A Novel Object Detector Based on High-Quality Rotation Proposal Generation and Adaptive Angle Optimization

Yajun Qiao, Lingjuan Miao, Zhiqiang Zhou, *Member, IEEE*, and Qi Ming

***Abstract*— Currently, reliable and accurately oriented detection in remote sensing images still needs to be improved. The wide variation of object shapes and orientations in the remote sensing images usually leads to two issues in two-stage oriented object detectors. One issue is how to generate high-quality rotation proposals. The other is the angular error sensitivity to the aspect ratios in the angle optimization process. In this article, we propose a novel rotation proposal generation and optimization detector, which is based on high-quality rotation proposal generation and adaptive angle optimization to solve these two issues. The proposed method mainly establishes the geometric relationship guided region proposal networks (GRG-RPNs) and the adaptive angle optimization head (AAO-Head) to achieve more accurate oriented object detection. The GRG-RPN only uses a simple network and a small number of horizontal anchors to predict high-quality rotation proposals. This approach was derived via the calculation based on the theoretical analysis of the geometric relationship between the oriented bounding boxes (OBBs) and their external horizontal bounding boxes (EHBBs). The AAO-Head solves the angular error sensitivity to the aspect ratios and achieves adaptive angle optimization using a new regression parameter, which is defined based on the theoretical analysis of the relationship between the intersection over union (IoU), the angular errors, and the aspect ratios. The experiments show that our method can achieve a 2.5% mAP improvement average versus the compared state-of-the-art (SOTA) methods, and achieve 0.5% mAP improvement versus the next best method-oriented regions with CNN features (R-CNN) with fewer regression parameters and a simpler regression approach.**

***Index Terms*— Adaptive angle optimization head (AAO-Head), geometric relationship guided region proposal network (GRG-RPN), oriented object detection.**

I. Introduction

THE technique of oriented object detection in remote sensing images can find essential applications in various tasks, for example, city planning, traffic service, fishery management, and military surveillance [1], [2], [3]. The tremendous amount of remote sensing images available in recent years has made oriented object detection more attractive for relevant applications and caught the great attention of researchers.

In recent years, encouraged by the great success of convolutional neural networks (CNNs) and deep-learning-based object detection in natural images [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], many researchers have proposed to utilize similar methodologies for oriented object detection. However, unlike most objects in natural images, the objects in remote sensing images vary widely in shape and orientation. In addition, these objects have particular properties making them harder to be accurately detected compared with general objects and easier to suffer from degradation caused by densely packed objects, mutually obscuring objects, and objects in different scales, as well as the impact from the complex background.

Most oriented object detectors can achieve higher accuracy based on the two-stage approach that contains a region proposal network (RPN) and a detection head [5], [6], [14], [15]. The RPN usually needs to predict the offsets of the proposals relative to the anchors and make a classification between the foreground and background. The detection head achieves more accurate regression and classification based on rotation proposals. The RPN usually needs transformation to get the final rotation proposals and ground-truth offsets of the ground-truth boxes relative to the anchors. The transformation uses the anchors and ground-truth boxes to parameterize the ground-truth offsets, which are used as supervisory information for training. The rotation proposals are obtained by decoding the predicted offsets and the anchors for training and testing, which can be seen as the reverse process of the parameterization. Although current-oriented object detectors can achieve compelling results, there is still much room for improvement compared with their counterparts for object detection in remote sensing images.

One significant issue in oriented object detection is how to generate rotation proposals. The rotation proposals are mainly obtained by the decoding process of the RPN in the two-stage oriented object detectors. Fig. 1(a) and (b) shows two representative RPNs, i.e., the rotated RPN [16] and the RoI transformer [17]. The rotated RPN gets the rotation proposals based on the predicted offsets and the rotated anchors. These rotated anchors are defined with different angles, scales, and aspect ratios. The massive defined rotated anchors will drastically increase the calculation and the imbalance between positive and negative samples, eventually leading to slow convergence speed and suboptimal results in oriented object detection. The RoI transformer gets the rotation proposals by two decoding processes. The first decoding process obtains the external horizontal bounding boxes (EHBBs) based on the horizontal anchors, and the second decoding process gets the

Manuscript received 25 May 2023; revised 21 July 2023; accepted 30 July 2023. Date of publication 3 August 2023; date of current version 16 August 2023. *(Corresponding author: Zhiqiang Zhou.)*

The authors are with the School of Automation, Beijing Institute of Technology, Beijing 100081, China (e-mail: yajun@bit.edu.cn; miaolingjuan@bit.edu.cn; zhzhzhou@bit.edu.cn; chaser.ming@gmail.com).

Digital Object Identifier 10.1109/TGRS.2023.3301610

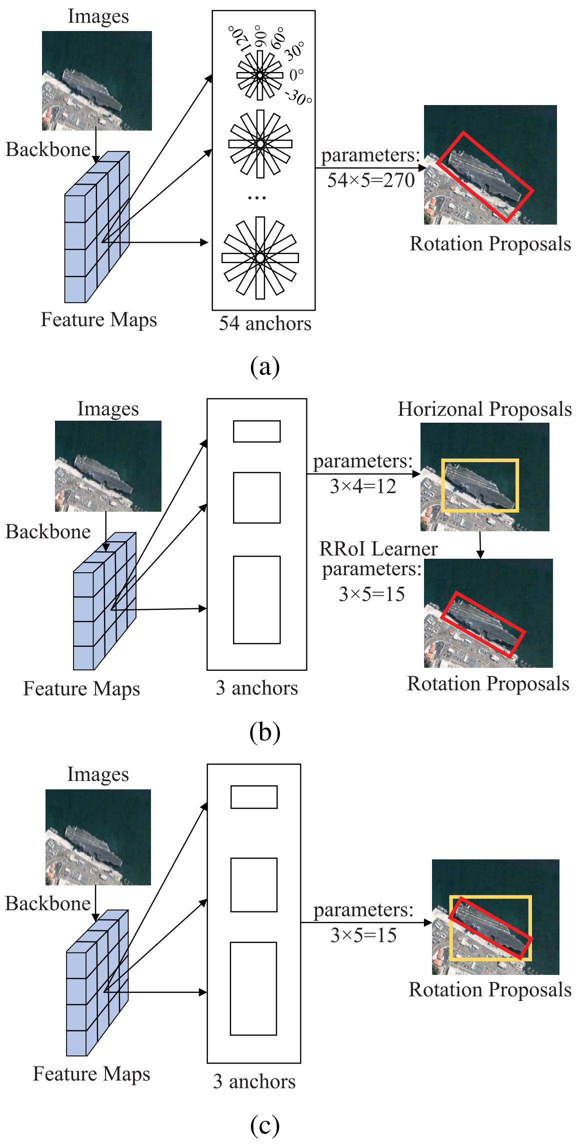


Fig. 1. Comparison of RPNs in different methods. (a) Rotated RPN. (b) RoI transformer. (c) GRG-RPN.

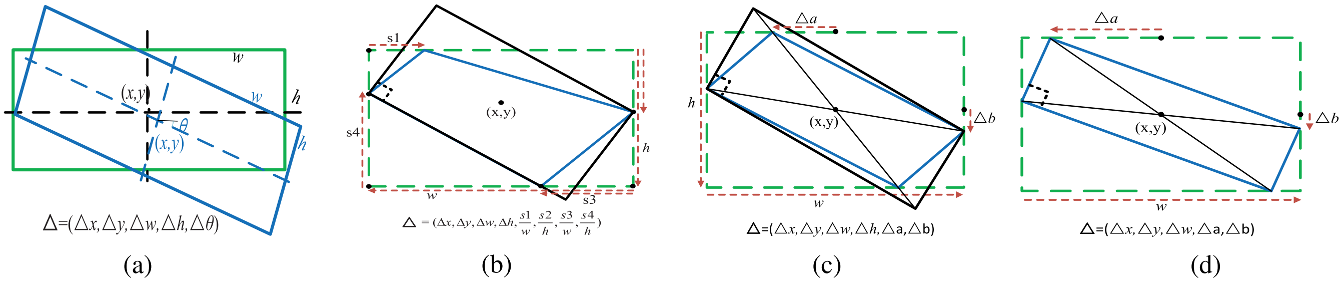
rotation proposals based on the EHBBs and the outputs of the RROI learner. This method reduces the number of anchors compared with the rotated RPN, but it needs two decoding processes to get the final rotation proposals. As shown in Fig. 1(b), the RoI transformer needs an intermediate process and 12 additional parameters to obtain the oriented bounding boxes (OBBs). This no-direct transformation would increase the computational complexity and degrade the quality of rotation proposal generation. In this article, we derive a direct transformation for rotation proposal generation based on the geometric relationship between the OBBs and their EHBBs and propose a novel geometric relationship guided RPN (GRG-RPN), which can use only five parameters to generate the high-quality rotation proposals directly from the horizontal anchors in one step [as shown in Fig. 1(c)]. The proposed GRG-RPN only needs a small number of predefined horizontal anchors for rotation proposal generation and can produce high-quality rotation proposals using a simple proposal network based on the derived transformation.

In addition, Fig. 2 shows some methods that generate rotation proposals from the horizontal anchors. Compared with the different approaches, our method can avoid the positioning errors induced by direct regression of rotation angle [faster R-CNN-O [6] as shown in Fig. 2(a)], and needs fewer steps and regression parameters compared with the other methods such as gliding vertex [14] and oriented R-CNN [18] [as shown in Fig. 2(b) and (c)]. We can see that the gliding vertex and oriented regions with CNN features (R-CNN) both need external processes to obtain the final rotation proposals, i.e., the gliding vertex needs to first generate the irregular quadrilaterals, while oriented R-CNN has to generate the parallelograms first before obtaining the final rotation rectangles. Our method can generate the rotation rectangles in one step and with fewer regression parameters compared with gliding vertex and oriented R-CNN, which can help to avoid the possible errors introduced by extra processes and obtain more reliable regression relying on fewer parameters.

Besides, the issue of the angular error sensitivity to the aspect ratios also needs to be considered during the angle optimization of the bounding boxes. As shown in Fig. 3, when the aspect ratios of OBBs change from 1 to 8 with identical angular errors of 10°, the intersection over union (IoU) will change from 0.86 to 0.49, which reveals the inconsistency between the angular error and IoU. Fig. 4 further shows the variation curves of IoU with different angular errors. It can be seen that the IoU will decrease as the aspect ratios of the objects increase when the angular errors remain unchanged. However, to achieve a good angular optimization for the predicted OBBs, we usually want that the decreasing of angular errors can directly reflect the changes of IoU, while the experiment results in Fig. 4 show that even when the angular error is decreased to a very small value in the optimization, it would not obtain a sufficiently large IoU for the objects with large aspect ratios. This indicates that the angular error would be sensitive to the aspect ratios in the training stage. To solve this problem, we derive a new representation of angular error that is consistent with the IoU by analyzing the mathematical relationship between the IoU, angular error, and aspect ratio, and then establish the adaptive angle optimization head (AAO-Head) using the newly defined angular-error parameter. Our code is publicly available at https://github.com/QYJ123/RPGAOD.

The main contributions of this article can be summarized as follows.

1) We establish a new transformation to directly generate rotation proposals from the horizontal anchors in one step and with fewer parameters, and propose the GRG-RPN that can use a small number of horizontal anchors and a simpler approach to generate high-quality rotation proposals for oriented object detection in remote sensing images.
2) We discover that the angular error would be sensitive to the aspect ratio of objects in the optimization process, and derive a new representation of angular error based on the mathematical relationship between



(a) (b) (c) (d)

Fig. 2. Comparison of different RPNs, which predict the rotation proposals directly from horizontal anchors. (a) Faster R-CNN-O. (b) Gliding vertex. (c) Oriented R-CNN. (d) GRG-RPN.

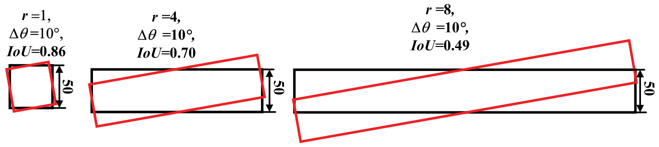


Fig. 3. IoU changes with different aspect ratios r under the same angular error $\Delta\theta$.

