Unified Five-Distance Bounding Box Representation for Remote Sensing Oriented Object Detection

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

***Abstract*—Currently, due to the densely packed and varied orientations of objects in the remote sensing images, oriented object detection is still a challenging task. In the existing methods, the oriented bounding box (OBB) representations either include angle parameters or require a larger number of parameters and additional processes to obtain the final rotated rectangles, which can potentially induce positional errors and decrease the accuracy of object detection in remote sensing images. In this article, we propose a unified five-distance bounding box representation for remote sensing-oriented object detection, which gets rid of the utilization of angle parameters and simultaneously represents the OBBs with the least distance-based parameters. We then construct a unified five-distance region proposal network (UF-RPN) based on the new OBB representation to achieve higher performance of oriented object detection. In addition, most existing two-stage methods assume that the localization errors of objects are relatively small in the second stage of the network and believe that directly regressing the errors of a general OBB representation would achieve sufficiently high performance of object detection, neglecting that there is still room for improvement in the detection head by utilizing a more advanced OBB representation. Therefore, a five-distance detection head (FD-Head) is developed by utilizing the five-distance OBB representation in the detection head. Experimental results on the HRSC2016, DIOR-R, DOTA-v1.0, and DOTA-v2.0 datasets demonstrate the superiority and robustness of our method. Our code is publicly available at https://github.com/QYJ123/UFDet**

***Index Terms*—Five-distance detection head (FD-Head), oriented object detection, unified five-distance oriented bounding box (OBB) representation, unified five-distance region of proposal network (UF-RPN).**

I. Introduction

With the development of computer vision, oriented object detection has been a fundamental task in the remote sensing field. This technology is also crucial for various applications such as city planning, traffic management, and military surveillance [1], [2], [3], [4], [5]. Moreover, with the advancement of convolutional neural networks (CNNs) and the considerable growth of available remote sensing images in recent years, oriented object detection is gradually attracting the attention of many researchers for the relevant applications.

Inspired by the success of existing object detection methods [6], [7], [8], [9], [10], [11], many researchers have attempted to achieve the object detection by directly regressing the horizontal bounding boxes (x, y, w, h) in remote sensing images. This bounding box representation uses (x, y) to represent the center point coordinates of the bounding boxes, w and h to denote the width and height of the bounding boxes. However, the objects in remote sensing images are typically densely packed and exhibit arbitrary orientations. As a result, the horizontal bounding boxes fail to tightly enclose the objects, leading to the poor performance of object detection. To address this problem, the oriented object detection gradually flourishes for achieving accurate object detection in remote sensing images, which uses the general oriented bounding boxes (OBBs) $(x_r, y_r, w_r, h_r, \theta_r)$ to localize the objects by incorporating the orientation parameter θ_r.

Most of the existing methods achieve higher performance of oriented object detection by transforming the preset horizontal anchors into rotational proposals in the region proposal network (RPN) and subsequently optimizing these rotational proposals in the detection head. Although these methods can obtain satisfactory performance in oriented object detection, there are still some issues that influence the accuracy of oriented object detection in remote sensing images.

The OBB representation is usually used to guide networks to localize objects and ultimately determines the accuracy of oriented object detection in remote sensing images. In the existing methods, the OBB representations either include angle parameters or require a larger number of parameters and additional processes to obtain the final rotated rectangles. As shown in Fig. 1, there are five typical OBB representations in existing methods, i.e., FCOS-O [12], Faster R-CNN-O [8], Gliding Vertex [13], Oriented R-CNN [14], and QPDet [15]. The OBB representations in Faster R-CNN-O, FCOS-O and QPDet rely on angle parameters [as shown in Fig. 1(a), (b), and (e)], leading to some limitations such as angle periodicity, boundary discontinuity [15], [16], [17], and differences in parameter dimensionality (e.g., distance parameters (x_r, y_r, w_r, h_r) and angle parameter θ_r), and thus restricting the performance of object detection. Moreover, some methods [13], [14] attempt to discard the angle parameter, but they require a larger number of parameters and additional processes to obtain the final rotated rectangles. For example, Gliding Vertex [as shown in Fig. 1(c)] needs to first generate the irregular quadrilaterals, while oriented R-CNN [as shown in Fig. 1(d)] has to generate the parallelograms first before obtaining the final rotated rectangles. These intermediate steps usually influence the regression results and, in turn, compromise the overall performance. In this article, we propose a unified five-distance OBB representation $(x_r, y_r, L_a, L_b, L_h)$, as shown in Fig. 1(f), which not only fully gets rid of the angle parameter, but also

Received 30 December 2024; revised 29 May 2025 and 17 July 2025; accepted 19 August 2025. Date of publication 22 August 2025; date of current version 4 September 2025. *(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; wayhao@bit.edu.cn).

Digital Object Identifier 10.1109/TGRS.2025.3601549

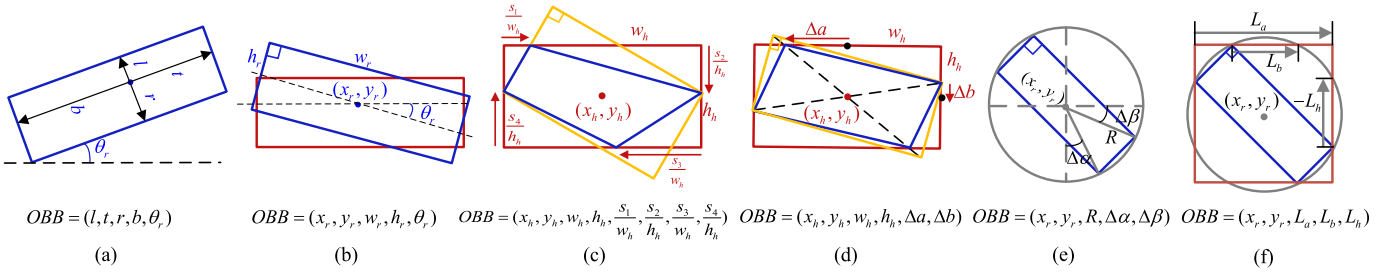

Fig. 1. Comparison of different OBB representations in the oriented object detection methods. (a) FCOS-O. (b) Faster R-CNN-O. (c) Gliding vertex. (d) Oriented R-CNN. (e) QPDet. (f) Our method.

directly determines the OBB positions based on the five (the least number of parameters for OBB representation) distance-based parameters without additional process. Moreover, based on the new five-distance OBB representation, we establish a unified five-distance RPN (UF-RPN), which can generate rotated proposals in one step and achieve higher performance of object detection.

In addition, most existing two-stage methods assume that the localization errors of objects are relatively small in the second stage and believe that directly regressing the errors of general OBB representation $(x_r, y_r, w_r, h_r, \theta_r)$ would achieve sufficiently high performance of object detection, neglecting that there is still room for improvement in the detection head by utilizing a more advanced OBB representation for two-stage detectors. Notably, the QPDet adopts a new OBB representation $(x_r, y_r, R, \Delta\alpha, \Delta\beta)$ in two-stage regressions of the network to achieve consistent regression of bounding boxes. However, we found that the consistency of OBB representation in two-stage regressions is insufficient to ensure optimal object detection results (as revealed by the ablation studies in Section IV-D3). The detection head also needs a well-performed OBB representation, which has usually been overlooked in current research, in order to further improve the accuracy of oriented object detection. Therefore, we establish the five-distance detection head (FD-Head) by utilizing the unified five-distance OBB representation $(x_r, y_r, L_a, L_b, L_h)$ in the detection head to achieve more accurate oriented object detection. Compared with the general detection head, the FD-Head uses unified five-distance parameters, avoiding the difficulty of directly regressing the angles in the detection head, thereby improving the performance of object detection in remote sensing images. Besides, the shape-constraint loss (SC-Loss) is used to accelerate the convergence of the model and achieve more accurate object detection by combining the smooth-L1 loss and shape constraint of objects to optimize the UF-RPN and FD-Head.

