our task. In the HRRS ship detection task, the volume of ships is much less than that of the backgrounds. The proper loss function should focus on the ships rather than the backgrounds. Therefore, we develop a new loss function for the prediction of confidence based on the standard CE. We name it biased CE (BCE) and its definition is

$$\begin{aligned}\mathcal{L}_{\text{confs}} &= -\frac{1}{n_t} \sum (c \cdot \log(\hat{c}) + (1 - c) \cdot \log(1 - \mathbb{T}(\hat{c}))) \\ \mathbb{T}(x) &= \begin{cases} 0, & x < t \\ x, & x \geq t \end{cases} \end{aligned} \tag{5}$$

where y_c and $\hat{y_c}$ are the ground-truth and predicted values of confidence, n_t equals the number of times that $\mathbb{T}(\hat{y_c})$ is not 0, and t is a hyperparameter that controls the influence of backgrounds. Similar to the standard CE, two terms of BCE aim at pushing the ships and backgrounds toward the positive and negative directions. Unlike the standard CE, due to the threshold scheme, our BCE pays more attention to the ships during the optimization so that the influence of the ships and backgrounds imbalanced problem can be reduced.

D. Training and the Inference With Score Adjustment

In the training stage, four parts should be optimized, which contains $\mathcal{L}_{\text{boxes}}$ and $\mathcal{L}_{\text{scores}}$ of SDet and $\mathcal{L}_{\text{centers}}$ and $\mathcal{L}_{\text{confs}}$ of CDet. In this article, we optimize them jointly and the loss function for training AR^{2}Det is formulated as follows:

$$\mathcal{L} = (\mathcal{L}_{\text{boxes}} + \mathcal{L}_{\text{scores}}) + (\mathcal{L}_{\text{centers}} + \mathcal{L}_{\text{confs}}). \tag{6}$$

In the inference stage, when users input an HRRS image into the trained AR^{2}Det, we can obtain the ships' bounding boxes, their scores, centers, and their confidences. Generally speaking, we can choose the bounding boxes according to their scores and overlap for ships (i.e., rotational nonmaximum suppression (NMS) [46]). Nevertheless, the incorrect scores may disturb the selection of bounding boxes. Therefore, we develop a score adjustment strategy to eliminate or adjust the noisy scores for improving the final detection results.

Here, we record the set of scores (generated by SDet) and relative coordinates (generated by CDet) S and C for clarity, and the size of S and C is $W_p \times H_p$. First, the hyperparameter t [see (5)] is used to be its threshold to select accurate relative coordinates $[x_r, x_r]$ from C. Then, for each of the selected relative coordinates, we can find the posterior feature pixel in C by adding feature pixel coordinates to the relative coordinates (we call this process as offset for short) and adjust a score of $\mathcal{S}$. Finally, only the adjusted scores will remain and others are set to be 0. The schematic of the two steps of score adjustment strategy is shown in Fig. 7. In the adjustment process, the posterior coordinate is getting closer and closer to the ship center, so its scores should be increased. If its score decreases, our strategy would replace its score value with the score of previous coordinate. At the same time, some incorrect high scores (i.e., the coordinates are far from the ship center, but their scores are high) will be eliminated by our strategy. Consequently, the remained scores are more "correct" for generating the final detection results. The simple processes of training and inference stages are described in Algorithms 1 and 2, respectively.

IV. Experiment Results

A. Dataset Introduction

The dataset HRSC2016 [47] is used to evaluate the proposed AR^{2}Det. It is a public HRRS image dataset for multiple

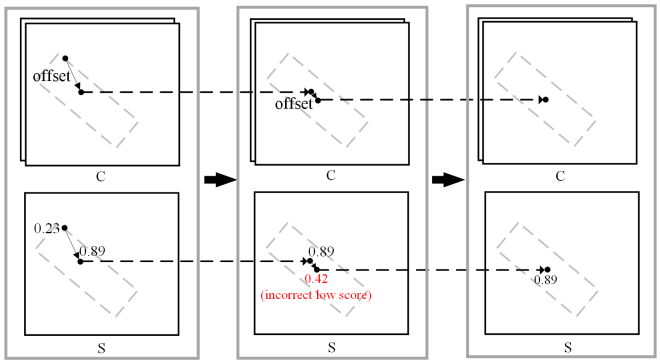

Fig. 7. Process of adjusting score set. The scores set is adjusted to obtain more accurate bounding boxes.

Algorithm 1 Training Process of AR^{2}Det

Input: The training Dataset; the side length d_c of R_1; the influence factor of background t; iteration number $\mathcal{N}$;

1: // Initialize the parameters of model and the score label;
2: **for** $n = 1 \rightarrow \mathcal{N}$ **do**
3: **Step1:** Extract features by ResNet and feature pyramid network;
4: **Step2:** Calculate the loss value according to Eq. 6;
5: **Step3:** Update the parameters of model by Adam optimizer;
6: **end**

Output: Trained AR^{2}Det.

Algorithm 2 Inference Process of AR^{2}Det

Input: The testing Dataset; the influence factor of background t; NMS threshold ξ; trained AR^{2}Det;

1: **Step1:** Extract features by ResNet and feature pyramid network;
2: **Step2:** Generate the bounding boxes and their scores and the ship centers and their confidence by SDet and CDet, respectively;
3: **Step3:** Adjust the scores of bounding boxes by the score adjustment strategy;
4: **Step4:** Generate the final detection results according to t and ξ;

Output: Detection results.

orientation ship detection, which was collected from Google Earth. The size of HRRS images within HRSC2016 ranges from 300×300 to 1500×900, and the ships within these images are either on the sea or near the inshore. In this dataset, the training, validation, and testing sets contain 436 images with 1207 samples, 181 images with 541 samples, and 444 images with 1228 samples, respectively.