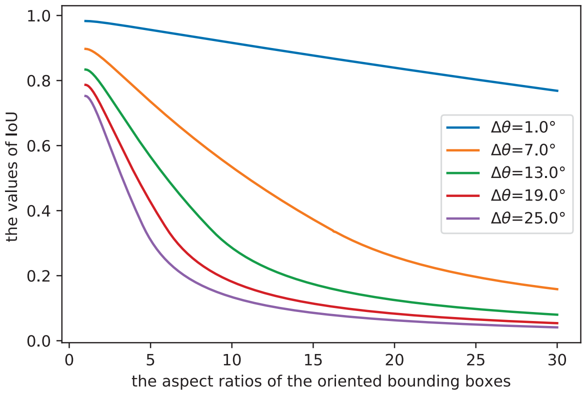


Fig. 4. Variation curve of IoU with different angular errors $\Delta\theta$.

the IoU, angular error, and aspect ratio. As a result, the AAO-Head is developed to improve the detection accuracy for oriented objects.

3) Extensive experiments demonstrate the superiority of our method over state-of-the-art (SOTA) competitors. Particularly, our method performs better in the accurate localization of oriented objects and can reduce false and missed detections in complex scenes.

II. Related Work

With the development of CNNs, object detection has improved dramatically in recent years [9], [11], [19], [20], [21]. Horizontal object detection is the beginning of the object detection task. With the object detection gradually being applied in remote sensing images, oriented object detection has received much attention in recent years.

A. Horizontal Object Detection

In the early stage, the object detection task uses the horizontal bounding boxes to localize the objects in the images. Therefore, this task of object detection is called horizontal object detection [4], [5], [12], [19]. The two-stage method based on the anchors is the initial method using the CNNs to achieve object detection. R-CNN [4] is the beginning of the CNN for horizontal object detection. It uses CNNs to obtain the feature maps of each selected region proposal. Then, it achieves classification and localization based on the feature maps. Fast R-CNN [5] and faster R-CNN [6] improve the R-CNN network, which can quickly obtain the feature maps and accurately select the region proposals. Thus, the detection speed and detection precision have been increasing. Mask R-CNN [7] achieves more accurate classification and localization by adding a mask branch. Furthermore, the single-stage method based on the anchors is another method that can obtain faster detection speed. For example, SSD [9] and YOLO [8] achieve classification and regression directly based on predefined anchors. Although the detection speed gets faster, the detection accuracy has decreased to a certain extent. To improve the performance of the single-stage method, YOLOv2 [10] and YOLOv3 [11] made great efforts. For example, these methods use multiscale anchors and batch normalization (BN) to achieve more accurate object detection. Focal loss [12] and GFocal loss [13] are also proposed to solve the imbalance of positive and negative samples and the boundary distribution of objects. Then, the loss can be designed considering the difficult classification samples, which can help achieve more accurate detection results. Feature pyramid network (FPN) [22] combines location information and semantic information of objects from different levels, which can perform a better detection by using these fusion feature maps. There are some other kinds of object detection methods based on transformer [19], [20], [21], [23] or anchor-free [24], [25], [26], [27] in recent years. These methods do not need to define anchors and can directly obtain the bounding boxes. But these methods usually need more time to learn the features of the objects, which leads to longer training time.

B. Oriented Object Detection

Due to the objects being arbitrary-oriented and densely packed in remote sensing images, some studies [2], [28], [29], [30], [31], [32], [33], [34], [35] have found that the OBBs can localize the objects more accurately. Then, oriented object detection is gradually developed d for more accurate detection. In the existing studies, most oriented object detectors can be divided into three types. First, the oriented object detectors directly regress the OBBs based on rotated anchors. The rotated RPN [16] achieves the OBB regression by defining

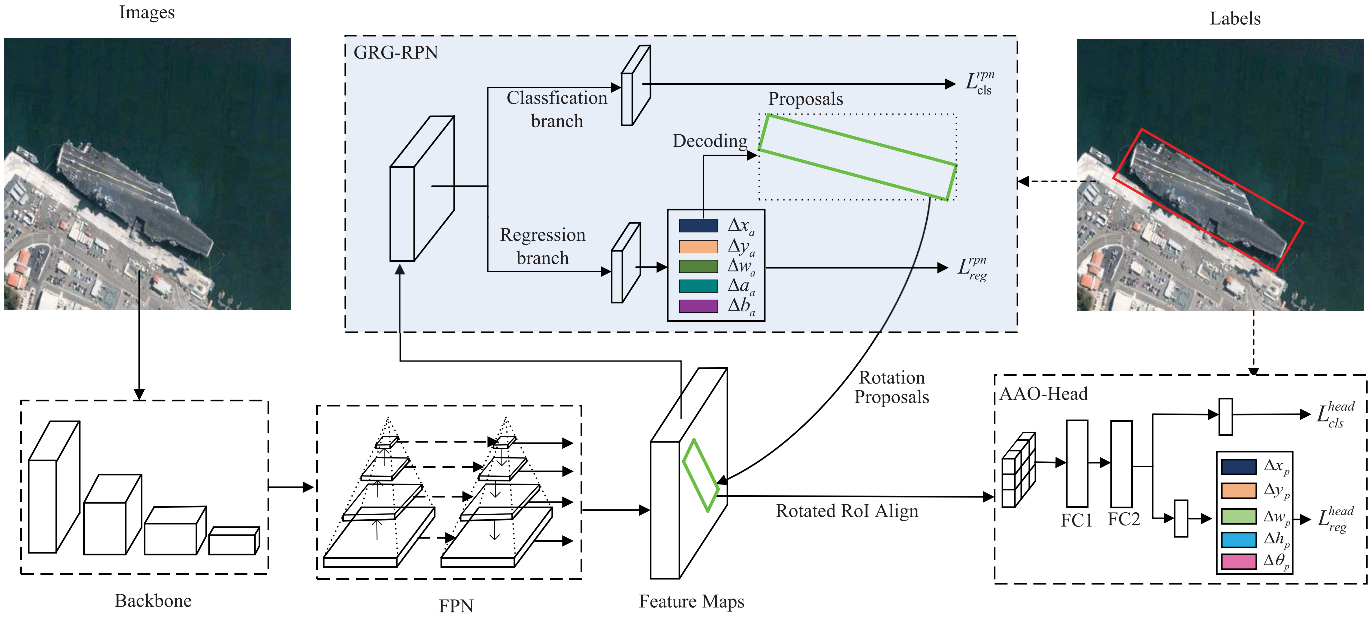


Fig. 5. Overall network architecture of the proposed method. Our method mainly contains four parts: backbone, FPN, GRG-RPN, and AAO-Head. The ground-truth labels are used as supervised information both for GRG-RPN and AAO-Head in the training process.

the rotated anchors with different widths, heights, and angles, which is the same as the horizontal object detection. But the massive rotated anchors will lead to the imbalance of the positive and negative samples and inferior object detection results. Second, the oriented object detectors obtain the OBBs by directly transforming the horizontal anchors. These methods mainly need angle alignment modules or feature alignment modules. EAST [36] and R2CNN [37] define new OBB forms to achieve the regression. It realizes the OBBs regression based on the horizontal anchors. RR-CNN [38] performs the rotating RoI pooling for aligning the oriented feature maps. Then, the rotating RoI pooling can help to achieve more precision localization and classification. The method based on Gaussian Wasserstein distance (GWD) [39] and Kullback-Leibler divergence (KLD) [40] regard the OBBs as 2-D Gaussian distributions and use different indicators to measure the similarity of the 2-D Gaussian distributions. The task-collaborated detector (TCD) [41] uses the task-collaborated assignment and task-collaborated head to improve the accuracy of the oriented detection. Finally, the oriented object detectors obtain the OBBs by using the EHBBs as auxiliary boxes. RoI transformer [17] proposes a new approach that realizes oriented object detection by using the transformation from horizontal anchors to OBBs. But it needs the RRoI learner network to align the oriented features from the horizontal feature maps. Gliding vertex [14] and oriented R-CNN [18] also use a transformation from horizontal anchors to OBBs. These methods usually need more parameters and are easy to generate localization errors. Besides, there also exists some anchor-free methods [1], [42], [43] to achieve oriented object detection, which usually needs more training time to obtain the final model parameters.

In this article, we propose a novel rotation proposal generation and optimization detector, termed RPGAOD, which mainly contains the GRG-RPN and the AAO-Head to achieve more accurate object detection in remote sensing images. The GRG-RPN directly transforms the horizontal anchors into rotation proposals using the least five parameters and a simple network. The AAO-Head considers the angular error sensitivity to the aspect ratios and realizes adaptive angle optimization based on newly designed regression parameters.

III. Proposed Method

The overall architecture of the proposed method is shown in Fig. 5. Our RPGAOD mainly contains two distinctive networks, the GRG-RPN and the AAO-Head. The GRG-RPN primarily uses a small number of horizontal anchors and a simple network to obtain high-quality rotation proposals. The rotation proposals are generated according to the geometric relationship between the OBBs and their EHBBs. The AAO-Head solves the angular error sensitivity to the aspect ratios based on the mathematical relationship between the angular errors, the IoU, and the aspect ratios in the angle optimization process. The primary difference between the GRG-RPN and the AAO-Head is the regression branch. The GRG-RPN regression directly obtains the high-quality rotation proposals from the horizontal anchors based on a new transformation, while the AAO-Head regression gets the final OBBs by applying a direct regression approach, due to the relatively-small angular errors between the rotation proposals of GRG-RPN and the ground-truth labels. In the following, we describe the two networks and other relevant parts of the proposed algorithm in detail.

A. Geometric Relationship Guided RPN

The GRG-RPN is a network for generating high-quality rotation proposals. It mainly includes the classification and regression branches. The classification branch estimates the probability that the anchors contain an object. The regression

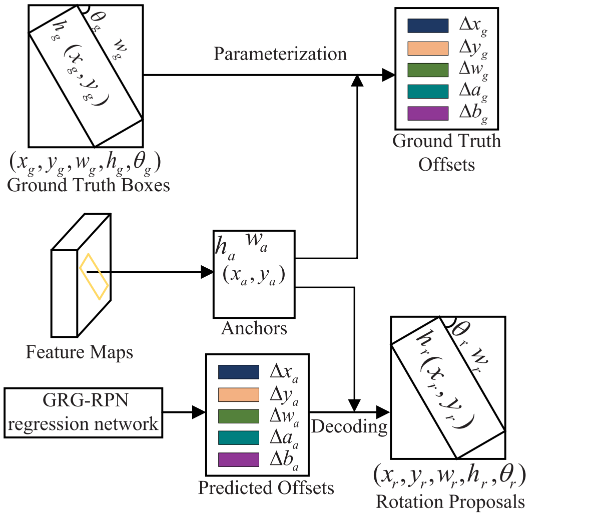


Fig. 6. Architecture of the GRG-RPN regression. The ground-truth offsets only are used in the training stage.

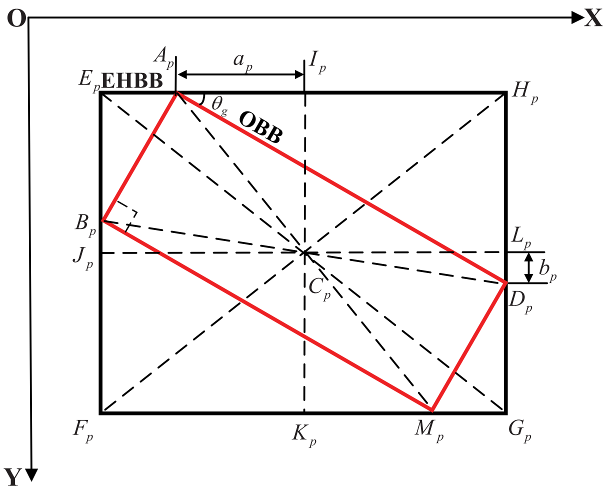


Fig. 7. Geometric relationship of a single OBB and its EHBB.

branch predicts the corresponding 5-D offsets of the region proposals relative to the anchors. In this article, we design a new set of offset parameters to express the differences between the ground-truth boxes and the anchors. As a result, the GRG-RPN can realize a direct transformation from the horizontal anchors to the rotation proposals using a small number of anchors and a simple network. This transformation is based on the geometric relationship between the OBBs and their EHBBs.