The main contributions of this article are summarized as follows.

1) We propose a unified five-distance OBB representation, avoiding the direct utilization of angle parameters, and use the least five-distance parameters to represent the OBBs. By using the new five-distance OBB representation, the UF-RPN can generate high-quality rotational proposals from the horizontal anchors in one step without additional processes to improve the performance of oriented object detection.
2) We found that there is still room for improvement in the detection head by utilizing a special OBB representation. Therefore, the FD-Head is developed by using the unified five-distance OBB representation in the detection head to achieve more accurate object detection in remote sensing images.
3) Extensive experiments on four public datasets show that our method achieves higher performance in object detection compared with other state-of-the-art (SOTA) methods in remote sensing images.

II. Related Work

With the development of CNNs in recent years, the object detection task has also achieved great success in the field of computer vision. Current object detection methods [18], [19], [20], [21], [22] primarily utilize horizontal bounding boxes to localize objects and identify specific categories of objects in natural images. As the object detection is gradually applied to remote sensing images [23], [24], [25], oriented object detection, which adopts OBBs to tightly surround the objects, achieves higher performance in the object detection task and has received much attention in recent years.

A. Horizontal Object Detection

In the mainstream object detection methods based on CNNs, the horizontal bounding boxes are usually used to achieve localization and classification of objects in natural images. These methods are termed as horizontal object detection [6], [7], [11], [20], and can be divided into two-stage and one-stage methods.

The two-stage methods usually utilize an RPN network to generate coarse bounding boxes of objects, and a detection head to refine the bounding box localization and categorization of the objects, for example, R-CNN [6], Fast R-CNN [7], Faster R-CNN [8], Mask R-CNN [9], and Cascade R-CNN [26]. R-CNN uses CNNs to extract feature maps based on each preselected region and subsequently classifies the objects based on these maps, leading to more time being consumed to achieve the object detection task. Compared with R-CNN, Fast R-CNN and Faster R-CNN directly obtain feature maps of the whole images and then use ROI pooling to extract regional feature maps from the complete feature maps, thereby increasing the speed and accuracy of object detection. Mask R-CNN and Cascade R-CNN can further improve the performance of horizontal object detection by adding a mask branch or employing multiple regressions in the detection head. FPN [27] generates fused feature maps by combining detailed information from lower levels with

semantic information from higher levels, thereby enhancing the accuracy of object detection.

Besides, the one-stage methods directly obtain the regression and classification results based on the feature maps, which usually increases the detection speed but decreases the accuracy of object detection, such as YOLO [10] and Focal loss [11]. YOLO can directly achieve classification and regression of objects based on the feature maps and preset anchors with faster detection speed. However, this also results in a certain degree of decrease in the accuracy of object detection. Focal loss improves the loss function, solving the imbalance of positive and negative samples in the model training and testing processes, which is used to achieve more accurate object detection. Lastly, the anchor-free object detection methods [12], [20], [21], [22], [28], [29], [30], [31] and multimodal models [32] are recently developed, which can also accurately predict the bounding boxes and categories of objects based on the feature maps. However, these methods usually require more training time and greater computational resources.

B. Oriented Object Detection

With the availability of massive remote sensing images in recent years, OBBs can be used to achieve more accurate localization of objects and enhance the performance of object detection in the remote sensing field. Consequently, the oriented object detection [2], [33], [34], [35], [36], [37], [38] has gradually attracted many researchers aiming to obtain more accurate object detection in remote sensing images. In existing studies, oriented object detection methods can be classified into three categories.

One category of methods achieves oriented object detection based on rotated anchors. The rotated RPN [39] obtains the rotational proposals by decoding the regression outputs and the predefined rotated anchors generated with different widths, heights, and angles. The number of anchors would increase proportionally with the number of angles, which results in an imbalance of the positive and negative samples, decreasing the object detection performance. The DRBox [40] and R3Det [41] achieve higher performance of object detection based on the rotated anchors by designing new angle regression parameters or feature refinement modules. Another category of methods achieves oriented object detection based on horizontal anchors. RR-CNN [42] and Faster R-CNN-O improve the performance of object detection by employing rotating RoI pooling to align the oriented feature maps based on the horizontal anchors. GWD [43], KLD [44], and KFIoU [45] regard the OBBs as 2-D Gaussian distributions and use different measurement approaches to evaluate the loss of bounding boxes. TIOE [46] proposes task interleaving and orientation estimation modules to improve the accuracy of the oriented detection. ReDet [47] and FPNFormer [48] separate the rotation-invariant features and the rotation-equivariant features to improve the performance of oriented object detection. GF-CSL [49] employs an aspect-ratio-aware factor in the loss function to measure the orientations of objects, improving the accuracy of square-like objects in remote sensing images. RoI Trans [50] achieves oriented object detection by adding an RRoI learner network to achieve the feature transformation from horizontal bounding boxes to OBBs. Gliding Vertex [13] and Oriented R-CNN [14] use more parameters to achieve the OBB representation and the transformation from horizontal anchors to rotational proposals, which both require additional processes to obtain the final rotational proposals and are easy to induce localization errors. QPDet [15] proposes a new OBB representation based on a polar coordinate system and achieves the consistent regression of bounding boxes in the network. In addition, oriented object detection can also be achieved based on anchor-free methods. The anchor-free methods [51], [52], [53], [54] achieve oriented object detection by directly obtaining the detection results based on each feature point on the feature maps without needing predefined anchors, which usually need more time to train the network.

In this article, we propose a unified five-distance detector (UFDet) for oriented object detection, which utilizes the unified five-distance OBB representation to establish the UF-RPN and FD-Head to achieve higher performance of oriented object detection. The UF-RPN can directly transform the horizontal anchors into rotational bounding boxes in one step without additional processing based on the unified five-distance parameters. The FD-Head utilizes the new five-distance OBB representation in the detection head to further improve the performance of oriented object detection.

III. Proposed Method

The overall architecture of our method is illustrated in Fig. 2. Our method mainly establishes two distinctive components, namely, the UF-RPN and FD-Head, by leveraging a unified five-distance OBB representation. The unified five-distance OBB representation is based on the mathematical relationships of different parameters to determine the localization of the OBBs and represents the five distances formed by the OBBs, external horizontal bounding boxes (EHBBs), and minimum enclosing circumcircles (MECs) of the OBBs (as shown in Fig. 3, which will be detailed in Section III-A). The UF-RPN generates high-quality rotational proposals from the horizontal anchors based on the unified five-distance OBB representation. The FD-Head also utilizes the new unified five-distance OBB representation in the detection head to design the regression parameters and further improve the performance of oriented object detection. Therefore, our method can achieve more accurate object detection results in remote sensing images. In the following, we first introduce the unified five-distance OBB representation, and then describe the two networks as well as the other relevant parts of the proposed algorithm in detail. Besides, we list the mathematical notations used in the following sections and corresponding meanings in Table I.