B. Implementation Details

The training and inference are implemented by PyTorch [55] on a high-performance computer with GeForce RTX 2080 Ti and 11-GB memory. In the training stage, the ResNet of FEM is initialized by the pretrained parameters (using the ImageNet dataset [14]), and other parts of our network are initialized randomly. We employ the Adam optimizer [56] with eight images per minibatch, and the model is trained with the learning rate of 10^{-4} and the epoch number of 1000. In addition, the horizontal and the vertical flipping are adopted for the data augmentation.

In the following experiments, all the RS images are resized to 512×512 so that the sizes of features F_{boxes} and F_{centers} are equal to 128×128 ($W_p \times H_p$). Here, d_c (the side length of region R_1) is set to be 4.0, and the hyperparameter t [see (5)] is set to be 0.5. The threshold of overlap in the rotational NMS is equal to 0.5. The influence of different free parameters is discussed in Section IV-F.

Furthermore, we would like to explain how to choose appropriate values for two free parameters d_c and t. For d_c, its value could be decided according to the following two factors. First, its value should not be less than 2 to ensure that SDet can find ship centers accurately. Second, it should not be too large because the size of region R_1 is directly proportional to the volume of training data for AR2Det. Considering diverse experimental results, we empirically find that AR2Det achieves good performance when the ratio of d_c to the side length of F_{boxes} is in the range of 0.03–0.04. Therefore, we let d_c be 4 in this article because the size of F_{boxes} equals 128×128. For t, its value indicates the proportion of backgrounds to objects. To reduce the influence of backgrounds on ship detection, we suggest that the value of t could be close to 0.7/0.3 if the areas of backgrounds are more/less than that of ships distinctly. In the HRSC2016 dataset, the areas of ships and backgrounds are relatively balanced. Thus, we set t at 0.5. Readers could set these two variables according to the datasets they select.

C. Evaluation Metrics

To evaluate our AR^{2}Det numerically, we select two common assessment criteria, including the MAP and FPS. To calculate MAP, we should define the true positive (TP), false positive (FP), and false negative (FN) first. Generally, if the IoU value between a predicted bounding box and a ground truth of bounding box is greater than the threshold (we chose 0.5 in this article), then this predicted box can be as a TP; otherwise, it is regarded as an FP. Meanwhile, the redundant predicted boxes are also treated as FP. The ground truth of bounding boxes is the FN if they have not the matched predicted bounding boxes. According to TP, FP, and FN, recall and precision can be defined as

$$\begin{aligned} \text{precision} &= \text{TP}/(\text{TP}+\text{FP}) \\ \text{recall} &= \text{TP}/(\text{TP}+\text{FN}). \end{aligned} \tag{7}$$

Then, we can get average precision (AP) that is the area under the precision–recall curve (PRC). Furthermore, MAP is defined as the mean of AP across all the categories. The other evaluation index FPS represents the speed of the proposed AR^{2}Det during the operation process, and the postprocess (i.e., rotational NMS) is included in the operation process.

TABLE I
COMPARISONS WITH THE OTHER METHODS IN BOTH ACCURACY AND SPEED

Method	Type	Backbone	Imagesize	MAP (%)	Device	FPS
Fast-RCNN+SRBBS [20]	Two-stage	VGG16	-	55.70	-	-
BL2 [20]		VGG16	800*800	69.60	-	-
R^2CNN [32]		ResNet101	800*800	73.07	-	2
RC1&RC2 [20]		VGG16	-	75.7	-	< 1
RRPN [48]		ResNet101	800*800	79.08	Titan X	3.5
R^2PN [19]		VGG16	-	79.6	-	< 1
R-DFPN [49]		ResNet101	600*600	85	Titan Xp	11
RoI-Transformer [33]		ResNet101	512*800	86.2	RTX 2080 Ti	12
Gliding Vertex [34]		ResNet101	-	88.2	Titan Xp	10
OPLD [50]		ResNet50	1024*1024	88.44	TITAN X	7.3
IENet [51]	One-stage	ResNet101	1024*1024	75.01	GTX 1080 Ti	17
Rotated YOLO-v2 [23]		Darknet19	600*600	75.6	Titan Xp	55
TOSO [52]		ResNet101	800*800	79.29	RTX 2080 Ti	17
DRBox [39]		VGG net [39]	600*600	81.4	Titan Xp	69
RetinaNet-H [17]		ResNet101	800*800	82.89	RTX 2080 Ti	14
RRD [53]		VGG16	384*384	84.3	Titan Xp	10
R^3Det [17]		MobileNetV2	300*300	86.67	RTX 2080 Ti	20
R^3Det [17]		ResNet101	300*300	87.14	RTX 2080 Ti	18
S^2ARN [40]		ResNet50	600*600	88.1	Titan Xp	32
GRS-Det [54]		ResNet50	800*800	88.90	GTX 1080 Ti	17
RetinaNet-R [17]		ResNet101	800*800	89.18	RTX 2080 Ti	10
R^3Det [17]		ResNet152	800*800	89.33	RTX 2080 Ti	10
GRS-Det [54]		ResNet101	800*800	89.57	GTX 1080 Ti	14
AR^2Det(proposed)	One-stage	ResNet34	512*512	**89.58**	RTX 2080 Ti	112
		ResNet18	512*512	81	RTX 2080 Ti	**129**