As shown in Fig. 6, the GRG-RPN contains two branches. One branch utilizes the ground-truth boxes and the horizontal anchors to achieve a new parameterization of the ground-truth offsets, which are used as supervisory information for training. The other calculates the rotation proposals by decoding the predicted offsets and the horizontal anchors for training and testing. With the predicted offsets gradually approximating the ground-truth offsets, the rotation proposals will converge progressively to the ground-truth boxes. In the following, we derive the transformation of the two branches.

The geometric relationship of a ground-truth OBB and its EHBB is shown in Fig. 7. $(x_g, y_g, w_g, h_g, \theta_g)$ and (x_h, y_h, w_h, h_h) represent the OBB and its EHBB, respectively, in which (x_g, y_g) denotes the center coordinates of the OBB, and w_g and y_g represents the width and height of the OBB, respectively (likewise for x_h, y_h, w_h, h_h). θ_g is the orientation of the OBB, which can be indicated by the angle between the EHBB and OBB. It can be seen that given the EHBB, the internal OBB can be uniquely determined by the locations of points A_p and D_p on the sides E_pH_p and H_pG_p of the EHBB. When the points A_p and D_p move on the corresponding lines, the orientation, width, and height of OBB would change continuously (maintaining that the OBB is always a rectangle). Therefore, we can uniquely define an OBB by its EHBB and the locations of its two vertexes on the EHBB. In our method, the locations of its two vertexes A_p and D_p are defined by their offsets relative to the midpoints I_p and L_p of the sides E_pH_p and H_pG_p (named as midpoint offsets), respectively. As a result, we introduce 5-D parameters $(\Delta x_g, \Delta y_g, \Delta w_g, \Delta a_g, \Delta b_g)$ to represent the offsets of the ground-truth OBB relative to the anchor, where $(\Delta x_g, \Delta y_g)$ denotes the center coordinate offsets of the EHBB relative to the anchor, Δw_g is the width offset of the EHBB relative to the anchor, and $(\Delta a_g, \Delta b_g)$ represents the midpoint offsets of the OBB relative to the EHBB. In summary, we can obtain the offsets of the OBB relative to the anchor by combining the offsets of the EHBB relative to the anchor and the offsets of the OBB relative to the EHBB with the above variables.

First, the EHBB and the OBB have an identical center coordinate $(x_g = x_h, y_g = y_h)$, which can be easily obtained from the location relationship between the OBB and its EHBB. Thus, the center coordinate offsets $(\Delta x_g, \Delta y_g)$ can be directly computed as follows:

$$\begin{cases} \Delta x_g = \dfrac{x_g - x_a}{w_a} \\ \Delta y_g = \dfrac{y_g - y_a}{h_a} \end{cases} \tag{1}$$

where (x_a, y_a) denotes the center coordinate of the anchor and (w_a, h_a) represents the width and height of the anchor. Then, Δw_g is defined as

$$\Delta w_g = \log\left(\frac{w_h}{w_a}\right). \tag{2}$$

As a result, the center coordinate of the OBB can be obtained by predicting the center coordinate offsets of the EHBB. The width of the EHBB w_h can be obtained by predicting Δw_g.

Second, when $-90° < \theta_g < 0°$ or $0° < \theta_g < 90°$, it is easy to obtain the following equation from the similar triangle relationship $(\Delta A_pE_pB_p \sim \Delta D_pH_pA_p)$ in Fig. 7:

$$\frac{|E_pA_p|}{|H_pD_p|} = \frac{|E_pB_p|}{|A_pH_p|} \tag{3}$$

where $|E_pA_p|$ is the distance between points E_p and A_p in Fig. 7 (likewise for $|H_pD_p|$, $|E_pB_p|$, $|A_pH_p|$). The midpoint offsets $(\Delta a_g, \Delta b_g)$ are defined as follows:

$$\Delta a_g = \frac{|E_pA_p| - |E_pI_p|}{|E_pH_p|}, \quad \Delta b_g = \frac{|H_pD_p| - |H_pL_p|}{|G_pH_p|} \tag{4}$$

when $0° < \theta_g < 90°$, considering the geometric relationship shown in Fig. 7, Δa_g and Δb_g can be more specifically calculated as follows:

$$\begin{cases} \Delta a_g = \dfrac{h_g \sin\theta_g - w_g \cos\theta_g}{2\left(h_g \sin\theta_g + w_g \cos\theta_g\right)} \\ \Delta b_g = \dfrac{w_g \sin\theta_g - h_g \cos\theta_g}{2\left(h_g \cos\theta_g + w_g \sin\theta_g\right)}. \end{cases} \tag{5}$$

Similarly, when $-90^\circ < \theta_g < 0^\circ$, Δa_g and Δb_g can be calculated as

$$\begin{cases} \Delta a_g = \dfrac{h_g \sin\theta_g + w_g \cos\theta_g}{2\left(w_g \cos\theta_g - h_g \sin\theta_g\right)} \\ \Delta b_g = \dfrac{w_g \sin\theta_g + h_g \cos\theta_g}{2\left(h_g \cos\theta_g - w_g \sin\theta_g\right)}. \end{cases} \tag{6}$$

In addition, the following relationships can be easily obtained from Fig. 7:

$$\begin{cases} |E_p I_p| = 0.5|E_p H_p| = 0.5 w_h \\ |H_p L_p| = 0.5|G_p H_p| = 0.5 h_h. \end{cases} \tag{7}$$

Then, we can obtain the following equations by combining (4) and (7):

$$\begin{aligned} &|E_p A_p| = \left(0.5 + \Delta a_g\right) w_h,\ |A_p H_p| = \left(0.5 - \Delta a_g\right) w_h \\ &|E_p A_p| = \left(0.5 + \Delta a_g\right) h_h,\ |A_p H_p| = \left(0.5 - \Delta a_g\right) h_h. \end{aligned} \tag{8}$$

Finally, we can obtain the following relation based on (3) and (8):

$$h_h = w_h \sqrt{\frac{1 - 4\Delta a_g^2}{1 - 4\Delta b_g^2}}. \tag{9}$$

It is apparent that h_h can be obtained using Δa_g, Δb_g, and w_h according to (9). Therefore, by predicting the offsets (Δx_g, Δy_g, Δw_g, Δa_g, Δb_g), the EHBB and the midpoint offsets can be obtained, and thus the OBB can be finally obtained as aforementioned.

Note that (9) is derived based on (3) by considering the similar triangle relationship ($\triangle A_p E_p B_p \sim \triangle D_p H_p A_p$) as shown in Fig. 7. From Fig. 7, it is obvious that maintaining the similar relationship $\triangle A_p E_p B_p \sim \triangle D_p H_p A_p$ can ensure that the angle $\angle B_p A_p D_p$ is a right angle, i.e., the OBB is a rectangle. Hence, using the computed h_h according to (9) (along with w_h, Δa_g, and Δb_g) can in turn ensure that the obtained OBB is a rectangle. This is why we can only use the three parameters (Δw_g, Δa_g, Δb_g) to directly determine the rectangle-shape of the OBB. The core of this process is the considering of the similar-triangle geometric relationship, and in this process, the EHBB can be regarded as an auxiliary box (or "a bridge") to compute the OBB. It is because of the used auxiliary box and relevant geometric relationship that we can define the new set of parameters enabling us to predict the OBBs directly based on the given horizontal anchors.

When θ_g is -90° or 0°, the OBB can be described only with four parameters (x_g, y_g, w_g, h_g). Since the two midpoint offsets are equal to -0.5, we only use Δb_g to indicate the midpoint offsets in this case. Besides, (9) can not be used to compute h_h anymore because the similar triangle relationship no longer exists. As a result, we use Δa_g to represent the height offset of the EHBB relative to the anchor. More specifically, Δa_g and Δb_g are defined as follows:

$$\Delta a_g = \log(h_h / h_a),\ \ \Delta b_g = -0.5. \tag{10}$$

In this case, by predicting (Δx_g, Δy_g, Δw_g, Δa_g), the EHBB can be obtained based on the anchor. Δb_g is -0.5, which indicates that the two midpoint offsets are both equal to -0.5. Then, the OBB can be finally determined using the EHBB and Δb_g. Consequently, the ground-truth offsets of the ground-truth boxes relative to the anchors can be represented by five parameters (Δx_g, Δy_g, Δw_g, Δa_g, Δb_g) regardless of whether the angles are in any of the above cases.

The calculation of the midpoint offsets (Δa_g, Δb_g) in different cases are summarized as follows:

$$\Delta a_g = \begin{cases} \dfrac{h_g \sin\theta_g + w_g \cos\theta_g}{2\left(w_g \cos\theta_g - h_g \sin\theta_g\right)}, & -90^\circ < \theta_g < 0^\circ \\ \dfrac{h_g \sin\theta_g - w_g \cos\theta_g}{2\left(h_g \sin\theta_g + w_g \cos\theta_g\right)}, & 0^\circ < \theta_g < 90^\circ \\ \log \dfrac{h_g \cos\theta_g - w_g \sin\theta_g}{h_a}. & \theta = -90^\circ \text{ or } 0^\circ \end{cases} \tag{11}$$

$$\Delta b_g = \begin{cases} \dfrac{w_g \sin\theta_g + h_g \cos\theta_g}{2\left(h_g \cos\theta_g - w_g \sin\theta_g\right)}, & -90^\circ < \theta_g < 0^\circ \\ \dfrac{w_g \sin\theta_g - h_g \cos\theta_g}{2\left(h_g \cos\theta_g + w_g \sin\theta_g\right)}, & 0^\circ < \theta_g < 90^\circ \\ -0.5. & \theta = -90^\circ \text{ or } 0^\circ. \end{cases} \tag{12}$$

In summary, we can define a new set of parameters (Δx_g, Δy_g, Δw_g, Δa_g, Δb_g) to represent the ground-truth offsets of the ground-truth box relative to the anchor, which are calculated by using. (1), (2), (11), and (12).

The decoding process is to calculate the predicted OBB [expressed by (x_r, y_r, w_r, h_r, θ_r)] using the predicted offset parameters (Δx_a, Δy_a, Δw_a, Δa_a, Δb_a) relative to the anchor. It is the inverse of the offset parameterization, which can be computed as follows:

$$x_r = \Delta x_a \cdot w_a + x_a,\ y_r = \Delta y_a \cdot h_a + y_a \tag{13}$$

$$w_h = w_a \cdot e^{\Delta w_a},\ h_h = \begin{cases} w_h \sqrt{\dfrac{1 - 4\Delta a_a^2}{1 - 4\Delta b_a^2}}, & |\Delta b_a| \neq 0.5 \\ h_a e^{\Delta a_a}, & |\Delta b_a| = 0.5 \end{cases} \tag{14}$$

$$w_r = \begin{cases} \sqrt{(0.5 - \Delta a_a)^2 w_h^2 + (0.5 + \Delta b_a)^2 h_h^2}, & |\Delta b_a| \neq 0.5 \\ w_h, & |\Delta b_a| = 0.5 \end{cases} \tag{15}$$

$$h_r = \begin{cases} \sqrt{(0.5 + \Delta a_a)^2 w_h^2 + (0.5 - \Delta b_a)^2 h_h^2} & |\Delta b_a| \neq 0.5 \\ h_h, & |\Delta b_a| = 0.5 \end{cases} \tag{16}$$

$$\theta_r = \begin{cases} \arctan \sqrt{\dfrac{(1 + 2\Delta a_a)(1 + 2\Delta b_a)}{(1 - 2\Delta a_a)(1 - 2\Delta b_a)}}, & |\Delta b_a| \neq 0.5 \\ -45(1 + 2\Delta b_a), & |\Delta b_a| = 0.5. \end{cases} \tag{17}$$

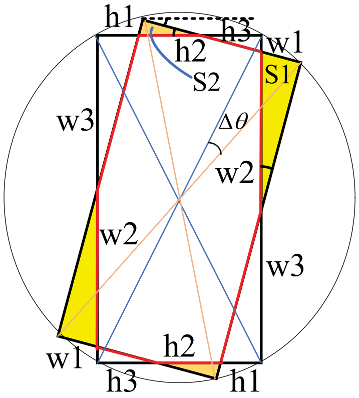


Fig. 8. Relationship between two OBBs with relatively-small angular error.

where w_h and h_h represent the width and height of the EHBB for the OBB to be predicted. In the training stage, when the errors between the predicted offset parameters and the ground-truth offset parameters become very small, we can obtain the OBBs that are very close to the ground-truth ones.