A. Unified Five-Distance OBB Representation

The OBB representation usually can guide the network to localize the objects in the remote sensing images, which also determines the complexity and the performance of object detection methods. In the general OBB representation $(x_r, y_r, w_r, h_r, \theta_r)$, the angle θ_r often causes the periodicity of angles and the notorious boundary discontinuity problems

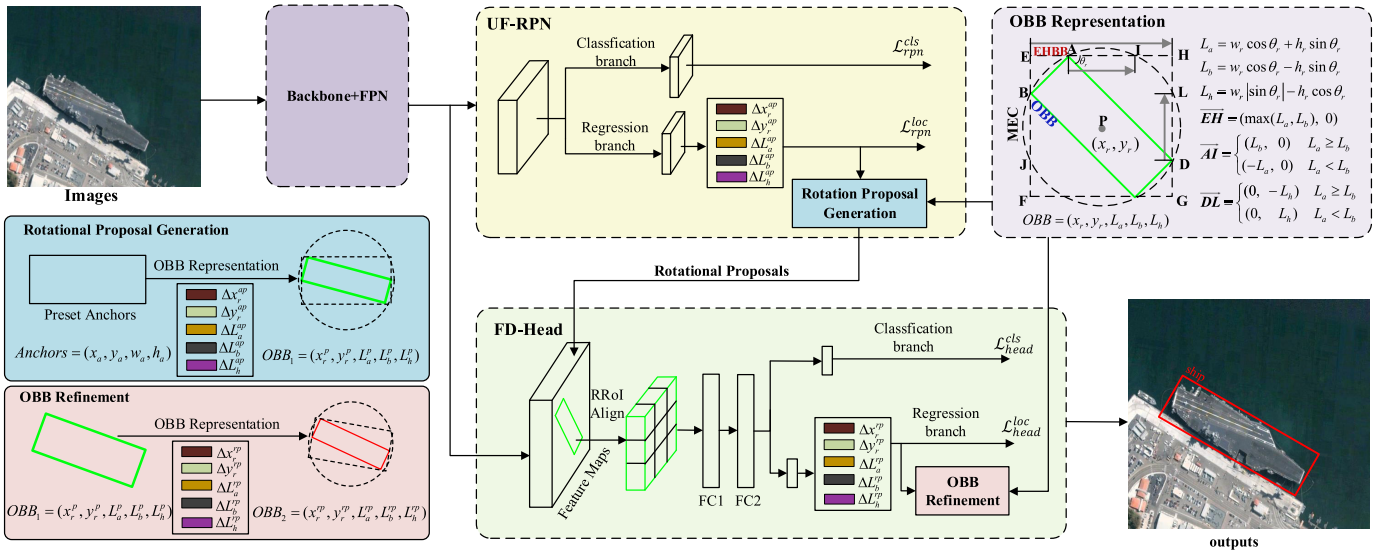

Fig. 2. Overall architecture of the proposed method. The proposed method mainly contains four parts: backbone, FPN, UF-RPN, and FD-Head. The UF-RPN and FD-Head are both based on the unified five-distance OBB representation.

TABLE I

SUMMARIZATION OF MATHEMATICAL NOTATIONS AND CORRESPONDING MEANINGS

Modules	Mathematical notations	Corresponding meanings
-	(x, y, w, h)	General horizontal bounding boxes
-	$(x_r, y_r, w_r, h_r, \theta_r)$	General OBB representation
-	(l, t, r, b, θ_r)	OBB representation in FCOS-O
-	$(x_h, y_h, w_h, h_h, \frac{s_1}{w_h}, \frac{s_2}{h_h}, \frac{s_3}{w_h}, \frac{s_4}{h_h})$	OBB representation in Gliding Vertex
-	$(x_h, y_h, w_h, h_h, \Delta a, \Delta b)$	OBB representation in Oriented R-CNN
-	$(x_r, y_r, L_a, L_b, L_h)$	Proposed five-distance OBB representation
UF-RPN	(x_a, y_a, w_a, h_a)	Preset horizontal anchors
UF-RPN	$(\Delta x_r^{ag}, \Delta y_r^{ag}, \Delta L_a^{ag}, \Delta L_b^{ag}, \Delta L_h^{ag})$	Ground-truth anchor offsets in UF-RPN
UF-RPN	$(\Delta x_r^{ap}, \Delta y_r^{ap}, \Delta L_a^{ap}, \Delta L_b^{ap}, \Delta L_h^{ap})$	Predicted anchor offsets in UF-RPN
UF-RPN	$(x_r^p, y_r^p, L_a^p, L_b^p, L_h^p)$	Predicted OBBs in UF-RPN
UF-RPN /FD-Head	$(x_r^g, y_r^g, w_r^g, h_r^g, \theta_r^g)$	Ground-truth OBBs
UF-RPN /FD-Head	$(x_r^p, y_r^p, w_r^p, h_r^p, \theta_r^p)$	Predicted rotation proposals
FD-Head	$(\Delta x_r^{pg}, \Delta y_r^{pg}, \Delta L_a^{pg}, \Delta L_b^{pg}, \Delta L_h^{pg})$	Ground-truth proposal offsets in FD-Head
FD-Head	$(\Delta x_r^{rp}, \Delta y_r^{rp}, \Delta L_a^{rp}, \Delta L_b^{rp}, \Delta L_h^{rp})$	Predicted proposal offsets in FD-Head
FD-Head	$(\Delta x_h^{pg}, \Delta y_h^{pg})$	Center point errors before RROI Align module
FD-Head	$(x_r^{rp}, y_r^{rp}, L_a^{rp}, L_b^{rp}, L_h^{rp})$	The final predicted OBBs
UF-RPN	$\mathcal{L}_{rpn}^{cls}, \mathcal{L}_{rpn}^{loc}$	UF-RPN classification and localization losses
FD-Head	$\mathcal{L}_{head}^{cls}, \mathcal{L}_{head}^{loc}$	FD-Head classification and localization losses
UF-RPN/FD-Head	$\mathcal{L}_{sc}$	Shape-constraint SC-Loss
UF-RPN/FD-Head	$\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4$	Four parts of loss in SC-Loss

[15], [16], [17], which decreases the accuracy of oriented object detection. In this article, we propose a unified five-distance OBB representation $(x_r, y_r, L_a, L_b, L_h)$, avoiding the direct utilization of angle parameters and representing the OBBs based on the least five-distance parameters without additional processes.

Similar to the general OBB representation, the five-distance OBB representation also uses (x_r, y_r) to represent the center point coordinates of OBBs, which also denotes the distances from the OBB center points to the origins of the images in both horizontal and vertical directions.

We then define the other three distances (L_a, L_b, L_h) in the five-distance OBB representation to determine the OBBs. In the following, we use $\square_{ABCD}$, $\square_{EFGH}$, and $\odot P$ to represent the OBB, EHBB, and MEC, respectively [as shown in Fig. 3(a) or (b)]. Specifically, points A, B, C, and D denote the four vertices of the OBB. Likewise, points E, F, G, and H denote the four vertices of the EHBB, while point P denotes the center of the MEC.