D. Results on HRSC2016

To study the performance of AR^2Det, we select the following two- and one-stage methods as the compared methods. The two-stage methods contain Fast-RCNN + SRBBS [20], BL2 [20], R^2CNN [32], RC1&RC2 [20], RRPN [48], R^2PN [19], R-DFPN [49], RoI-Transformer [33], Gliding Vertex [34], and OPLD [50]. The one-stage methods include IENet [51], Rotated YOLO-v2 [23], TOSO [52], DRBox [39], RetinaNet-H [17], RRD [53], S^2ARN [40], RetinaNet-R [17], R^3Det [17], and GRS-Det [54]. Note that, in the following experiment, ResNet34 and ResNet18 are adopted as our backbone to deeply analyze the behavior of our AR^2Det.

The results of performance comparison are shown in Table I. It is easy to find that most of the time, our AR^2Det outperforms other compared methods in both the accuracy (MAP) and the speed (FPS). For two-stage methods, twice detection mechanism leads them to obtain satisfactory accuracy, and RoI-Transformer [33], Gliding Vertex [34], and OPLD [50] achieve the superior performance. However, their detection speed is not satisfactory enough. The fastest method (RoI-Transformer [33]) can only achieve 12 FPS. For one-stage methods, they obtain the predicted results with once detection process so that the speed can be increased. The fastest compared methods (Rotated YOLO-v2 [23]) can up to 69 FPS. Meantime, with the appearance of the new ideas, their behavior is getting strong increasingly. For example, MAP values of RetinaNet-R [17] with ResNet101, R^3Det [17] with ResNet152, and GRS-Det with ResNet101 are 89.18%, 89.33%, and 89.57%, respectively.

Although the compared methods perform well, our AR^2Det can still achieve better performance, no matter the aspects of Map values and FPS. For instance, compared with the classical two-stage detection method RC1&RC2 [20], our model's (with ResNet34) MAP values are increased by 18.36% and its FPS is more than 110 times as much with RC1&RC2 [20]. For another example, compared with two-stage methods, RoI-Transformer [33], Gliding Vertex [34], and OPLD [50], the enhancements achieved by our AR^2Det (with ResNet34) in MAP are 3.38%, 1.38%, and 1.14%, respectively. Also, our model's detection speed is almost ten times as much with RoI-Transformer, Gliding Vertex, and OPLD. For one-stage methods, taking three advanced models RetinaNet-R [17], R^3Det [17], and GRS-Det [54] as examples, the improvements of AR^2Det (with ResNet34) in MAP are 0.4%, 0.25%, and 0.01%, and our FPS is more than about ten times as much