In the proposed GRG-RPN, the predicted OBBs are used to produce high-quality rotation proposals. With the newly-defined offset parameters and the corresponding decoding process, it enables us to generate high-quality rotation proposals only with a small number of horizontal anchors and a relatively simple network. These rotation proposals are then employed as the basis to obtain accurate detection results by the proposed AAO-Head.

B. Adaptive Angle Optimization Head

The primary function of the detection head is to achieve more accurate localization and classification of the objects in the classification and regression branches. Due to the relatively-small localization errors of the rotation proposals in the second stage, it is unnecessary to convert the OBBs into the EHBB and the midpoint offsets. As a result, the regression using typical offset parameters can be directly employed in the detection head. However, as explained in Section I, even when the angular error is decreased to a very small value in the angle optimization process, it would not obtain a sufficiently large IoU for the objects with large aspect ratios.

The above issue, in essence, reveals the inconsistency between the angular error and IoU. To reveal this point more clearly, we assume that the predicted OBB $(x_p, y_p, w_p, h_p, \theta_p)$ and the corresponding ground-truth box $(x_g, y_g, w_g, h_g, \theta_g)$ only have a relatively-small angular error. From Fig. 8, we can obtain that $h1$, $h2$, $h3$, $w1$, $w2$, and $w3$ satisfy the following equations:

$$\begin{cases} h_1 + h_2 + h_3 = h_g \\ w_1 + w_2 + w_3 = w_g \\ w_1 = h_3 \tan(\Delta\theta) \\ h_1 = w_3 \tan(\Delta\theta) \\ h_2 \cos(\Delta\theta) + \sqrt{w_1^2 + h_3^2} + w_2 \sin(\Delta\theta) = h_g \\ w_2 \cos(\Delta\theta) + \sqrt{h_1^2 + w_3^2} + h_2 \sin(\Delta\theta) = w_g. \end{cases} \tag{18}$$

Then, the IoU of the two boxes can be calculated as follows:

$$\begin{cases} S_1 = 0.5 \sin(\Delta\theta) \cdot \cos(\Delta\theta) \cdot w_2^2 \\ S_2 = 0.5 \sin(\Delta\theta) \cdot \cos(\Delta\theta) \cdot h_2^2 \\ r = \dfrac{w_g}{h_g} \\ \text{IoU} = \dfrac{h_g \cdot w_g - 2(S_1 + S_2)}{h_g \cdot w_g + 2(S_1 + S_2)} \end{cases} \tag{19}$$

where $\Delta\theta$ represents the angular error and r represents the aspect ratio of the OBB. As a result, the IoU can be calculated based on (18) and (19) as follows in (20), as shown at the bottom of the next page. The IoU for the Taylor expansion on the angular errors neglecting the influence of higher order terms can be finally obtained as follows:

$$\text{IoU} = 1 - \frac{(r + 1/r)\Delta\theta}{2}. \tag{21}$$

Equation (21) gives a clear relationship between the angular error $\Delta\theta$ and IoU. We can see that the variable of aspect ratio r would influence the constant relationship between $\Delta\theta$ and IoU and consequently make it hard to achieve a steady optimization on the angle of OBB during training. To solve this problem, in this article, we define a new angular-error parameter $\Delta\theta_t$ for the regression in the detection head based on the relationship given in (21):

$$\Delta\theta_t = 1 - \text{IoU} = 0.5(r + 1/r)\Delta\theta. \tag{22}$$

Using the new angular-error parameter, we could eliminate the influence of aspect ratio for training and effectively improve the detection accuracy (Section IV-D gives the ablation experiments to verify the improvement).

In summary, we design the AAO-Head with the new angular-error parameter, which can be used to calculate the ground-truth errors $(\Delta x_t, \Delta y_t, \Delta w_t, \Delta h_t, \Delta\theta_t)$ of the ground-truth boxes $(x_g, y_g, w_g, h_g, \theta_g)$ relative to the rotation proposals $(x_r, y_r, w_r, h_r, \theta_r)$ as follows:

$$\begin{cases} \Delta x_t = \dfrac{\cos\theta_r\left(x_g - x_r\right) + \sin\theta_r\left(y_g - y_r\right)}{w_r} \\ \Delta y_t = \dfrac{-\sin\theta_r\left(x_g - x_r\right) + \cos\theta_r\left(y_g - y_r\right)}{h_r} \\ \Delta w_t = \log\left(\dfrac{w_g}{w_r}\right) \\ \Delta h_t = \log\left(\dfrac{h_g}{h_r}\right) \\ \Delta\theta_t = 0.5\left(\theta_g - \theta_r\right)\left(\dfrac{w_g}{h_g} + \dfrac{h_g}{w_g}\right) \end{cases} \tag{23}$$

where $(\Delta x_t, \Delta y_t)$ denotes the errors of center coordinates, Δw_t and Δh_t are the errors of the width and height and $\Delta\theta_t$ contains the orientation errors of the ground-truth OBB relative to the rotation proposal and the aspect ratio of the ground-truth OBB.

Besides, we also need to compute the predicted bounding boxes $(x_p, y_p, w_p, h_p, \theta_p)$, which is the inverse process of generating the ground-truth errors. Specifically, the calculation of the predicted bounding boxes can be summarized as follows:

$$\begin{cases} x_p = \Delta x_p \cdot w_p \cdot \cos\theta_r - \Delta y_p \cdot h_p \cdot \sin\theta_r + x_r \\ y_p = \Delta x_p \cdot w_p \cdot \sin\theta_r + \Delta y_p \cdot h_p \cdot \cos\theta_r + y_r \\ w_p = w_r \cdot e^{\Delta w_p} \\ h_p = h_r \cdot e^{\Delta h_p} \\ \theta_p = \dfrac{2w_p \cdot h_p \cdot \Delta\theta_p}{w_p^2 + h_p^2} + \theta_r \end{cases} \tag{24}$$

where $(\Delta x_p, \Delta y_p, \Delta w_p, \Delta h_p, \Delta\theta_p)$ represents the predicted errors and the output of the AAO-Head. $(x_p, y_p, w_p, h_p, \theta_p)$ denotes the final predicted OBB. In the training stage, when the errors between the predicted errors and the ground-truth errors become very small, we can obtain the predicted OBBs that are very close to the ground-truth ones.

In the proposed AAO-Head, the predicted OBBs also are the outputs of our method for the final oriented object detection. With the newly-defined regression parameters, the AAO-Head can solve the sensitivity of the angular errors to the aspect ratios and achieve adaptive angle optimization by maintaining the constant relationship between $\Delta\theta$ and IoU.

C. Loss Function

As shown in Fig. 5, the whole loss of the RPGAOD contains two parts: the GRG-RPN loss and the AAO-Head loss. Each part includes the classification loss and regression loss. Hence, the whole loss L is defined as follows:

$$L = \lambda_1 L_{\mathrm{cls}}^{\mathrm{rpn}} + \lambda_2 L_{\mathrm{reg}}^{\mathrm{rpn}} + \lambda_3 L_{\mathrm{cls}}^{\mathrm{head}} + \lambda_4 L_{\mathrm{reg}}^{\mathrm{head}} \tag{25}$$

where $L_{\mathrm{cls}}^{\mathrm{rpn}}$ and $L_{\mathrm{reg}}^{\mathrm{rpn}}$ are the classification and regression losses of the GRG-RPN, respectively. $L_{\mathrm{cls}}^{\mathrm{rpn}}$ and $L_{\mathrm{reg}}^{\mathrm{rpn}}$ are the classification and regression losses of the AAO-Head, respectively. Each loss function is defined as follows:

$$\begin{aligned} L_{\mathrm{cls}}^{\mathrm{rpn}} &= \frac{1}{N_1}\sum_{i=1}^{N_1} L_{\mathrm{cls}}\left(p_i, p_i^*\right) \\ L_{\mathrm{reg}}^{\mathrm{rpn}} &= \frac{1}{N_1} p_i^* \sum_{i=1}^{N_1} L_{\mathrm{reg}}\left(\boldsymbol{t_i}, \boldsymbol{t_i^*}\right) \\ L_{\mathrm{cls}}^{\mathrm{head}} &= \frac{1}{N_2}\sum_{j=1}^{N_2} L_{\mathrm{cls}}\left(q_j, q_j^*\right) \\ L_{\mathrm{reg}}^{\mathrm{head}} &= \frac{1}{N_2} p_i^* \sum_{j=1}^{N_2} L_{\mathrm{reg}}\left(\boldsymbol{s_j}, \boldsymbol{s_j^*}\right) \end{aligned} \tag{26}$$

where i is the index of the anchors and N_1 (by default $N_1 = 256$) is the total number of sampling anchors in the GRG-RPN. p_i and p_i^* represent the predicted probability of GRG-RPN classification and the ground-truth labels, respectively. $\boldsymbol{t_i}$ $(\Delta x_a, \Delta y_a, \Delta w_a, \Delta a_a, \Delta b_a)$ and $\boldsymbol{t_i^*}$ $(\Delta x_g, \Delta y_g, \Delta w_g, \Delta a_g, \Delta b_g)$ are the predicted offsets of the GRG-RPN regression outputs and the ground-truth offsets of the ground-truth boxes relative to the horizontal anchors, respectively. Similarly, j is the index of proposals and N_2 (by default $N_2 = 512$) is the total number of sampling rotation proposals in the AAO-Head. q_j and q_j^* represent the predicted probability of AAO-Head classification and the ground-truth labels, respectively. $\boldsymbol{s_i}$ $(\Delta x_p, \Delta y_p, \Delta w_p, \Delta h_p, \Delta\theta_p)$ and $\boldsymbol{s_i^*}$ $(\Delta x_t, \Delta y_t, \Delta w_t, \Delta h_t, \Delta\theta_t)$ represent the predicted offsets of the AAO-Head regression outputs and the ground-truth offsets of the ground-truth bounding boxes relative to the rotation proposals, respectively. $L_{\mathrm{cls}}^{\mathrm{rpn}}$ and $L_{\mathrm{cls}}^{\mathrm{head}}$ use cross-entropy loss, while $L_{\mathrm{reg}}^{\mathrm{rpn}}$ and $L_{\mathrm{reg}}^{\mathrm{head}}$ use smooth L1 loss.

In the whole classification loss, $L_{\mathrm{cls}}^{\mathrm{rpn}}$ is only used to distinguish the foreground and background. In other words, $L_{\mathrm{cls}}^{\mathrm{rpn}}$ is used to indicate whether the objects have an intersection with the anchors, which can be regarded as a binary classification. $L_{\mathrm{cls}}^{\mathrm{head}}$ is used to distinguish the specific categories of the objects in the rotation proposals, which is often a multiclassification problem. The difficulty of the classification problem will generally increase with the increase of the object categories. Therefore, appropriately increasing the weight of $L_{\mathrm{cls}}^{\mathrm{head}}$ is contributive to obtain more accurate oriented object detection. In our method, it allows us to emphasize more on the reduction of $L_{\mathrm{cls}}^{\mathrm{head}}$ with higher weight in the training loss, due to the relatively simple and more accurate regression we proposed, which would help us to improve the detection accuracy. In Section IV-D, we will give the ablation experiments to verify the superiority of weighted loss against the commonly-used average loss.