Note that (x_r, y_r) represents the same center point coordinates for $\square_{ABCD}$, $\square_{EFGH}$ and $\odot P$. Consequently, $\odot P$ is

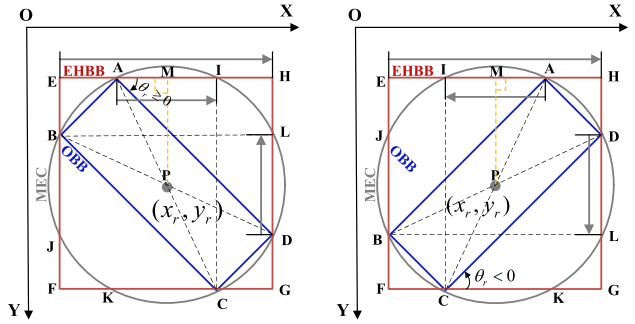

Fig. 3. Position relationship of a single OBB, EHBB, and MEC in different angle conditions. All angles of OBBs can be divided into two cases. (a) $\theta_r \geq 0$. (b) $\theta_r < 0$.

uniquely determined by its diameter $|\overrightarrow{BD}|$ (the length of vector $\overrightarrow{BD}$), which is calculated as follows:

$$\left|\overrightarrow{BD}\right| = \sqrt{\left|\overrightarrow{EH}\right|^2 + \left|\overrightarrow{DL}\right|^2}. \tag{1}$$

$\square_{EFGH}$ is determined by its width ($|\overrightarrow{EH}|$) and height ($|\overrightarrow{HG}|$), where $|\overrightarrow{HG}|$ can be calculated as follows:

$$\left|\overrightarrow{HG}\right| = \sqrt{\left|\overrightarrow{AC}\right|^2 - \left|\overrightarrow{AI}\right|^2}. \tag{2}$$

Considering that $|\overrightarrow{ac}|$ is equal to $|\overrightarrow{BD}|$, and according to (1), $|\overrightarrow{HG}|$ can be expressed as follows:

$$\left|\overrightarrow{HG}\right| = \sqrt{\left|\overrightarrow{EH}\right|^2 + \left|\overrightarrow{DL}\right|^2 - \left|\overrightarrow{AI}\right|^2}. \tag{3}$$

n a word, the diameter of $\odot P$, and the width and height of $\square_{EFGH}$ can all be calculated by using $\overrightarrow{EH}$, $\overrightarrow{AI}$, and $\overrightarrow{DL}$. Moreover, points A, B, C, and D are located at the intersections of $\odot P$ and $\square_{EFGH}$. Given $\overrightarrow{AI}$, $\overrightarrow{DL}$, and $\odot P$, the positions of points A and D can be determined, considering that $\overrightarrow{AI}$ passes through $\odot P$ horizontally. The positions of points C and B can then be determined as symmetrical to points A and D with

respect to point P. As a result, the OBB of object $\square_{ABCD}$ can finally be determined by $\overrightarrow{EH}$, $\overrightarrow{AI}$, and $\overrightarrow{DL}$. However, these vectors have different calculation formulas under different angle conditions [as shown in the following (4) and (5)]. To solve this issue, we construct an alternative parameters (L_a, L_b, L_h) to represent $(\overrightarrow{EH}, \overrightarrow{AI}, \overrightarrow{DL})$ irrespective of the angle cases. The following provides the detailed derivation of (L_a, L_b, L_h).

In general OBB representation, the range of θ_r is $[-90, 90)$. As shown in Fig. 3(a), when $\theta_r \geq 0$, $(\overrightarrow{EH}, \overrightarrow{AI}, \overrightarrow{DL})$ can be expressed as

$$\begin{cases} \overrightarrow{EH} = (w_r \cos\theta_r + h_r \sin\theta_r,\ 0) \\ \overrightarrow{AI} = (w_r \cos\theta_r - h_r \sin\theta_r,\ 0) \\ \overrightarrow{DL} = (0,\ h_r \cos\theta_r - w_r |\sin\theta_r|)\,. \end{cases} \tag{4}$$

When $\theta_r < 0$ [as shown in Fig. 3(b)], $(\overrightarrow{EH}, \overrightarrow{AI}, \overrightarrow{DL})$ can be expressed as

$$\begin{cases} \overrightarrow{EH} = (w_r \cos\theta_r - h_r \sin\theta_r,\ 0) \\ \overrightarrow{AI} = (-w_r \cos\theta_r - h_r \sin\theta_r,\ 0) \\ \overrightarrow{DL} = (0,\ w_r |\sin\theta_r| - h_r \cos\theta_r)\,. \end{cases} \tag{5}$$

Based on (4) and (5), we can define (L_a, L_b, L_h) as

$$\begin{cases} L_a = w_r \cos\theta_r + h_r \sin\theta_r \\ L_b = w_r \cos\theta_r - h_r \sin\theta_r \\ L_h = w_r |\sin\theta_r| - h_r \cos\theta_r. \end{cases} \tag{6}$$

Then, considering (6) and the abovementioned two cases, $(\overrightarrow{EH}, \overrightarrow{AI}, \overrightarrow{DL})$ can be calculated as follows:

$$\begin{aligned} \overrightarrow{EH} &= (\max(L_a, L_b)\,,\ 0) \\ \overrightarrow{AI} &= \begin{cases} (L_b,\ 0)\,, & L_a \geq L_b \\ (-L_a,\ 0)\,, & L_a < L_b \end{cases} \\ \overrightarrow{DL} &= \begin{cases} (0,\ -L_h)\,, & L_a \geq L_b \\ (0,\ \ L_h)\,, & L_a < L_b \end{cases} \end{aligned} \tag{7}$$

where $L_a \geq L_b$ corresponds to the case $\theta_r \geq 0$, and $L_a < L_b$ corresponds to the case $\theta_r < 0$. As a result, the five-distance OBB representation can determine the OBBs by using (L_a, L_b, L_h) to represent $(\overrightarrow{EH}, \overrightarrow{AI}, \overrightarrow{DL})$.

Using (6), we can also transform the general OBB representation into the five-distance OBB representation, which enables the existing OBB annotations to be converted to the five-distance representations without reannotation.

Besides, we can also derive the transformation from the five-distance OBB representation to the general OBB representation. To this end, we define an auxiliary variable L_d as follows:

$$L_d = w_r |\sin\theta_r| + h_r \cos\theta_r. \tag{8}$$

When $L_a \geq L_b$, L_d can be expressed as

$$L_d = \sqrt{L_h^2 + L_a^2 - L_b^2} \tag{9}$$

and when $L_a < L_b$, L_d can be expressed as

$$L_d = \sqrt{L_h^2 + L_b^2 - L_a^2}. \tag{10}$$

Thus, considering the above two cases, L_d can be calculated by L_a, L_b, and L_h as follows:

$$L_d = \sqrt{L_h^2 + \left|L_a^2 - L_b^2\right|}. \tag{11}$$

Next, the following equations can be obtained from (6) and (8):

$$\begin{cases} w_r \cos\theta_r = \dfrac{L_a + L_b}{2} \\ h_r \sin\theta_r = \dfrac{L_a - L_b}{2}, \end{cases} \quad \begin{cases} w_r |\sin\theta_r| = \dfrac{L_d + L_h}{2} \\ h_r \cos\theta_r = \dfrac{L_d - L_h}{2} \end{cases} \tag{12}$$

where L_d can be calculated according to (11). Finally, w_r, h_r, and θ_r can be calculated by

$$\begin{cases} w_r = 0.5\sqrt{(L_a + L_b)^2 + (L_d + L_h)^2} \\ h_r = 0.5\sqrt{(L_a - L_b)^2 + (L_d - L_h)^2} \\ \theta_r = \arctan\dfrac{L_a - L_b}{L_d - L_h}. \end{cases} \tag{13}$$

Consequently, we can transform the five-distance OBB representation into the general OBB representation based on (11) and (13), which enables us to plot the OBBs of the objects by using the five-distance OBB representation.