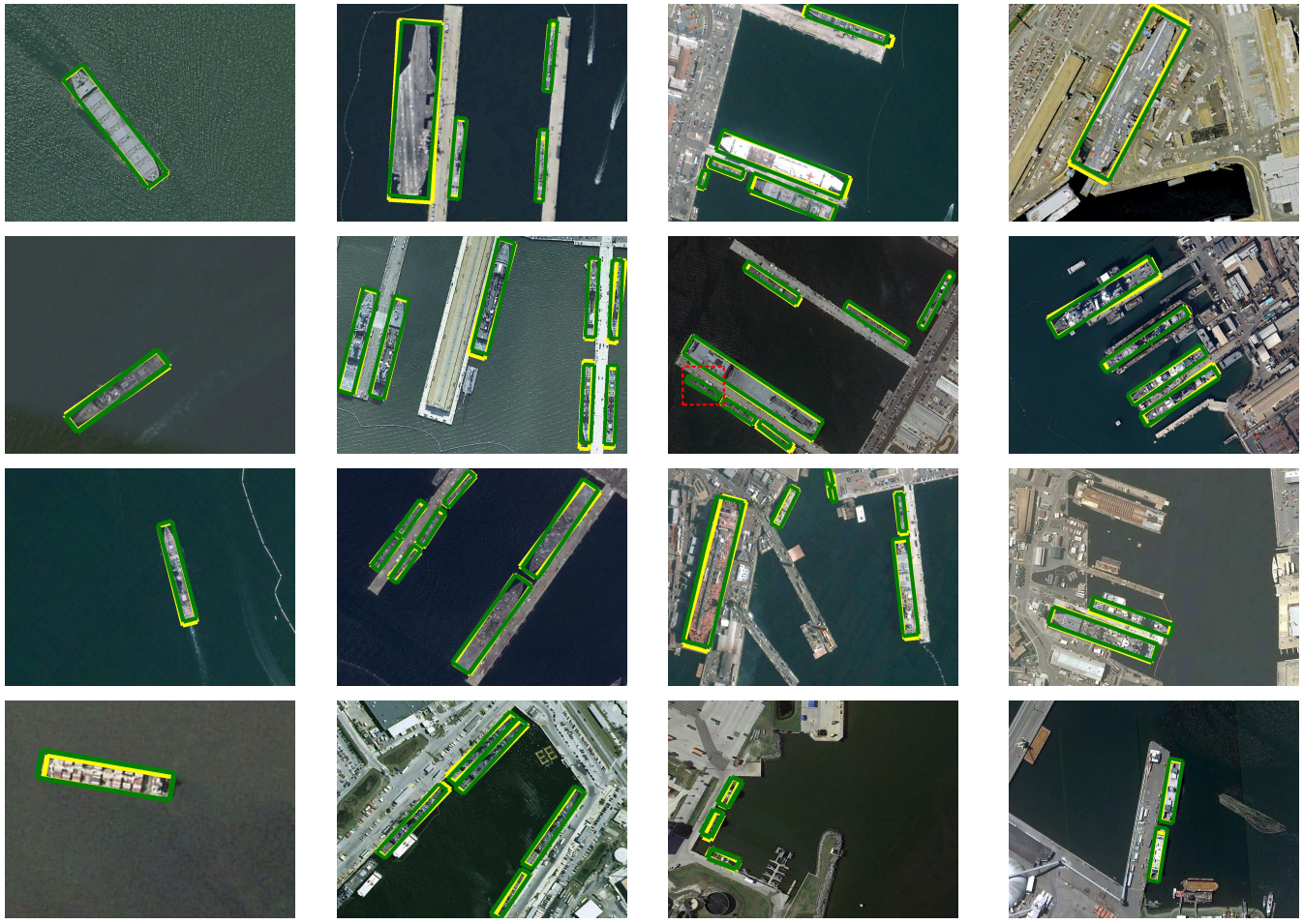

Fig. 8. Predicted results of the proposed AR^{2}Det in four different scenarios. (a) Ships on the sea. (b) Ships close inshore. (c) Ships with multiple scales. (d) Ships with complex background. The ground-truth and prediction results of bounding boxes are marked with green and yellow, respectively. Also, the red bounding boxes are utilized to denote the incorrect prediction of ships.

with three compared networks. The encouraging experiments discussed above confirm that the proposed AR^{2}Det is effective in ship detection.

There is another point we want to touch on, that is, different from most of the compared methods, the backbone of our AR^{2}Det is a relative light network (ResNet34). Nevertheless, our model can still achieve the best performance that proves the usefulness of AR^{2}Det again. The reasons why AR^{2}Det performs the best can be summarized as follows. First, the light backbone network and the simple bounding boxes prediction scheme ensure the efficiency of AR^{2}Det. Second, the introduced score label and developed BCE loss guarantee the accuracy of our model. Finally, the developed scores adjustment strategy in the inference stage can further improve the performance of AR^{2}Det. Besides the accuracy and speed, our model also has an advantage in the volume of parameters. The scale of parameters (Params) of AR^{2}Det is shown in Table II. We also report Params of two two-stage methods and two one-stage methods for reference. The selected four compared methods have stronger behavior than others. It is easy to find that the Params of AR^{2}Det is less than 50% of that of the other methods.

TABLE II
COMPARISONS IN THE PARAMS

Method	Type	Backbone	Params	MAP (%)
RRPN [48]	Two-stage	ResNet101	181MB	79.08
RoI-Transformer [33]		ResNet101	273MB	86.2
IENet [51]	One-stage	ResNet101	212MB	75.01
R^{3}Det [17]		ResNet101	227MB	87.14
AR^{2}Det(proposed)	One-stage	ResNet34	90MB	**89.58**
		ResNet18	**49MB**	81

Apart from the numerical assessment, we also provide some ship detection examples visually in Fig. 8. There are four columns of detection results. The ships in the first column images are on the sea and the ships in the second column images are near the inshore. The third column images illustrate the performance of AR^{2}Det under the ships with multiple-scale scenario, whereas the fourth column images demonstrate the behavior of AR^{2}Det under the ships