IV. Experiments

In this section, we first introduce the datasets and implementation details and then demonstrate the superiority of our method compared with other SOTA methods. Finally, we verify the effectiveness of different parts of our method in the ablation studies.

A. Datasets

In the following experiments, the proposed method is evaluated on two publicly available datasets: the HRSC2016 dataset[1] and the DOTA dataset.[2]

The HRSC2016 dataset is a classic single-class detection dataset in the remote sensing field. On the whole, the HRSC2016 dataset contains 1061 images with sizes ranging from 300 × 300 to 1500 × 900. All images in the dataset are divided into three parts: the training set (436 images), the validation set (181 images), and the testing set (444 images). In the experiments, we use the training and validation sets for training and the testing set for testing.

$$\mathrm{IoU} = \begin{cases} \dfrac{\sin(2\Delta\theta) - (r + 1/r - 2\sin(\Delta\theta)) \cdot (1 - \cos(\Delta\theta))}{\sin(2\Delta\theta) + (r + 1/r - 2\sin(\Delta\theta)) \cdot (1 - \cos(\Delta\theta))} & \Delta\theta \neq 0 \\ 1 & \Delta\theta = 0 \end{cases} \tag{20}$$

[1] https://sites.google.com/site/hrsc2016/

[2] https://captain-whu.github.io/DOTA/index.html

Besides, the DOTA dataset is a large-scale remote sensing dataset. The image sizes of the DOTA dataset range from 800 × 800 to 4000 × 4000 pixels. There are 15 object classes in the dataset: plane (PL), baseball diamond (BD), bridge (BR), ground track field (GTF), small vehicle (SV), large vehicle (LV), ship (SH), tennis court (TC), basketball court (BC), storage tank (ST), soccer-ball field (SBF), roundabout (RA), harbor (HA), swimming pool (SP), and helicopter (HC). The DOTA dataset contains 2806 images and 188 282 instances with four vertex coordinates annotations, which are transformed as rectangles for training. In the experiments, all images of the dataset are divided into two parts. The training images contain the training and validation sets. The testing images only contain the testing set. The mAP can be obtained by submitting the testing results to the DOTA's evaluation server.

B. Implementation Details

The computer used in the experiments is configured with the Intel i7-10700K CPU 32G (8G × 4) RAM and two NVIDIA RTX 2080Ti 11G GPUs. The whole experiments were conducted on the Ubuntu 18.04 operating system and Pytorch 1.7.1 software framework, and the model training process was accelerated via the cuda 10.1. All codes are developed based on the mmdetection platform [44] with python 3.7. In the training, two RTX 2080Ti GPUs are used with a batch size of 2. The inference process is only tested on a single RTX 2080Ti GPU. The ResNet50 and ResNet101 with pretrained parameters on ImageNet are used in the training, which makes the network converge quickly. The overall network is optimized with the simple stochastic gradient descent (SGD) algorithm, in which the momentum and the weight decay are set to 0.9 and 0.0001, respectively. In addition, horizontal and vertical flipping are adopted for data augmentation in the training stage.

For the DOTA dataset, we crop the original images into 1024 × 1024 patches due to the large scale of the image. The stride of cropping is set to 824, and the pixel overlap between two adjacent patches is 200. For multiscale training and testing, we first resize the original images into three scales (0.5, 1.0, and 1.5) and crop them into 1024 × 1024 patches with the stride 524. We train our method with 12 epochs. The initial learning rate is set to 0.01 and divided by 10 at epochs 8 and 11. The non-maximum suppression (NMS) threshold is set to 0.1 for obtaining the final bounding boxes.

For the HRSC2016 dataset, the aspect ratios of images need to be maintained. The shorter sides of images are resized to 800, while the longer sides are less than or equal to 1333. The initial learning rate is set to 0.01 and divided by 10 at epochs 24 and 33. The total epoch is set to 36 in the training. Besides, we use the mAP computed with two different metrics to compare the performance of different algorithms. They are PASCAL VOC2007 and VOC2012 [45], [46], [47], [48], respectively. The primary difference between the two metrics is the way to calculate AP for each category. The PASCAL VOC07 metric uses 11-point interpolation to calculate AP for each category, while the PASCAL VOC12 metric adopts all-point interpolation for the calculation.

C. Comparison With State-of-the-Arts

We compare our method with the other 22 methods in this section. These methods can be divided into two categories. The first category is one-stage techniques including RetinaNet-O [12], DRN [49], R3Det [50], PIoU [51], RSDet [52], DAL [53], KLD [40], P2P [54], SASM [55], and S^2ANet [56]. The second is two-stage techniques including R2CNN [37], rotated RPN [16], RoI transformer [17], gliding vertex [14], ReDet [57], mask OBB [58], oriented RepPoints [1], AOPG [43], oriented R-CNN [18], and DODet [15]. In these methods, some methods propose the new bounding box loss to achieve oriented object detection, such as KLD, P2P, and PIoU. Some other methods mainly utilize the new representation of the OBB to obtain the final bounding boxes, for example, gliding vertex, oriented R-CNN, and AOPG, while the other methods use some improved strategies (e.g., label assignment, feature selection, and network structure) to optimize the network and obtain more accurate object detection results, such as the DAL, SASM, and DRN.

1) *Results on the HRSC2016 Dataset:* We compare our method on the HRSC2016 dataset with other 22 advanced oriented object detectors in Table I. As shown in Table I, our method achieves 90.40% and 96.49% mAP under PASCAL VOC 07 and PASCAL VOC 12 metrics based on the ResNet50 with FPN, respectively. In addition, our method can achieve 90.57% mAP under PASCAL VOC 07 metric and 98.06% mAP under PASCAL VOC 12 metric based on the ResNet101 with FPN, which are very competitive results compared with the current SOTA methods. Compared with the rotated RPN and RoI transformer, our method has an increase of 12.42% mAP under PASCAL VOC 12 metric and 4.37% mAP under PASCAL VOC 07 metric, respectively. Besides, with multiscale training and testing, our method can obtain 90.44% and 90.58% mAP under PASCAL VOC 07 metric and achieve 97.22% and 98.07% mAP under PASCAL VOC 12 metric, respectively. Some detection results on the HRSC2016 dataset are shown in Fig. 9.

Besides, Fig. 10 shows some detection results for the same samples on the HRSC2016 dataset using different methods. Compared with the faster R-CNN-O [6], gliding vertex [14], S^2ANet [56], and oriented R-CNN [18], our method not only can decrease the false detections (the blue dashed ellipses in rows 2 and 4) and the missed detections (the red dashed ellipses in rows 1, 2, 3, and 5), but also performs better in accurate localization of oriented objects (the yellow dashed ellipses in rows 2–5).

2) *Results on the DOTA Dataset:* We compare the proposed method on the DOTA dataset with other SOTA methods in Table II. Without any special tricks, our method achieves 76.47% mAP with single-scale training and testing. Moreover, the proposed method can obtain 81.20% mAP using multiscale training and testing, which are competitive compared with other methods. Compared with the rotated RPN and RoI transformer, our method has an increase of 16.46% and 6.91% mAP with single-scale training and testing, respectively. The comparison results in Table II show that our method can obtain higher mAP compared with

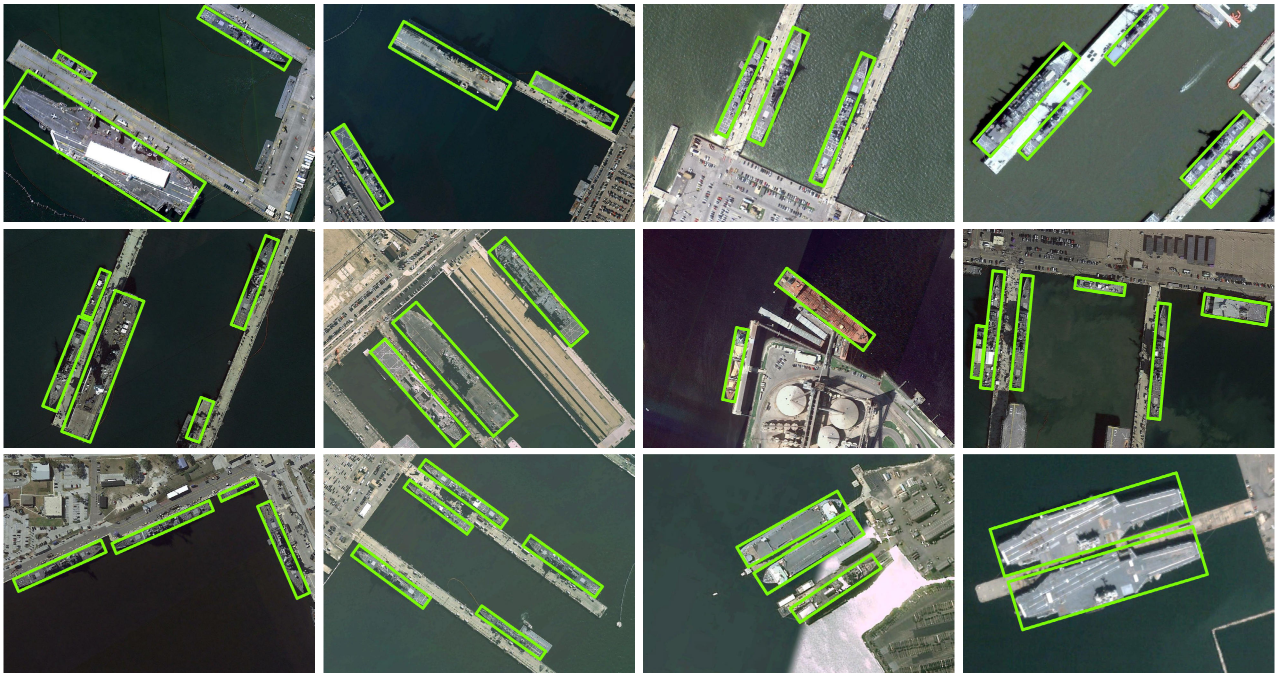


Fig. 9. Visualization of some detection results on the HRSC2016 dataset. Finally, OBBs need NMS with IoU greater than 0.1 and scores greater than 0.3.