The above new OBB representation utilizes the least and unified five-distance parameters to represent the OBBs and can avoid the direct regression of angle parameters, making it easier to optimize and achieve more accurate prediction results. The following will further describe the UF-RPN and the FD-Head based on the new five-distance OBB representation.

B. UF-RPN

The UF-RPN is a significant network for generating high-quality rotational proposals from horizontal anchors, which includes a classification branch and a regression branch. The classification branch aims to predict the probability for each anchor that contains an object. The regression branch predicts the offsets of the anchors relative to the rotational proposals. In the UF-RPN, we adopt new five-distance offsets based on the five-distance OBB representation to indicate the difference between the horizontal anchors and the rotational proposals for achieving higher performance of object detection.

As shown in Fig. 4, the rotational proposals are also the outputs of UF-RPN, obtained by decoding the predicted anchor offsets and the preset horizontal anchors in the training and testing processes. The ground-truth anchor offsets are only used as supervisory information for training, obtained by the parameterization process to indicate the difference between the ground-truth boxes and the anchors. With the predicted anchor offsets gradually getting closer to the ground-truth anchor offsets, the rotational proposals will steadily converge to the ground-truth boxes. In the following, we give the decoding and parameterization processes of UF-RPN.

The parameterization process is to mathematically represent the difference between the ground-truth bounding boxes $(x_r^g, y_r^g, w_r^g, h_r^g, \theta_r^g)$ and the preset horizontal anchors (x_a, y_a, w_a, h_a), which is also the ground-truth of the UF-RPN regression. Based on the anchors and the new five-distance

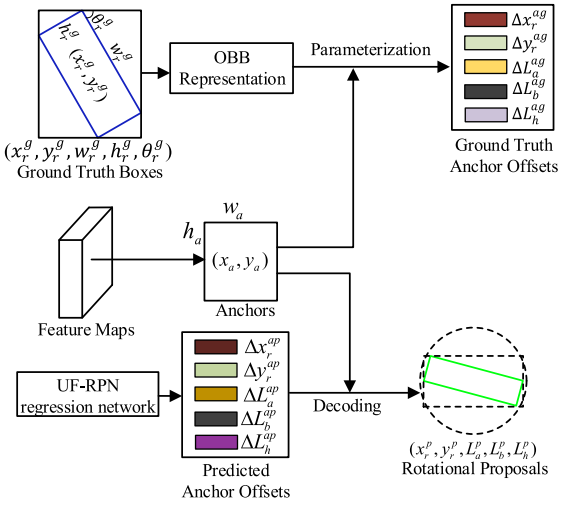

Fig. 4. Process of UF-RPN obtains the rotational proposals from horizontal anchors.

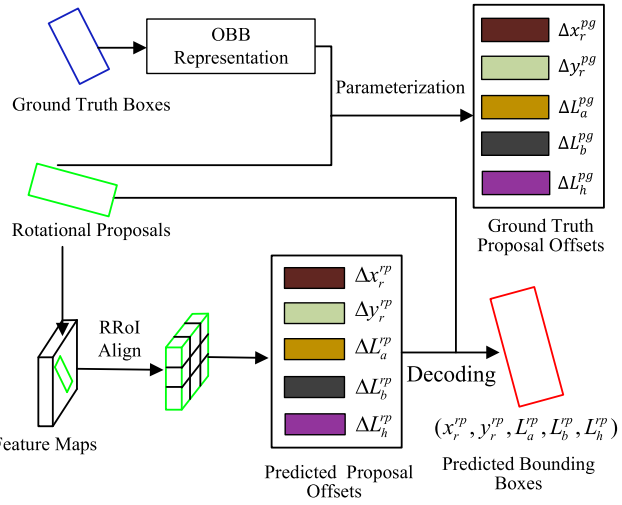

Fig. 5. Process of FD-Head obtains the final predicted bounding boxes from the rotational proposals.

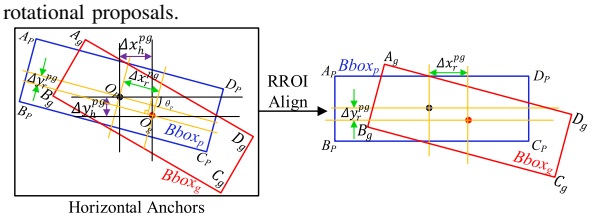

Fig. 6. Difference between the ground-truth errors of center points and the center point errors before RROI align.

OBB representation obtained by transforming the ground-truth boxes according to (6), the ground-truth anchor offsets $(\Delta x_r^{\mathrm{ag}}, \Delta y_r^{\mathrm{ag}}, \Delta L_a^{\mathrm{ag}}, \Delta L_b^{\mathrm{ag}}, \Delta L_h^{\mathrm{ag}})$ can be defined as follows:

$$\begin{cases} \Delta x_r^{\mathrm{ag}} = \dfrac{x_r^g - x_a}{\sqrt{w_a^2 + h_a^2}} \\ \Delta y_r^{\mathrm{ag}} = \dfrac{y_r^g - y_a}{\sqrt{w_a^2 + h_a^2}} \\ \Delta L_a^{\mathrm{ag}} = \dfrac{w_r^g \cos\theta_r^g + h_r^g \sin\theta_r^g}{\sqrt{w_a^2 + h_a^2}} \\ \Delta L_b^{\mathrm{ag}} = \dfrac{w_r^g \cos\theta_r^g - h_r^g \sin\theta_r^g}{\sqrt{w_a^2 + h_a^2}} \\ \Delta L_h^{\mathrm{ag}} = \dfrac{w_r^g |\sin\theta_r^g| + h_r^g \cos\theta_r^g}{\sqrt{w_a^2 + h_a^2}}. \end{cases} \tag{14}$$

Besides, the predicted OBBs $(x_r^p, y_r^p, L_a^p, L_b^p, L_h^p)$ can be obtained by decoding the anchors and the predicted anchor offsets $(\Delta x_r^{\mathrm{ap}}, \Delta y_r^{\mathrm{ap}}, \Delta L_a^{\mathrm{ap}}, \Delta L_b^{\mathrm{ap}}, \Delta L_h^{\mathrm{ap}})$ of the UF-RPN regression branch. The specific formulas of the predicted OBBs $(x_r^p, y_r^p, L_a^p, L_b^p, L_h^p)$ in UF-RPN are computed as follows:

$$\begin{cases} x_r^p = x_a + \Delta x_r^{\mathrm{ap}} \sqrt{w_a^2 + h_a^2} \\ y_r^p = y_a + \Delta y_r^{\mathrm{ap}} \sqrt{w_a^2 + h_a^2} \\ L_a^p = \Delta L_a^{\mathrm{ap}} \sqrt{w_a^2 + h_a^2} \\ L_b^p = \Delta L_b^{\mathrm{ap}} \sqrt{w_a^2 + h_a^2} \\ L_h^p = \Delta L_h^{\mathrm{ap}} \sqrt{w_a^2 + h_a^2}. \end{cases} \tag{15}$$

In the proposed UF-RPN, the predicted OBBs also represent the high-quality rotational proposals. With the newly defined five-distance regressing parameters and the corresponding decoding process, the UF-RPN can generate high-quality rotational proposals in one step without additional process. These rotational proposals are then employed as the basis for bounding boxes to obtain more accurate object detection results.