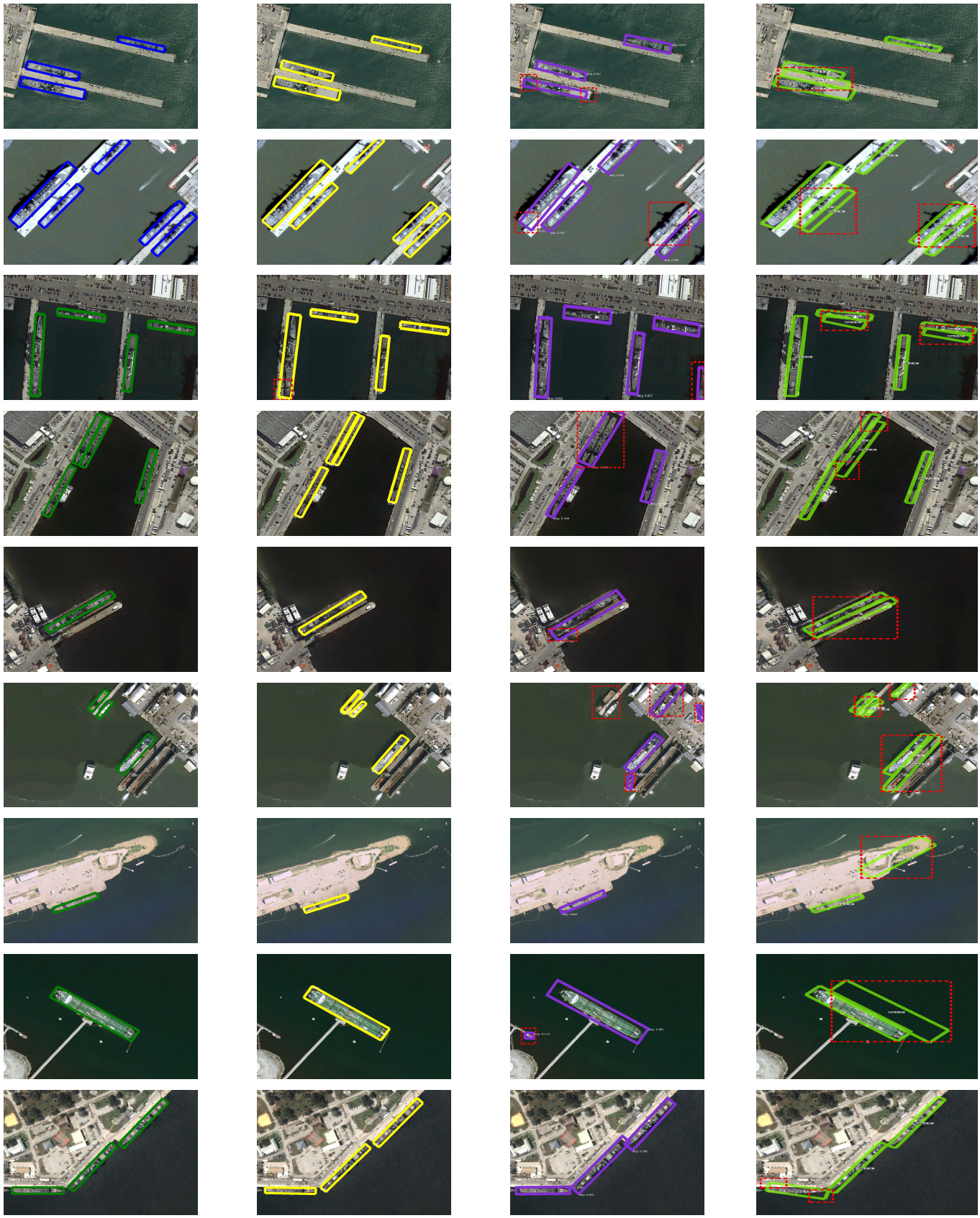

Fig. 9. Visual detection results of different methods on the HRSC2016 dataset. (a) Ground Truth. (b) AR^{2}Det. (c) RoI-Transformer [33]. (d) R^{3}Det [17]. The red bounding boxes are utilized to denote the incorrect and inaccurate prediction of ships.

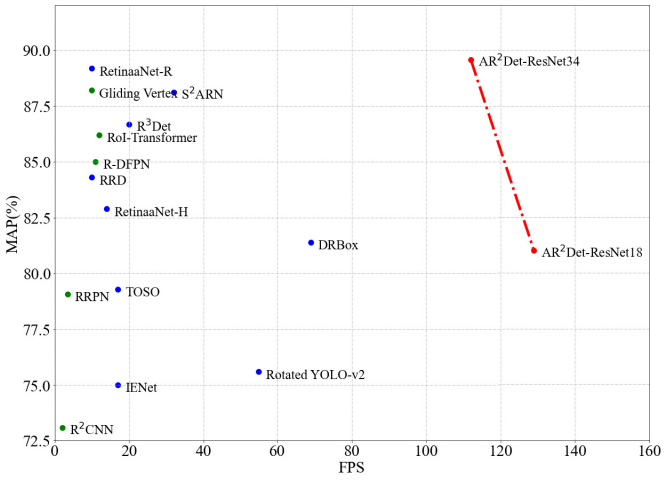

Fig. 10. Comparison with other methods in both MAP and FPS. The blue and green points represent the one-stage and the two-stage methods respectively, and the red points indicate the behavior of our AR^{2}Det with different backbone networks. It is easy to find that the proposed AR^{2}Det outperforms others methods in speed and accuracy.

TABLE III
CONSTRUCTED MODELS AND THEIR COMPONENTS

Model	FEM	SDet	RC	SL	CDet	MAP (%)
Model$_1$	✓	✓	×	×	×	76.19
Model$_2$	✓	✓	✓	×	×	80.31
Model$_3$	✓	✓	✓	✓	×	81.79
Model$_4$	✓	✓	✓	✓	✓	89.57

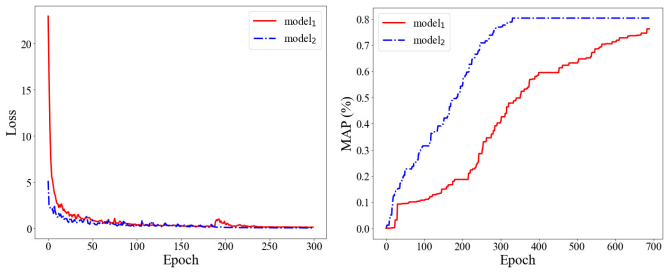

Fig. 11. Diagram about MAP and loss values in the different epoch. We can find that the performance of model$_2$ is better.

with complex background. The ground-truth and prediction results of bounding boxes are marked with green and yellow, respectively. Also, the red bounding boxes are utilized to denote the incorrect prediction of ships. According to the results, we can find that AR^{2}Det has good robustness and performance. In addition, the visual detection results of three methods are shown in Fig. 9, including the proposed AR^{2}Det, the best two-stage method (RoI-Transformer) [33], and the best one-stage method (R^{3}Det) [17]. It is noticed that AR^{2}Det has distinct advantages. To illustrate the performance of our method more clearly, we draw the scatter diagram to illustrate the performance of different methods (both MAP and FPS) in Fig. 10, where blue and green points represent the one- and two-stage methods, respectively, and the red points indicate the behavior of our AR^{2}Det with different backbone networks. It is worth noting that the proposed method is superior to other methods in both speed and accuracy.