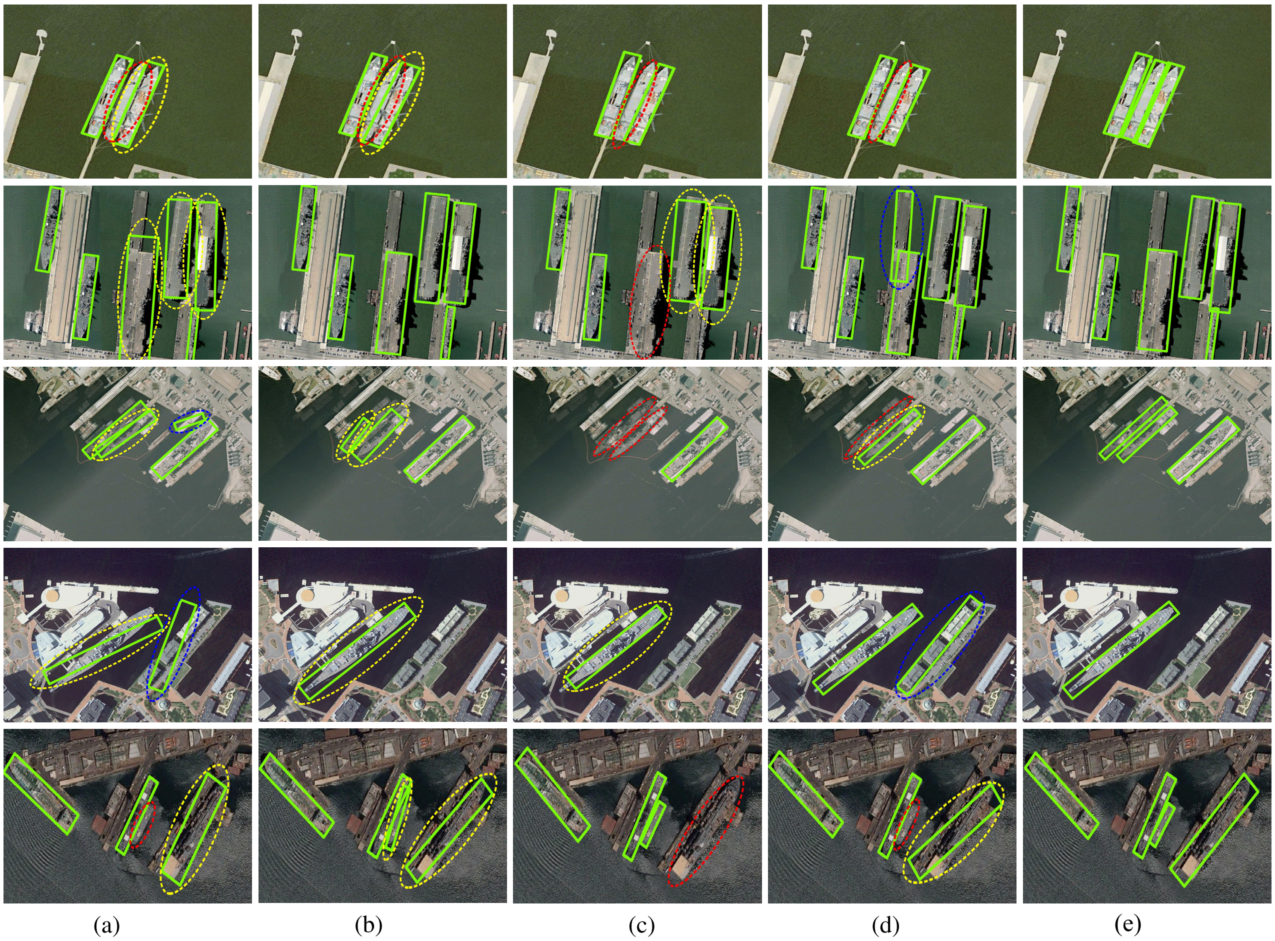


Fig. 10. Detection results of some samples on the HRSC2016 dataset using different methods. The yellow dashed ellipses denote the detection results with positioning errors. The red dashed ellipses and the blue dashed ellipses mark the missed and the false detections, respectively. (a) Faster R-CNN-O. (b) Gliding vertex. (c) S^2ANet. (d) Oriented R-CNN. (e) Oriented R-CNN.

other SOTA methods. Some visualization results on the DOTA dataset are presented in Fig. 11. In addition, we also show some detection results for the same samples on the DOTA dataset using different methods in Fig. 12. Compared with faster R-CNN-O, gliding vertex, S^2ANet, and oriented R-CNN, our method can achieve more accurate detection

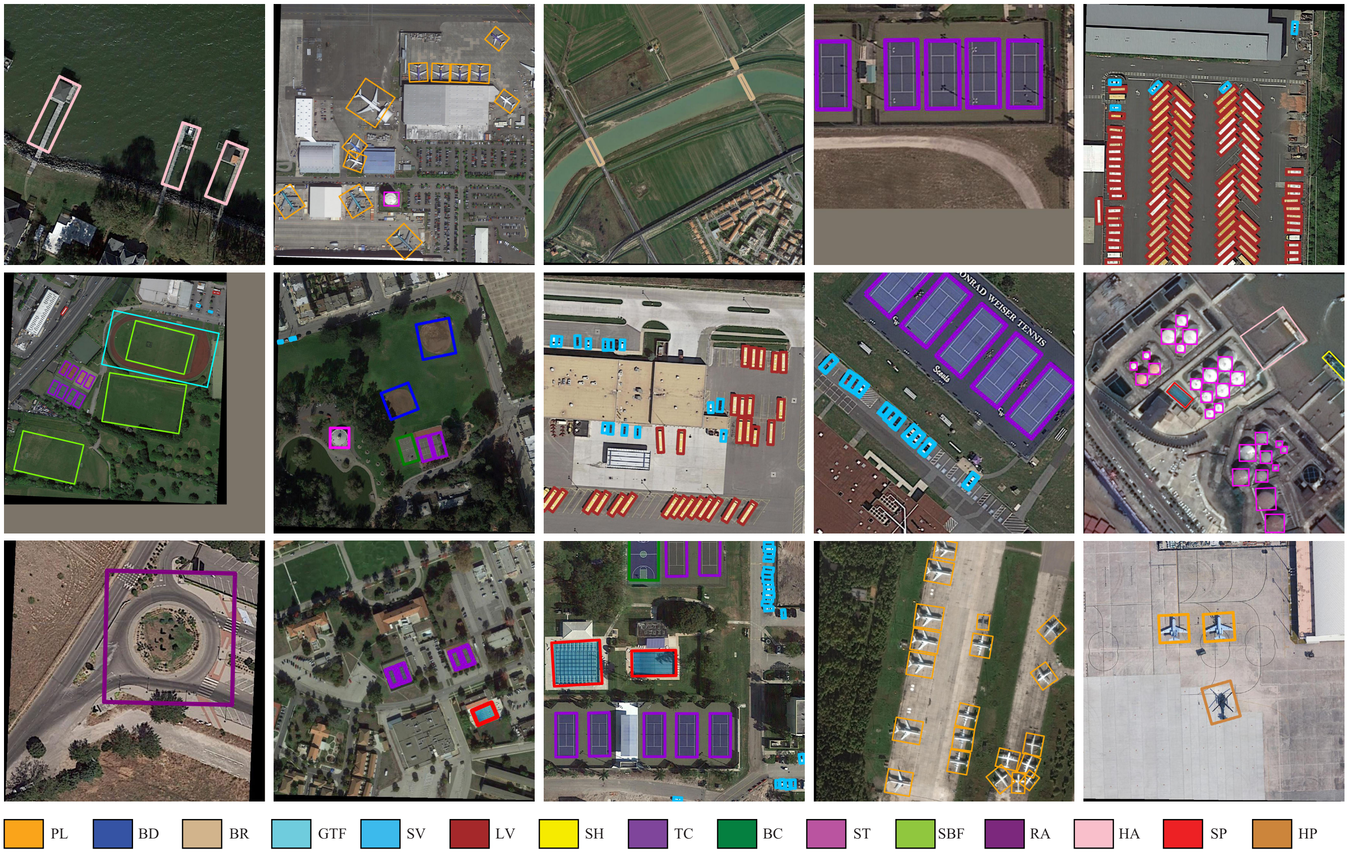


Fig. 11. Visualization of some detection results on the DOTA dataset. Finally, OBBs need NMS with IoU greater than 0.1 and scores greater than 0.05.

TABLE I

COMPARISON WITH SOME ADVANCED METHODS ON THE HRSC2016 DATASET. ‡ MEANS MULTISCALE TRAINING AND TESTING. THE QUANTITATIVE RESULTS WERE TAKEN FROM THE CITED REFERENCES IN THE FIRST COLUMN

Methods	Backbone	mAP(07)	mAP(12)
RetinaNet-O [12]	R-50-FPN	84.28	-
DRN [49]	H-34	-	92.70
R3Det [50]	R-101-FPN	89.26	96.01
PIoU [51]	DLA-34	89.20	-
RSDet [52]	R-152-FPN	86.50	-
DAL [53]	R-101-FPN	89.77	-
KLD [40]	R-50-FPN	87.45	-
P2P [54]	R-50-FPN	89.19	-
SASM [55]	R-101-FPN	90.27	-
S^{2}ANet [56]	R-101-FPN	90.17	95.01
Rotated RPN [16]	R-101-FPN	79.08	85.64
R2CNN [37]	R-101-FPN	73.07	79.73
Faster R-CNN-O [6]	R-50-FPN	83.93	-
Centermap-Net [59]	R-50-FPN	-	92.80
Mask OBB [58]	R-50-FPN	-	96.70
Gliding vertex [14]	R-101-FPN	88.20	-
RoI Transformer [17]	R-101-FPN	86.20	-
AOPG [43]	R-50-FPN	90.34	96.22
DODet [15]	R-101-FPN	90.89	97.14
ReDet [57]	ReR50-FPN	90.46	97.63
Oriented RepPoints [1]	R-50-FPN	90.38	97.26
Oriented R-CNN [18]	R-101-FPN	90.50	97.60
Ours			
RPGAOD	R-50-FPN	90.40	96.49
RPGAOD	R-101-FPN	90.57	98.06
RPGAOD‡	R-50-FPN	90.44	97.22
RPGAOD ‡	R-101-FPN	90.58	98.07

results, which can be seen from the detections for the densely packed objects (rows 2–5) and slightly blurry objects (rows 1 and 6).

Besides, we also show some samples that our method can not perform well in Fig. 13. These situations are still challenging due to the similar objects compared with the ships and the extremely small objects (small ships or vehicles) which are very close to the large objects.

D. Ablation Studies

1) *Different Settings of Our Method:* We conduct ablation experiments on the DOTA dataset to verify the superiority of our method. As shown in Table III, our method based on the faster R-CNN-O (baseline method) mainly contains three modules: the GRG-RPN, the AAO-Head, and the weighted loss. Compared with the baseline method, the GRG-RPN and the AAO-Head can obtain 5.56% and 5.16% mAP improvement, respectively. In addition, our method can achieve 76.47% with single scale training and testing after the weighted loss was applied, which verifies the effectiveness of our method.

2) *Effectiveness of the GRG-RPN:* As shown in Table IV, we compare different RPN approaches to verify the superiority of GRG-RPN. Compared with the RPN approaches in faster R-CNN-O and RoI transformer, GRG-RPN can achieve 75.89% mAP, increasing 6.84% and 1.28% mAP on the DOTA dataset, respectively. In addition, the GRG-RPN's detection speed also can achieve 16.0 FPS increasing 2.8 and 1.2 FPS based on 1024 × 1024 images compared with the RPN approaches in RoI transformer and gliding vertex, while the detection speed of GRG-RPN is almost the same as that of the RPN approach in faster R-CNN-O. In terms of GFLOPs and the number of model parameters (denoted as "Params" in