C. FD-Head

Most of the existing two-stage methods achieve higher performance of object detection by further obtaining more accurate classification and regression based on the rotational proposals. However, these methods usually use various special OBB representations for generating rotational proposals in the RPN, while using the general OBB representation $(x_r, y_r, w_r, h_r, \theta_r)$ to represent the OBBs in the detection head. These methods neglect that there is still room for improvement in the detection head by utilizing a more advanced OBB representation for a two-stage network. Therefore, we propose the FD-Head, which employs the new unified five-distance OBB representation in the detection head to achieve more accurate object detection in remote sensing images.

As shown in Fig. 5, the FD-Head can obtain the final predicted bounding boxes by decoding the predicted proposal offsets and the rotational proposals in the training and testing processes, which are also the outputs of the overall network. In the training process, the FD-Head also needs the parameterization to calculate the ground-truth proposal offsets based on the new five-distance OBB representation, supervising the network training. The parameterization process in detection head is to mathematically represent the difference between the ground-truth boxes and the predicted OBBs $(x_r^p, y_r^p, L_a^p, L_b^p, L_h^p)$, which also can be transformed to the rotational proposals $(x_r^p, y_r^p, w_r^p, h_r^p, \theta_r^p)$ using (11) and (13). Compared with the UF-RPN, the FD-Head utilizes the RRoI Align module to align the feature maps and further predict more accurate object detection results, which also needs to be considered in the parameterization process.

As shown in Fig. 6, due to the existing of RROI Align module in detection head, the ground-truth errors of center point $(\Delta x_r^{\mathrm{pg}}, \Delta y_r^{\mathrm{pg}})$ after RROI align are different from the directly calculated errors of center point $(\Delta x_h^{\mathrm{pg}}, \Delta y_h^{\mathrm{pg}})$ before

RROI align. This difference appears because the RROI Align module extracts the object feature maps by rotating angle θ_r^p and cropping the overall image feature maps according to the rotation proposals. As a result, we can derive the mathematical relationship between $(\Delta x_r^{\text{pg}}, \Delta y_r^{\text{pg}})$ and $(\Delta x_h^{\text{pg}}, \Delta y_h^{\text{pg}})$ as follows:

$$\begin{cases} \Delta x_r^{\text{pg}} = \Delta x_h^{\text{pg}} \cos\theta_r^p + \Delta y_h^{\text{pg}} \sin\theta_r^p \\ \Delta y_r^{\text{pg}} = -\Delta x_h^{\text{pg}} \sin\theta_r^p + \Delta y_h^{\text{pg}} \cos\theta_r^p \\ \Delta x_h^{\text{pg}} = x_r^g - x_r^p \\ \Delta y_h^{\text{pg}} = y_r^g - y_r^p \end{cases} \tag{16}$$

where $x_r^g, y_r^g, w_r^g, h_r^g$, and θ_r^g represent the ground-truth bounding boxes. Considering the UF-RPN offset calculation in (14), we can define the ground-truth proposal offsets of FD-Head as follows:

$$\begin{cases} \Delta x_r^{\text{pg}} = \dfrac{\left(x_r^g - x_r^p\right)\cos\theta_r^p + \left(y_r^g - y_r^p\right)\sin\theta_r^p}{\sqrt{\left(w_r^p\right)^2 + \left(h_r^p\right)^2}} \\ \Delta y_r^{\text{pg}} = \dfrac{-\left(x_r^g - x_r^p\right)\sin\theta_r^p + \left(y_r^g - y_r^p\right)\cos\theta_r^p}{\sqrt{\left(w_r^p\right)^2 + \left(h_r^p\right)^2}} \\ \Delta L_a^{\text{pg}} = \dfrac{w_r^g \cos\Delta\theta_r^g + h_r^g \sin\Delta\theta_r^g}{\sqrt{\left(w_r^p\right)^2 + \left(h_r^p\right)^2}} \\ \Delta L_b^{\text{pg}} = \dfrac{w_r^g \cos\Delta\theta_r^g - h_r^g \sin\Delta\theta_r^g}{\sqrt{\left(w_r^p\right)^2 + \left(h_r^p\right)^2}} \\ \Delta L_h^{\text{pg}} = \dfrac{w_r^g |\sin\Delta\theta_r^g| + h_r^g \cos\Delta\theta_r^g}{\sqrt{\left(w_r^p\right)^2 + \left(h_r^p\right)^2}} \end{cases} \tag{17}$$

where $\Delta\theta_r^g$ represents the errors between ground-truth bounding boxes and the rotation proposals. Besides, the final predicted OBBs $(x_r^{\text{rp}}, y_r^{\text{rp}}, L_a^{\text{rp}}, L_b^{\text{rp}}, L_h^{\text{rp}})$ also can be obtained by decoding the rotational proposals and the predict proposal offsets $(\Delta x_r^{\text{rp}}, \Delta y_r^{\text{rp}}, \Delta L_a^{\text{rp}}, \Delta L_b^{\text{rp}}, \Delta L_h^{\text{rp}})$ of the FD-Head regression branch. The specific formulas in the decoding process are calculated as follows:

$$\begin{cases} x_r^{\text{rp}} = x_r^p + \left(\Delta x_r^{\text{rp}} \cos\theta_r^p - \Delta y_r^{\text{rp}} \sin\theta_r^p\right) \sqrt{\left(w_r^p\right)^2 + \left(h_r^p\right)^2} \\ y_r^p = y_r^p + \left(\Delta x_r^{\text{rp}} \sin\theta_r^p + \Delta y_r^{\text{rp}} \cos\theta_r^p\right) \sqrt{\left(w_r^p\right)^2 + \left(h_r^p\right)^2} \\ L_a^{\text{rp}} = \Delta L_a^{\text{rp}} \sqrt{\left(w_r^p\right)^2 + \left(h_r^p\right)^2} \\ L_b^{\text{rp}} = \Delta L_b^{\text{rp}} \sqrt{\left(w_r^p\right)^2 + \left(h_r^p\right)^2} \\ L_h^{\text{rp}} = \Delta L_h^{\text{rp}} \sqrt{\left(w_r^p\right)^2 + \left(h_r^p\right)^2}. \end{cases} \tag{18}$$

By utilizing the five-distance OBB representation in the detection head, the proposed FD-Head can avoid the direct regression of angle parameters and achieve OBB prediction using the fewest parameters without additional processes. As a result, compared with the general detection head, the FD-Head can further improve the accuracy of oriented object detection in remote sensing images.