E. Ablation Study

In the AR^{2}Det, SDet with FEM can be regarded as a basic detection model. However, its performance is not satisfactory. Therefore, based on the basic model, we propose three blocks to improve the performance of SDet, including the relative coordinates (RCs) for improving the accuracy of the predicted ship center coordinates, the scores label (SL) for optimizing the ground truth of scores, and CDet for adjusting the predicted scores. To study their contributions, four models are constructed to complete ship detection, and their components and accuracy are shown in Table III. We implement the experiments of four models based on the ResNet34 and keep the other components the same in the training and inference.

For RC, compared with the absolute coordinates, it utilizes the position information of feature pixel to search the ship center in a smaller range. It makes the model$_2$ can find ship centers rapidly and accurately. Therefore, the accuracy is increased significantly from 76.19% to 80.31%. We also draw the curve of the loss and MAP about the model$_1$ and model$_2$ in Fig. 11. It shows that RC is superior to the absolute coordinates in both the convergence rate and MAP growth rate simultaneously.

For SL, it is proposed to optimize the ground truth of scores in SDet. SL expands the scores from region R_1 to the ground truth of bounding box. The overlap can be computed more precisely, and the information of the width and height of ship can be saved effectively. Therefore, the performance of model$_3$ (MAP) is increased from 80.31% to 81.79%. In addition, the speed of training can be accelerated from 4.72 to 37.36 FPS (about eight times) since the SL abandons the complex computation of the IoU.

For the CDet, it is developed to adjust the scores of SDet in the inference, which can eliminate the inaccurate bounding boxes effectively. Therefore, the detected accuracy of model$_4$ can be improved from 81.79% to 89.57%. To demonstrate the reason why CDet works, we draw four curves of the precision and recall in Fig. 12 according to whether the adjustment (score adjustment strategy) is applied or not on the predicted scores and the true scores (ground truth of scores). Here, considering the accuracy, the ground truth of scores is obtained by computing the IoU between the ground truth of bounding boxes and the predicted bounding boxes. From the observation of four curves, the accuracy of the true and predicted scores is improved dramatically although the adjustment leads the recall rate to decrease slightly. In Table IV, we list the numerical results, where the predicted number and the true number represent the number of bounding boxes. It is worth noting that CDet removes the most of inaccurate bounding boxes successfully. Consequently, the predicted number is closer to the true number, and the accuracy is improved greatly. It proves that CDet is an effective block.

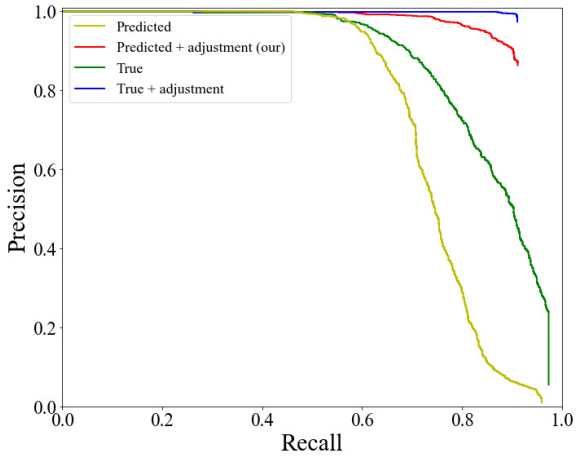

Fig. 12. Curves of precision and recall according to whether the adjustment (scores adjustment strategy) is applied or not on the predicted scores and the true scores. It is worth noting that adjustment improves the accuracy in both the true and predicted scores significantly although the highest recall rate decreases slightly.

TABLE IV
THE NUMBERS OF BOUNDING BOXES AND MAP VALUES WITH/WITHOUT CDET

Scores type	Correction	Predicted number	True number	MAP (%)
Predicted	×	108968	1228	72.92
Predicted	✓	**1296**		89.58
True	×	21753		82.53
True	✓	1148		**90.80**

TABLE V
MAP WITH THE CHANGE OF SIDE LENGTH d_c

d_c	1.0	2.0	4.0	6.0	8.0
MAP (%)	88.23	88.94	**89.58**	89.25	89.22

F. Sensitivity Analysis

To analyze the effect of hyperparameters of AR^{2}Det, two experiments are conducted, i.e., d_c (the side length of region R_1) and the threshold t of BCE. To ensure the stability of experiments, we change d_c when fixing the value of t to 0.5. Analogously, t is changed when the value of d_c is fixed to 4.0. All the other settings remain the same in our experiments.

1) Side Length of Region R_1: The change of d_c can influence the performance of AR^{2}Det slightly since d_c decides the number of the feature pixels that are trained for predicting the ships. Thus, we select five different values of d_c (1.0, 2.0, 4.0, 6.0, and 8.0) to study the sensitivity of d_c. Table V shows the final results of MAP. It is obvious that the model has the best accuracy when d_c is set to 4.0. However, setting d_c to 1.0 (i.e., the region R_1 is shrunken to a point) can result in a significant decrease in the final accuracy. It proves that selecting a region from features for predicting bounding boxes is more effective compared with selecting a point.

2) Hyperparameter t of BCE: BCE is utilized to compute loss for training CDet. As the hyperparameter of BCE, t can control how much attention is paid to the background. Here, two special values of t need to be explained. First, setting t to 0.0 would degenerate BCE into the standard CE. Second, setting t to 1.0 would disable CDet. Therefore, we set t from 0.1 to 0.9 with the interval of 0.2 for studying the sensitivity of t, and the results are shown in Table VI. According to the results, the optimal value of t is 0.5. Compared with the standard CE, BCE improves the accuracy from 87.06% to 89.58% when t is set to 0.5.