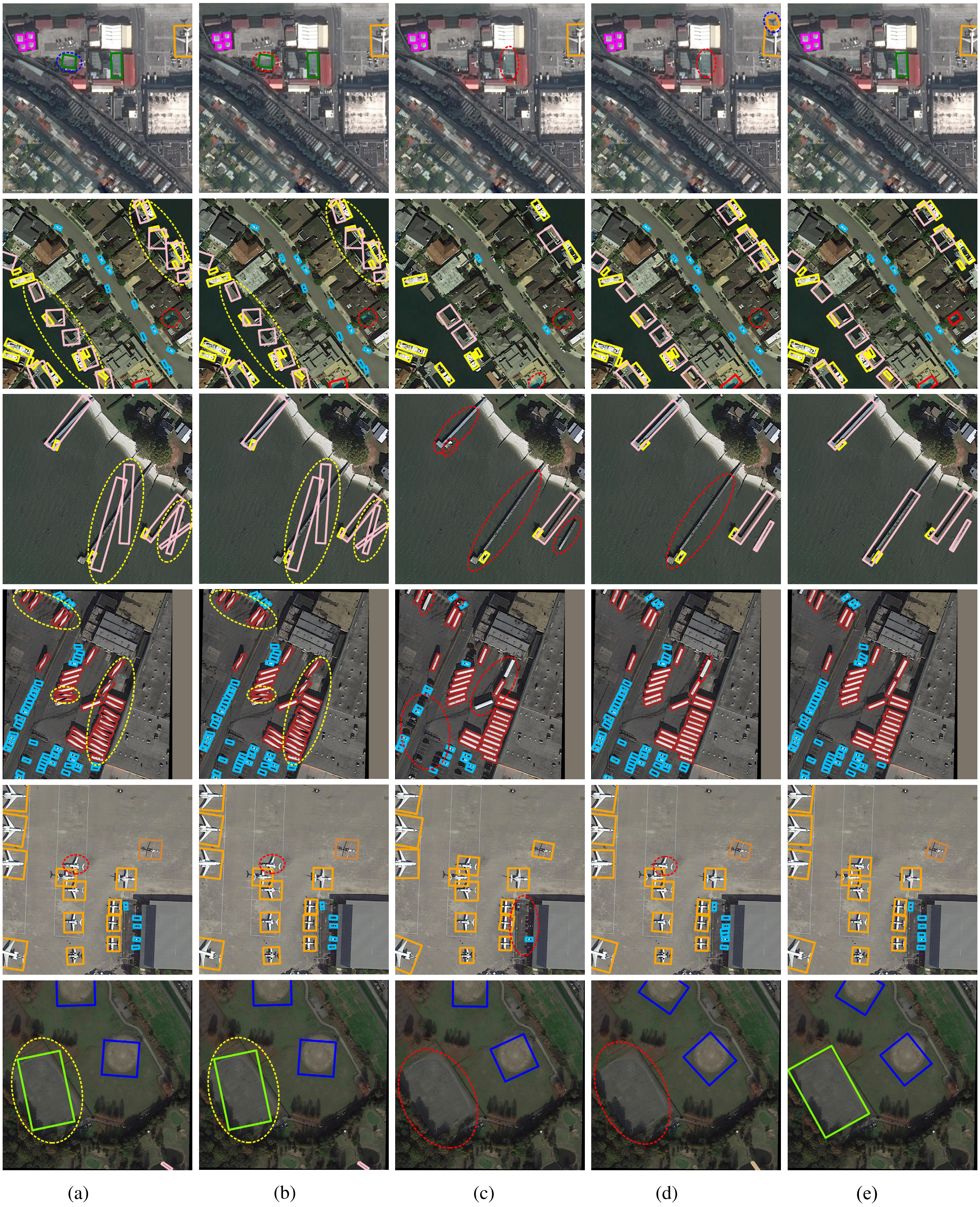


Fig. 12. Detection results of some samples on the DOTA dataset using different methods. The yellow dashed ellipses denote the detection results with positioning errors. The red dashed ellipses and the blue dashed ellipses mark the missed and the false detections, respectively. The different color boxes are the detection results of different categories, which are similar to Fig. 11. (a) Faster R-CNN-O. (b) Gliding vertex. (c) S^2ANet. (d) Oriented R-CNN. (e) Oriented R-CNN.

Table IV), GRG-RPN has no significant change compared with the RPN approach in faster R-CNN-O and reduces GFLOPs and the number of model parameters compared with RPN approaches in RoI transformer and gliding vertex. In summary, GRG-RPN can obtain high-quality rotation proposals and achieve higher mAP compared with other RPN approaches.

3) *Effectiveness of the AAO-Head:* We use different representations of angular error to verify the superiority of the

TABLE II

COMPARISON WITH SOME ADVANCED METHODS ON THE DOTA DATASET. ‡ MEANS MULTISCALE TRAINING AND TESTING. THE QUANTITATIVE RESULTS WERE TAKEN FROM THE CITED REFERENCES IN THE "METHODS" COLUMN

Methods	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
RetinaNet-O [12]	R-50-FPN	88.67	77.62	41.81	58.17	74.58	71.64	79.11	90.29	82.18	74.32	54.75	60.60	62.57	69.67	60.64	68.43
DRN [49]	H-104	88.91	80.22	43.52	63.35	73.48	70.69	84.94	90.14	83.85	84.11	50.12	58.41	67.62	68.60	52.50	70.70
R3Det [50]	R-101-FPN	88.76	83.09	50.91	67.27	76.23	80.39	86.72	90.78	84.68	83.24	61.98	61.35	66.91	70.63	53.94	73.79
PIoU [51]	DLA-34	80.90	69.70	24.10	60.20	38.30	64.40	64.80	90.90	77.20	70.40	46.50	37.10	57.10	61.90	64.00	60.50
RSDet [52]	R-101-FPN	89.80	82.90	48.60	65.20	69.50	70.10	70.20	90.50	85.60	83.40	62.50	63.90	65.60	67.20	68.00	72.20
DAL [53]	R-50-FPN	88.68	76.55	45.08	66.80	67.00	76.76	79.74	90.84	79.54	78.45	57.71	62.27	69.05	73.14	60.11	71.44
KLD [40]	R-50-FPN	88.91	83.71	50.10	68.75	78.20	76.05	84.58	89.41	86.15	85.28	63.15	60.90	75.06	71.51	67.45	75.28
P2P [54]	R-50-FPN	88.96	76.97	48.47	69.85	79.43	80.14	88.14	90.85	84.98	85.46	59.89	62.91	71.88	66.18	56.41	74.04
SASM [55]	R-50-FPN	86.42	78.97	52.47	69.84	77.3	75.99	86.72	90.89	82.63	85.66	60.13	68.25	73.98	72.22	62.37	74.92
S^{2}ANet [56]	R-50-FPN	89.11	82.84	48.37	71.11	78.11	78.39	87.25	90.83	84.90	85.64	60.36	62.60	65.26	69.13	57.94	74.12
Rotated RPN [16]	R-101-FPN	80.94	65.75	35.34	67.44	59.92	50.91	55.81	90.67	66.92	72.39	55.06	52.23	55.14	53.35	48.22	60.01
R2CNN [37]	R-101-FPN	80.94	65.67	35.34	67.44	59.92	50.91	55.81	90.67	66.92	72.39	55.06	52.23	55.14	53.35	48.22	60.67
Faster R-CNN-O [6]	R-50-FPN	88.44	73.06	44.86	59.09	73.25	71.49	77.11	90.84	78.94	83.90	48.59	62.95	62.18	64.91	56.18	69.05
Centermap-Net [59]	R-50-FPN	88.88	81.24	53.15	60.65	78.62	66.55	78.10	88.83	77.80	83.61	49.36	66.19	72.10	72.36	58.70	71.74
Mask OBB [58]	R-50-FPN	89.61	85.09	51.85	72.90	75.28	73.23	85.57	90.37	82.08	85.05	55.73	68.39	71.61	69.87	66.33	74.86
Gliding vertex [14]	R-101-FPN	89.64	**85.00**	52.26	77.34	73.01	73.14	86.82	90.74	79.02	86.81	59.55	70.91	72.94	70.86	57.32	75.02
RoI Transformer [17]	R-101-FPN	88.64	78.52	43.44	75.92	68.81	73.68	83.59	90.74	77.27	81.46	58.39	53.54	62.83	58.93	47.67	69.56
AOPG [43]	R-101-FPN	89.14	82.74	51.87	69.28	77.65	82.42	88.08	90.89	86.26	85.13	60.60	66.30	74.05	67.76	58.77	75.39
DODet [15]	R-101-FPN	89.61	83.10	51.43	72.02	79.16	81.99	87.71	90.89	86.53	84.56	62.21	65.38	71.98	70.79	61.93	75.89
ReDet [57]	ReR50-FPN	88.79	82.64	53.97	74.00	78.13	84.06	88.04	90.89	**87.78**	85.75	61.76	60.39	75.96	68.07	63.59	76.25
Oriented RepPoints [1]	R-50-FPN	87.02	83.17	54.13	71.16	**80.18**	78.40	87.28	**90.90**	85.97	86.25	59.90	70.49	73.53	72.27	58.97	75.97
Oriented R-CNN [18]	R-50-FPN	89.46	82.12	54.78	70.86	78.93	83.00	88.20	**90.90**	87.50	84.68	63.97	67.69	74.97	68.84	52.28	75.87
Ours																	
RPGAOD	R-50-FPN	89.34	83.53	54.39	76.59	78.09	81.44	87.64	90.83	85.89	85.33	65.44	64.96	73.73	70.31	59.50	76.47
RPGAOD	R-101-FPN	89.40	83.87	54.96	75.78	77.88	82.03	87.62	90.90	86.49	85.42	64.82	63.11	75.79	69.22	57.45	76.31
RPGAOD‡	R-50-FPN	89.93	85.00	60.16	**81.58**	79.63	**84.94**	**88.71**	**90.90**	86.24	87.65	**72.94**	**72.61**	81.24	**79.78**	**76.76**	**81.20**
RPGAOD‡	R-101-FPN	**90.20**	**85.42**	**61.21**	80.59	79.48	**84.94**	88.39	90.88	86.95	**87.79**	69.62	71.63	**82.82**	77.96	71.95	80.65

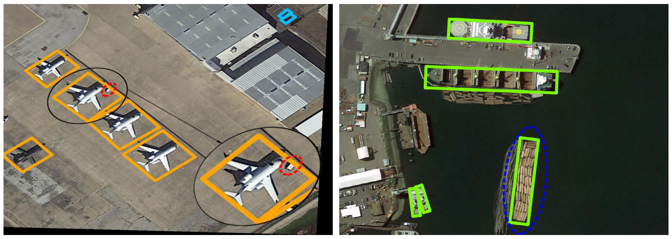


Fig. 13. Some samples that our method can not perform well. The red dashed ellipses and the blue dashed ellipses mark the missed and the false detections, respectively. (a) DOTA sample. (b) HRSC2016 sample.

TABLE III

DIFFERENT SETTINGS OF OUR METHOD ON THE DOTA DATASET. THE SYMBOL "✓" REPRESENTS USING THE MODULE. ON THE CONTRARY, THE SYMBOL "✕" REPRESENTS NOT USING THE MODULE

Module	Baseline	Different Settings of our method				
GRG-RPN	✕	✓	✕	✓	✓	✓
AAO-Head	✕	✕	✓	✓	✕	✓
weighted loss	✕	✕	✕	✕	✓	✓
mAP/(%)	69.05	74.61	74.21	75.43	75.89	76.47

TABLE IV

COMPARISON OF DIFFERENT RPNS. ALL METHODS USE THE GENERAL DETECTION HEAD DIRECTLY REGRESSING THE ANGULAR ERRORS

RPN approach	mAP/(%)	FPS	GFLOPs	Params
Faster R-CNN-O	69.05	16.5	211.30	41.14M
RoI Transformer	74.61	13.2	225.49	55.13M
Gliding Vertex	71.84	14.8	211.43	41.27M
GRG-RPN	75.89	16.0	211.36	41.14M

AAO-Head in Table V. In Table V, we compare four different angular error representations. The first row is the general angular error representation usually used in the detection head, and the second row is the proposed angular error representation in AAO-Head. The rest are two other angular error representations in the detection head. From Table V, it can be seen that the mAP can be improved using different angular error representations combining the aspect ratios and angular errors.

TABLE V

COMPARISON OF DIFFERENT DETECTION HEAD. $\Delta\theta_t$ DENOTES THE ANGLE REGRESSION VARIABLES OF THE AAO-HEAD, AND $\Delta\theta$ REPRESENTS THE ANGLE ERRORS BETWEEN THE ROTATION PROPOSALS AND GROUND-TRUTH BOUNDING BOXES. ALL DETECTION HEADS USE THE GRG-RPN TO GENERATE THE ROTATION PROPOSALS

Angular error representation	mAP/(%)
$\Delta\theta_t = \Delta\theta$	75.89
$\Delta\theta_t = (r + 1/r)^2 * \Delta\theta$	75.93
$\Delta\theta_t = \sqrt{r + 1/r} * \Delta\theta$	76.12
$\Delta\theta_t = (r^2 + 1)/(2r) * \Delta\theta$	76.47

TABLE VI

COMPARISON OF DIFFERENT WEIGHTED LOSS. ALL EXPERIMENTS USE THE GRG-RPN TO GENERATE ROTATION PROPOSALS AND THE AAO-HEAD TO OPTIMIZE THE ROTATION PROPOSALS. [λ_1, λ_2, λ_3, λ_4] IS THE WEIGHTING PARAMETERS OF THE LOSS L

[λ_1,λ_2,λ_3,λ_4]	Normalized weights	mAP/(%)
[1.0,1.0,1.0,1.0]	[0.25,0.25,0.25,0.25]	75.43
[1.5,1.0,1.0,1.0]	[0.34,0.22,0.22,0.22]	75.62
[2.0,1.0,1.0,1.0]	[0.40,0.20,0.20,0.20]	75.51
[1.0,1.5,1.0,1.0]	[0.22,0.34,0.22,0.22]	75.79
[1.0,2.0,1.0,1.0]	[0.20,0.40,0.20,0.20]	75.61
[1.0,1.0,1.5,1.0]	[0.22,0.22,0.34,0.22]	76.47
[1.0,1.0,2.0,1.0]	[0.20,0.20,0.40,0.20]	75.89
[1.0,1.0,1.0,1.5]	[0.22,0.22,0.22,0.34]	75.74
[1.0,1.0,1.0,2.0]	[0.20,0.20,0.20,0.40]	75.27

Specifically, the direct angular error representation achieves 75.89% mAP, while the AAO-Head can obtain 76.47% mAP increasing 0.58% mAP on the DOTA dataset with single-scale training and testing. In short, the AAO-Head can solve the angular error sensitivity to the aspect ratios and achieve higher mAP.