D. Loss Function

As shown in Fig. 2, the total loss of the proposed method includes the UF-RPN loss and the FD-Head loss, with each part containing classification loss and regression loss. Therefore, the total loss $\mathcal{L}$ is defined as follows:

$$\mathcal{L} = \alpha_1 \mathcal{L}_{\text{rpn}}^{\text{cls}} + \alpha_2 \mathcal{L}_{\text{rpn}}^{\text{loc}} + \alpha_3 \mathcal{L}_{\text{head}}^{\text{cls}} + \alpha_4 \mathcal{L}_{\text{head}}^{\text{loc}} \tag{19}$$

where $\mathcal{L}_{\text{rpn}}^{\text{cls}}$ and $\mathcal{L}_{\text{rpn}}^{\text{loc}}$ are the classification and regression losses of the UF-RPN, respectively. $\mathcal{L}_{\text{head}}^{\text{cls}}$ and $\mathcal{L}_{\text{head}}^{\text{loc}}$ are the classification and regression losses of the FD-Head, respectively. α_1, α_2, α_3, and α_4 are the weighting parameters of different losses, which are both set to 1.0 in the following experiments. More specifically, each loss function in (19) is defined as follows:

$$\begin{aligned} \mathcal{L}_{\text{rpn}}^{\text{cls}} &= \frac{1}{N_1} \sum_{i=1}^{N_1} \mathcal{L}_{\text{cls}}\left(p_i, p_i^*\right) \\ \mathcal{L}_{\text{rpn}}^{\text{loc}} &= \frac{1}{N_1} p_i^* \sum_{i=1}^{N_1} \mathcal{L}_{\text{reg}}\left(\boldsymbol{t_i}, \boldsymbol{t_i^*}\right) \\ \mathcal{L}_{\text{head}}^{\text{cls}} &= \frac{1}{N_2} \sum_{j=1}^{N_2} \mathcal{L}_{\text{cls}}\left(q_j, q_j^*\right) \\ \mathcal{L}_{\text{head}}^{\text{loc}} &= \frac{1}{N_2} p_i^* \sum_{j=1}^{N_2} \mathcal{L}_{\text{reg}}\left(\boldsymbol{s_j}, \boldsymbol{s_j^*}\right) \end{aligned} \tag{20}$$

in which i is the index of the preset horizontal anchors, and N_1 (by default $N_1 = 256$) is the total number of sampling anchors in UF-RPN. p_i and p_i^* represent the predicted probability of UF-RPN classification and the ground-truth labels, respectively. In UF-RPN, the ground-truth labels only contain two cases (foreground and background). $\boldsymbol{t_i}$ $(\Delta x_r^{\text{ap}}, \Delta y_r^{\text{ap}}, \Delta L_a^{\text{ap}}, \Delta L_b^{\text{ap}}, \Delta L_h^{\text{ap}})$ and $\boldsymbol{t_i^*}$ $(\Delta x_r^{\text{ag}}, \Delta y_r^{\text{ag}}, \Delta L_a^{\text{ag}}, \Delta L_b^{\text{ag}}, \Delta L_h^{\text{ag}})$ are the predicted offsets of UF-RPN regression and the ground-truth offsets of ground-truth boxes relative to the horizontal anchors, respectively. Similarly, j is the index of rotational proposals, and N_2 (by default $N_2 = 512$) is the total number of sampling proposals in FD-Head. q_j and q_j^* denote the predicted probability of FD-Head classification and the ground-truth labels, respectively. $\boldsymbol{s_i}$ $(\Delta x_r^{\text{rp}}, \Delta y_r^{\text{rp}}, \Delta L_a^{\text{rp}}, \Delta L_b^{\text{rp}}, \Delta L_h^{\text{rp}})$ and $\boldsymbol{s_i^*}$ $(\Delta x_r^{\text{pg}}, \Delta y_r^{\text{pg}}, \Delta L_a^{\text{pg}}, \Delta L_b^{\text{pg}}, \Delta L_h^{\text{pg}})$ represent the predicted offsets of FD-Head regression and the ground-truth offsets of the ground-truth bounding boxes relative to the rotational proposals, respectively. $\mathcal{L}_{\text{rpn}}^{\text{cls}}$ and $\mathcal{L}_{\text{head}}^{\text{cls}}$ use cross-entropy loss, while $\mathcal{L}_{\text{rpn}}^{\text{loc}}$ and $\mathcal{L}_{\text{head}}^{\text{loc}}$ adopt the SC-loss $\mathcal{L}_{\text{sc}}$. The $\mathcal{L}_{\text{sc}}$ loss incorporates the smooth-L1 loss [7] $(\mathcal{L}_1)$, the center point loss $\mathcal{L}_2$, the loss of auxiliary variable $\mathcal{L}_3$ and the loss of the MECs diameter $\mathcal{L}_4$. The specific formula of $\mathcal{L}_{\text{sc}}$ is defined as follows:

$$\mathcal{L}_{\text{sc}} = \mathcal{L}_1 + \beta_1 \mathcal{L}_2 + \beta_2 \mathcal{L}_3 + \beta_3 \mathcal{L}_4. \tag{21}$$

For the $\mathcal{L}_{\text{sc}}$ of $\mathcal{L}_{\text{rpn}}^{\text{loc}}$, $\mathcal{L}_1$ represents the smooth-L1 loss of the $(\Delta L_a^{\text{ap}}, \Delta L_b^{\text{ap}}, \Delta L_h^{\text{ap}})$ and $(\Delta L_a^{\text{ag}}, \Delta L_b^{\text{ag}}, \Delta L_h^{\text{ag}})$. The other loss in (21) can be expressed as follows:

$$\mathcal{L}_2 = \sqrt{\left(\Delta x_r^{\text{ap}} - \Delta x_r^{\text{ag}}\right)^2 + \left(\Delta y_r^{\text{ap}} - \Delta y_r^{\text{ag}}\right)^2} \tag{22}$$

$$\begin{cases} \mathcal{L}_3 = \left|L_d\left(\boldsymbol{t_i}\right) - L_d\left(\boldsymbol{t_i^*}\right)\right| \\ L_d\left(\boldsymbol{t_i}\right) = \sqrt{\left(\Delta L_h^{\text{ap}}\right)^2 + \left|\left(\Delta L_a^{\text{ap}}\right)^2 - \left(\Delta L_b^{\text{ap}}\right)^2\right|} \\ L_d\left(\boldsymbol{t_i^*}\right) = \sqrt{\left(\Delta L_h^{\text{ag}}\right)^2 + \left|\left(\Delta L_a^{\text{ag}}\right)^2 - \left(\Delta L_b^{\text{ag}}\right)^2\right|} \end{cases} \tag{23}$$

$$\begin{cases} \mathcal{L}_4 = \left| D(\boldsymbol{t}_i) - D\left(\boldsymbol{t}_i^*\right) \right| \\ D(\boldsymbol{t}_i) = \sqrt{\left(\Delta L_a^{\text{ap}}\right)^2 + \left(\Delta L_b^{\text{ap}}\right)^2 + \left(\Delta L_h^{\text{ap}}\right)^2 + \left(L_d(t_i)\right)^2} \\ D\left(\boldsymbol{t}_i^*\right) = \sqrt{\left(\Delta L_a^{\text{ag}}\right)^2 + \left(\Delta L_b^{\text{ag}}\right)^2 + \left(\Delta L_h^{\text{ag}}\right)^2 + \left(L_d\left(t_i^*\right)\right)^2}. \end{cases} \tag{24}$$

In the localization loss $\mathcal{L}_{\text{sc}}$, the $\mathcal{L}_2$, $\mathcal{L}_3$, and $\mathcal{L}_4$ losses can be seen as the SC-Loss ($\beta_1 = 1.0$, $\beta_2 = 0.5$, and $\beta_3 = 0.1$), helping the network achieve faster convergence and improve the accuracy of object detection. Besides, $\mathcal{L}_{\text{head}}^{\text{loc}}$ adopts the similar calculation with the $\mathcal{L}_{\text{rpn}}^{\text{loc}}$. In Section IV-D, we will give the ablation experiments to verify the superiority of SC-Loss compared with the commonly used smooth-L1 loss.