TABLE VI
MAP WITH THE CHANGE OF t

t	0.0 (CE)	0.1	0.3	0.5	0.7	0.9
MAP (%)	87.06	88.90	89.30	**89.58**	89.47	89.12

In addition, we explore the reason why the optimal value of t is 0.5. First, AR^{2}Det is initiated randomly. Then, the HRRS images are input to AR^{2}Det, and the mean value of predicted confidences (MPC) of CDet is counted. Finally, the first two steps are repeated 400 times. We find that the mean of MPC is 0.5004464, which means that the initial values of confidences are close to 0.5. Therefore, AR^{2}Det can be trained rapidly and simply for distinguishing the ships from backgrounds when t is set to 0.5.

V. CONCLUSION

In this article, considering the characteristics of HRRS images, we have proposed an end-to-end one-stage ship detection method named AR^{2}Det, which can be used to accomplish the rotational ship detection task rapidly accurately. To ensure detection accuracy, two submodels (SDet and CDet) have been developed for generating the bounding boxes and enhancing their quality. To keep the high speed of detection, AR^{2}Det adopts a lightweight network and simple postprocessing scheme. Experimental results on the public ship detection dataset HRSC2016 have demonstrated the effectiveness and efficiency of AR^{2}Det in the HRRS ship detection task.

Although our AR^{2}Det is developed for HRRS ship detection, it can also be used to detect multiclass targets from HRRS images. To justify this, we have investigated the proposed model on the NWPU VHR-10 dataset [57]. The encouraging experimental results demonstrate that AR^{2}Det can handle more challenging detection tasks. The details of multiclass detection experiments can be found in the Supplementary Material. From the observation of detection results counted on NWPU VHR-10, we can find that the performance of the proposed model is acceptable, although there is still a room to further enhance its performance. Therefore, we will extend AR^{2}Det to multiclass HRRS object detection tasks in the future.

REFERENCES

[1] W. Guo, W. Yang, H. Zhang, and G. Hua, "Geospatial object detection in high resolution satellite images based on multi-scale convolutional neural network," *Remote Sens.*, vol. 10, no. 1, p. 131, Jan. 2018.
[2] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1797–1801, Oct. 2014.

[3] M. Lan, Y. Zhang, L. Zhang, and B. Du, "Global context based automatic road segmentation via dilated convolutional neural network," *Inf. Sci.*, vol. 535, pp. 156–171, Oct. 2020.

[4] Z. Li and L. Itti, "Saliency and gist features for target detection in satellite images," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 2017–2029, Jul. 2011.

[5] 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.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[7] Q. Zhu, Y. Zhong, L. Zhang, and D. Li, "Adaptive deep sparse semantic modeling framework for high spatial resolution image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 6180–6195, Oct. 2018.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

[9] 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.

[10] C. Dong, J. Liu, and F. Xu, "Ship detection in optical remote sensing images based on saliency and a rotation-invariant descriptor," *Remote Sens.*, vol. 10, no. 3, p. 400, Mar. 2018.

[11] B. Hou, W. Yang, S. Wang, and X. Hou, "SAR image ship detection based on visual attention model," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2013, pp. 2003–2006.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[13] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[16] 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.

[17] X. Yang, J. Yan, Z. Feng, and T. He, "R3Det: Refined single-stage detector with feature refinement for rotating object," 2019, *arXiv:1908.05612*. [Online]. Available: http://arxiv.org/abs/1908.05612

[18] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 117, pp. 11–28, Jul. 2016.

[19] 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.

[20] 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.

[21] L. Liu *et al.*, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Jan. 2020.

[22] 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.

[23] W. Liu, L. Ma, and H. Chen, "Arbitrary-oriented ship detection framework in optical remote-sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 6, pp. 937–941, Jun. 2018.

[24] 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.

[25] Y. Cao, X. Niu, and Y. Dou, "Region-based convolutional neural networks for object detection in very high resolution remote sensing images," in *Proc. 12th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery (ICNC-FSKD)*, Aug. 2016, pp. 548–554.

[26] X. Han, Y. Zhong, R. Feng, and L. Zhang, "Robust geospatial object detection based on pre-trained faster R-CNN framework for high spatial resolution imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 3353–3356.

[27] 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.

[28] Y. Wang, Z. Dong, and Y. Zhu, "Multiscale block fusion object detection method for large-scale high-resolution remote sensing imagery," *IEEE Access*, vol. 7, pp. 99530–99539, 2019.

[29] Z. Dong, M. Wang, Y. Wang, Y. Zhu, and Z. Zhang, "Object detection in high resolution remote sensing imagery based on convolutional neural networks with suitable object scale features," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 2104–2114, Mar. 2020.

[30] J. Pang, C. Li, J. Shi, Z. Xu, and H. Feng, "$\mathcal{R}^2$-CNN: Fast tiny object detection in large-scale remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5512–5524, Mar. 2019.

[31] M.-E. Tang, T.-Q. Lin, and G.-J. Wen, "Overview of ship detection methods in remote sensing image," *Jisuanji Yingyong Yanjiu*, vol. 28, no. 1, pp. 29–36, 2011.

[32] 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

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

[34] Y. Xu *et al.*, "Gliding vertex on the horizontal bounding box for multi-oriented object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1452–1459, Apr. 2021.

[35] A. Van Etten, "You only look twice: Rapid multi-scale object detection in satellite imagery," 2018, *arXiv:1805.09512*. [Online]. Available: http://arxiv.org/abs/1805.09512

[36] B. Hou, Z. Ren, W. Zhao, Q. Wu, and L. Jiao, "Object detection in high-resolution panchromatic images using deep models and spatial template matching," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 956–970, Feb. 2020.