4) *Effectiveness of the weighted loss:* The loss of our method has four parts: $L_{\text{cls}}^{\text{head}}$, $L_{\text{cls}}^{\text{rpn}}$, $L_{\text{reg}}^{\text{rpn}}$, $L_{\text{cls}}^{\text{head}}$, and $L_{\text{reg}}^{\text{head}}$. We set different weights for each loss to verify the superiority of the weighted loss. The comparison results are shown in Table VI. The first row is the average loss, and the rest are the other weighted losses with different weights. From Table VI,

it can be seen that appropriately increasing the weights of different losses is beneficial to improve the mAP. When the weighting parameters are set to [1.0, 1.0, 1.5, 1.0], our method can achieve 76.47% mAP on the DOTA dataset increasing 1.04% mAP compared with the average loss.

V. Conclusion

This article proposes a novel object detector called RPGAOD for remote-sensing images. The proposed method mainly establishes the GRG-RPN and AAO-Head for more accurate object detection. The GRG-RPN performs a high-quality rotation proposal generation method that directly transforms the horizontal anchors into OBBs by using the geometric relationship between the EHBBs and the OBBs. The AAO-Head solves the angular error sensitivity to the aspect ratios and achieves adaptive angle optimization in the detection head. Besides, the weighted loss is used to get more accurate object detection results. We conduct extensive experiments on two frequently used remote-sensing image datasets. The experiments show that our method can achieve higher detection accuracy in mAP only with fewer regression parameters and a simpler regression approach, and obtains an average 2.5% mAP improvement compared with the other SOTA methods.

References

[1] W. Li, Y. Chen, K. Hu, and J. Zhu, "Oriented RepPoints for aerial object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1819–1828.

[2] C. T. C. Doloriel and R. D. Cajote, "Improving the detection of small oriented objects in aerial images," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. Workshops (WACVW)*, Jan. 2023, pp. 176–185.

[3] Z. Sun, X. Leng, Y. Lei, B. Xiong, K. Ji, and G. Kuang, "BiFA-YOLO: A novel YOLO-based method for arbitrary-oriented ship detection in high-resolution SAR images," *Remote Sens.*, vol. 13, no. 21, p. 4209, Oct. 2021.

[4] 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.

[5] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

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

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[8] 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.

[9] W. Liu et al., "SSD: Single shot MultiBox detector," in *Computer Vision—ECCV 2016*. Cham, Switzerland: Springer, 2016, pp. 21–37.

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

[11] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

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

[13] X. Li et al., "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., Dec. 2020, pp. 21002–21012. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/f0bda020d2470f2e74990a07a607ebd9-Abstract.html

[14] 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.

[15] G. Cheng et al., "Dual-aligned oriented detector," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5618111.

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

[17] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning RoI transformer for oriented object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2844–2853.

[18] X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han, "Oriented R-CNN for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 3500–3509.

[19] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, Austria, May 2021, pp. 1–16. [Online]. Available: https://openreview.net/forum?id=gZ9hCDWe6ke

[20] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision—ECCV 2020*. Cham, Switzerland: Springer, 2020, pp. 213–229.

[21] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[23] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, Austria, May 2021, pp. 1–21. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[24] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Computer Vision—ECCV 2020*. Cham, Switzerland: Springer, 2020, pp. 474–490.

[25] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. 15th Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, Munich, Germany, vol. 11218, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Switzerland: Springer, Sep. 2018, pp. 765–781, doi: 10.1007/978-3-030-01264-9_45.

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

[27] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9756–9765.

[28] X. Xiao, Z. Zhou, B. Wang, L. Li, and L. Miao, "Ship detection under complex backgrounds based on accurate rotated anchor boxes from paired semantic segmentation," *Remote Sens.*, vol. 11, no. 21, p. 2506, Oct. 2019.

[29] L. Li, Z. Zhou, B. Wang, L. Miao, and H. Zong, "A novel CNN-based method for accurate ship detection in HR optical remote sensing images via rotated bounding box," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 686–699, Jan. 2021.

[30] Q. Ming, L. Miao, Z. Zhou, and Y. Dong, "CFC-Net: A critical feature capturing network for arbitrary-oriented object detection in remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5605814.

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

[32] H. Zhou, W. Guo, and Q. Zhao, "An anchor-free network for increasing attention to small objects in high resolution remote sensing images," *Appl. Sci.*, vol. 13, no. 4, p. 2073, Feb. 2023.

[33] X. Jiang, H. Xie, J. Chen, J. Zhang, G. Wang, and K. Xie, "Arbitrary-oriented ship detection method based on long-edge decomposition rotated bounding box encoding in SAR images," *Remote Sens.*, vol. 15, no. 3, p. 673, Jan. 2023.

[34] Y. Wang et al., "Remote sensing image super-resolution and object detection: Benchmark and state of the art," *Expert Syst. Appl.*, vol. 197, Jul. 2022, Art. no. 116793.

[35] C. Deng, D. Jing, Y. Han, S. Wang, and H. Wang, "FAR-Net: Fast anchor refining for arbitrary-oriented object detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[36] X. Zhou et al., "EAST: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2642–2651.

[37] Y. Jiang et al., "R2CNN: Rotational region CNN for orientation robust scene text detection," 2017, *arXiv:1706.09579*.

[38] 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.

[39] X. Yang, J. Yan, Q. Ming, W. Wang, X. Zhang, and Q. Tian, "Rethinking rotated object detection with Gaussian Wasserstein distance loss," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, in Proceedings of Machine Learning Research, vol. 139, M. Meila and T. Zhang, Eds., Jul. 2021, pp. 11830–11841. [Online]. Available: http://proceedings.mlr.press/v139/yang21l.html

[40] X. Yang et al., "Learning high-precision bounding box for rotated object detection via Kullback–Leibler divergence," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., Dec. 2021, pp. 18381–18394. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/98f13708210194c475687be6106a3b84-Abstract.html

[41] C. Zhang, B. Xiong, X. Li, and G. Kuang, "TCD: Task-collaborated detector for oriented objects in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4700714.

[42] Z. Sun et al., "An anchor-free detection method for ship targets in high-resolution SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 7799–7816, 2021.

[43] G. Cheng et al., "Anchor-free oriented proposal generator for object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5625411.

[44] K. Chen et al., "MMDetection: Open MMLab detection toolbox and benchmark," 2019, *arXiv:1906.07155*.

[45] 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, Sep. 2009.

[46] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jun. 2014.

[47] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics*, vol. 10, no. 3, p. 279, Jan. 2021.

[48] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A survey on performance metrics for object-detection algorithms," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Jul. 2020, pp. 237–242.

[49] X. Pan et al., "Dynamic refinement network for oriented and densely packed object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11204–11213.

[50] X. Yang, J. Yan, Z. Feng, and T. He, "R3DET: Refined single-stage detector with feature refinement for rotating object," in *Proc. 35th AAAI Conf. Artif. Intell., 33rd Conf. Innov. Appl. Artif. Intell. (IAAI), 11th Symp. Educ. Adv. Artif. Intell. (EAAI)*, Feb. 2021, pp. 3163–3171. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16426

[51] Z. Chen et al., "PIoU loss: Towards accurate oriented object detection in complex environments," in *Computer Vision—ECCV 2020*. Cham, Switzerland: Springer, 2020, pp. 195–211.

[52] W. Qian, X. Yang, S. Peng, J. Yan, and Y. Guo, "Learning modulated loss for rotated object detection," in *Proc. 35th AAAI Conf. Artif. Intell., 33rd Conf. Innov. Appl. Artif. Intell. (IAAI), 11th Symp. Educ. Adv. Artif. Intell. (EAAI)*, Feb. 2021, pp. 2458–2466. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16347

[53] Q. Ming, Z. Zhou, L. Miao, H. Zhang, and L. Li, "Dynamic anchor learning for arbitrary-oriented object detection," in *Proc. 35th AAAI Conf. Artif. Intell., 33rd Conf. Innov. Appl. Artif. Intell. (IAAI), 11th Symp. Educ. Adv. Artif. Intell. (EAAI)*, Feb. 2021, pp. 2355–2363. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16336

[54] Y. Yang, J. Chen, X. Zhong, and Y. Deng, "Polygon-to-polygon distance loss for rotated object detection," in *Proc. 36th AAAI Conf. Artif. Intell., 34th Conf. Innov. Appl. Artif. Intell. (IAAI), 12th Symp. Educ. Adv. Artif. Intell. (EAAI)*, Feb./Mar. 2022, pp. 3072–3080. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/20214

[55] L. Hou, K. Lu, J. Xue, and Y. Li, "Shape-adaptive selection and measurement for oriented object detection," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, vol. 36, no. 1, pp. 923–932.

[56] J. Han, J. Ding, J. Li, and G.-S. Xia, "Align deep features for oriented object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5602511.

[57] J. Han, J. Ding, N. Xue, and G.-S. Xia, "ReDet: A rotation-equivariant detector for aerial object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2785–2794.

[58] J. Wang, J. Ding, H. Guo, W. Cheng, T. Pan, and W. Yang, "Mask OBB: A semantic attention-based mask oriented bounding box representation for multi-category object detection in aerial images," *Remote Sens.*, vol. 11, no. 24, p. 2930, Dec. 2019.

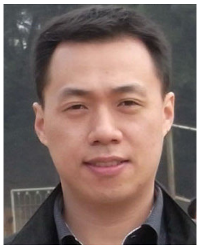
[59] J. Wang, W. Yang, H.-C. Li, H. Zhang, and G.-S. Xia, "Learning center probability map for detecting objects in aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4307–4323, May 2021.

Yajun Qiao received the B.S. degree in measurement and control technology and instrument from the School of Information Engineering, Taiyuan University of Technology (TYUT), Taiyuan, China, and the M.S. degree in automation from the School of Automation, Beijing Institute of Technology (BIT), Beijing, China, in 2017 and 2020, respectively, where he is currently pursuing the Ph.D. degree.

His research interests include autonomous driving, object detection, and remote sensing image understanding.

Lingjuan Miao received the B.S. and M.S. degrees in control theory and engineering from the Harbin Institute of Technology, Harbin, China, in 1986 and 1989, respectively, and the Ph.D. degree in control theory and engineering from the China Academy of Launching Vehicle Technology, Beijing, China, in 2001.

Since 1992, she has been a Lecturer with the Beijing Institute of Technology, Beijing, where she has been an Associate Professor since 1996 and a Professor since 2001. Her main research interests include GPS, inertial navigation systems, INS/GPS integrated navigation, and multisensor fusion technique.



Zhiqiang Zhou (Member, IEEE) received the B.S. degree in automation and the Ph.D. degree in control science and engineering from the School of Automation, Beijing Institute of Technology (BIT), Beijing, China, in 2004 and 2009, respectively.

From 2009 to 2012, he was a Post-Doctoral Researcher with the Institute of Automation, Chinese Academy of Sciences, Beijing. He is currently an Associate Professor with the School of Automation, BIT. His research interests include information fusion, pattern recognition, digital image processing, and vision-based navigation.

Qi Ming received the B.S. degree in automation from the School of Automation, Beijing Institute of Technology (BIT), Beijing, China, in 2018, where he is currently pursuing the Ph.D. degree in navigation, guidance, and control.

His research interests include computer vision, object detection, and remote sensing image analysis.