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

[38] P. Wang, X. Sun, W. Diao, and K. Fu, "FMSSD: Feature-merged single-shot detection for multiscale objects in large-scale remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3377–3390, May 2020.

[39] L. Liu, Z. Pan, and B. Lei, "Learning a rotation invariant detector with rotatable bounding box," 2017, *arXiv:1711.09405*. [Online]. Available: http://arxiv.org/abs/1711.09405

[40] S. Bao, X. Zhong, R. Zhu, X. Zhang, Z. Li, and M. Li, "Single shot anchor refinement network for oriented object detection in optical remote sensing imagery," *IEEE Access*, vol. 7, pp. 87150–87161, 2019.

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

[42] 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.

[43] 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. (ICCV)*, Oct. 2019, pp. 6569–6578.

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

[45] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[46] J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4507–4515.

[47] 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.

[48] J. Ma *et al.*, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, Nov. 2018.

[49] 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.

[50] Q. Song, F. Yang, L. Yang, C. Liu, M. Hu, and L. Xia, "Learning point-guided localization for detection in remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1084–1094, 2021.

[51] Y. Lin, P. Feng, J. Guan, W. Wang, and J. Chambers, "IENet: Interacting embranchment one stage anchor free detector for orientation aerial object detection," 2019, *arXiv:1912.00969*. [Online]. Available: http://arxiv.org/abs/1912.00969
[52] P. Feng, Y. Lin, J. Guan, G. He, H. Shi, and J. Chambers, "TOSO: Student's-T distribution aided one-stage orientation target detection in remote sensing images," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 4057–4061.
[53] M. Liao, Z. Zhu, B. Shi, G.-S. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5909–5918.
[54] X. Zhang, G. Wang, P. Zhu, T. Zhang, C. Li, and L. Jiao, "GRS-det: An anchor-free rotation ship detector based on Gaussian-mask in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 3518–3531, Apr. 2021.
[55] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.
[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[57] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.

Yuqun Yang (Student Member, IEEE) received the B.Sc. degree in information and computing science from the Xi'an University of Technology, Xi'an, China, in 2019. He is pursuing the Ph.D. degree with the School of Artificial Intelligence, Xidian University, Xi'an.

His research interests include machine learning, object detection, and image classification.

Xu Tang (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electronic circuit and system from Xidian University, Xi'an, China, in 2007, 2010, and 2017 respectively.

From 2015 to 2016 he was a joint Ph.D. student along with Prof. W. J. Emery at the University of Colorado at Boulder, Boulder, CO, USA. He is an Associate Professor with the Key Laboratory of Intelligent Perception and Image Understanding, Ministry of Education, Xidian University. His research interests include remote sensing image content-based retrieval and reranking, hyperspectral image processing, remote sensing scene classification, and object detection. For more details, please refer to: https://web.xidian.edu.cn/tangxu/

Yiu-Ming Cheung (Fellow, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, in 2000.

He is a Full Professor with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His research interests include machine learning, data science, pattern recognition, visual computing, and optimization.

Dr. Cheung is also a fellow of the International Engineering and Technology Institute (IET) and the British Computer Society (BCS). He serves as an Associate Editor for several prestigious journals, including IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and *Pattern Recognition*, to name a few. For more details, please refer to http://www.comp.hkbu.edu.hk/%7eymc

Xiangrong Zhang (Senior Member, IEEE) received the B.S. and M.S. degrees from the School of Computer Science, Xidian University, Xi'an, China, in 1999 and 2003, respectively, and the Ph.D. degree from the School of Electronic Engineering, Xidian University, in 2006.

From January 2015 to March 2016, she was a Visiting Scientist with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA. She is a Professor with the Key Laboratory of Intelligent Perception and Image Understanding, Ministry of Education, Xidian University. Her research interests include pattern recognition, machine learning, and remote sensing image analysis and understanding.

Fang Liu (Member, IEEE) was born in China, in 1990. She received the B.S. degree in information and computing science from Henan University, Kaifeng, China, in 2012, and the Ph.D. degree in intelligent information processing from Xidian University, Xi'an, China, in 2018.

She is a Lecturer with the Nanjing University of Science and Technology, Nanjing, China. Her research interests include deep learning, object detection, polarimetric synthetic aperture radar (SAR) image classification, and change detection.

Jingjing Ma (Member, IEEE) received the B.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2004 and 2012, respectively.

She is an Associate Professor of the Key Laboratory of Intelligent Perception and Image Understanding, Ministry of Education, Xidian University. Her research interests include computational intelligence and image understanding.

Licheng Jiao (Fellow, IEEE) received the B.S. degree in high voltage from Shanghai Jiao Tong University, Shanghai, China, in 1982, and the M.S. and Ph.D. degrees in electronic engineering from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

From 1984 to 1986, he was an Assistant Professor with the Civil Aviation Institute of China, Tianjin, China. From 1990 to 1991, he was a Post-Doctoral Fellow with the Key Laboratory for Radar Signal Processing, Xidian University, Xi'an, where he is the Director of the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China. He has authored or coauthored more than 200 scientific articles. His research interests include signal and image processing, nonlinear circuits and systems theory, wavelet theory, natural computation, and intelligent information processing.

Dr. Jiao is also a member of the IEEE Xian Section Executive Committee and an Executive Committee Member of the Chinese Association of Artificial Intelligence. He is also the Chairman of the Awards and Recognition Committee.