IV. Experiments

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

A. Datasets

In the following experiments, the oriented object detection methods are evaluated on four publicly available datasets: the high-resolution ship collection 2016 (HRSC2016 dataset) [55], DIOR-R dataset [56], the large-scale dataset for Object detection in aerial images (DOTA-v1.0 dataset) [57] and DOTA-v2.0 dataset [58].

The HRSC2016 dataset is a classic ship dataset with OBB annotations in the remote sensing field. Overall, this dataset contains 1061 images and 2976 instances, which are divided into three parts: a training set (436 images), a validation set (181 images), and a testing set (444 images). The sizes of images in the HRSC2016 dataset range from 300 × 300 to 1500 × 900. In the experiments, the training and validation sets are used for training, and the testing set is only used for testing. The detection results are measured by the mean average precision (mAP_{50}) under both the PASCAL VOC07 and VOC12 metrics [59], [60], [61], [62].

The DIOR-R dataset is an optical remote sensing dataset containing a total of 192 518 instances and 23 463 aerial images with oriented annotations. The size of images in this dataset is 800 × 800, and the spatial resolution ranges from 0.5 to 30 m. In addition, this dataset consists of 20 categories of objects, including airplane (APL), airport (APO), baseball field (BF), basketball court (BC), bridge (BR), chimney (CH), expressway service area (ESA), expressway toll station (ETS), dam (DAM), golf field (GF), ground track field (GTF), harbor (HA), overpass (OP), ship (SH), stadium (STA), storage tank (STO), tennis court (TC), train station (TS), vehicle (VE), and windmill (WM). In the experiment, the training and validation sets (11 725 images) are used to train the model, and the testing set (11 738 images) is used to evaluate the model and compare the detection results of different methods.

The DOTA-v1.0 dataset is a large-scale remote sensing dataset with image sizes ranging from 800 × 800 to 4000 × 4000. The DOTA-v1.0 dataset contains 15 classes of objects: 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), ST, soccer-ball field (SBF), roundabout (RA), harbor (HA), swimming pool (SP), and helicopter (HC). This dataset includes 2806 images and 188 283 instances with bounding box annotations of four vertex coordinates. In the experiments, all images are divided into two parts: the training images (1869 images), which contain the training and validation sets, and the testing images (937 images), which contain only the testing set. The mAP of detection results is evaluated by submitting the test results to the DOTA-v1.0 server.

Besides, compared with the DOTA-v1.0 dataset, the DOTA-v2.0 dataset incorporates more Google Earth, GF-2 Satellite, and aerial images for object detection. It includes a training set of 1830 images, a validation set of 593 images, and a testing set of 2792 images. The DOTA-v2.0 dataset includes 18 categories. In addition to all the classes from DOTA-v1.0, it further introduces the container crane (CC), airport (AP), and helipad (HP). The mAP of the detection results is also evaluated by submitting the test results to the DOTA-v2.0 server.

B. Implementation Details

The following experiments are based on computer hardware configured with the Intel i7-10700K CPU, 32-GB RAM (8 GB × 4) and two NVIDIA RTX 2080Ti 11-GB GPUs. All experiments are performed in an environment running the Ubuntu 18.04 operating system and the PyTorch 1.7.1 software framework. The network training process is accelerated via CUDA 10.1 with Python 3.7. The proposed methods are developed based on the public MMDetection platform [63]. In the training process, two RTX 2080Ti GPUs are used with a batch size of 2, while the inference process is only tested on a single RTX 2080Ti GPU for achieving a fair comparison of different methods. ResNet50 and ResNet101 with pretrained parameters on ImageNet are used as the backbone in the training, facilitating quick network convergence. The overall network adopts the simple SGD optimizer, with momentum and weight decay set to 0.9 and 0.0001, respectively. In addition, horizontal and vertical flipping are used as data augmentation techniques to achieve more accurate object detection results in remote sensing images.

For the HRSC2016 dataset with single-scale training and testing, the input images are resized to 1333 × 800 while maintaining the aspect ratio for oriented object detection. More specifically, the images are resized proportionally so that their shorter sides are 800 pixels. This makes the longer sides of the resized images equal to or less than 1333 pixels. In addition, for the images whose longer sides are less than 1333 pixels, padding is applied to make their longer sides reach this size. The total number of training epochs is set to 48, with the learning rate initialized at 0.01 and reduced by a factor of 10 at epochs 36 and 44. For multiscale training and testing on the HRSC2016 dataset, the original images are first resized to three different scales (0.5, 1.0, and 1.5) and cropped into 800 × 800 patches with a stride of 524 pixels and a pixel overlap of 200 pixels between adjacent patches. The cropped images are then input to the network for training and testing. The total number of multiscale training epochs is set to 24, with the learning rate initialized at 0.01 and divided by a factor of 10

at epochs 16 and 22. In addition, the mAP_{50} is obtained by setting the IoU threshold to 0.5 for evaluating the performance of different algorithms.

For the DIOR-R dataset, the input image size uses the original size 800 × 800, and maintains the aspect ratios of the original images. The total number of training epochs is also set to 48, with the learning rate initialized at 0.01 and reduced by a factor of 10 at epochs 36 and 44. The mAP_{50} are used to evaluate the oriented object detection performance on the DIOR-R dataset for different methods. In addition, mAP_{75} and $mAP_{50:95}$ are also utilized to compare the performance of object detection under a higher IoU threshold.

For the DOTA-v1.0 dataset, the original images are cropped into 1024 × 1024 pixels due to the large image scales. The stride of cropping and the pixel overlap between two adjacent patches are set to 824 and 200 pixels, respectively. For multiscale training and testing, the original images are first resized to three scales (0.5, 1.0, and 1.5) and cropped into 1024 × 1024 pixels with a stride of 524 pixels. The total number of training epochs is set to 12. The learning rate is set to 0.01 and is divided by 10 at epochs 8 and 11. The final NMS threshold is set to 0.1 for obtaining the bounding boxes and categories of objects. The mAP is obtained by submitting the detection results to the DOTA-v1.0 server, ensuring a fair comparison among different detection methods.x

Besides, for the DOTA-v2.0 dataset, the original images are also cropped into 1024 × 1024 pixels due to the large image scales. The stride of cropping and the pixel overlap between two adjacent patches are set to 824 and 200 pixels, respectively. The total number of training epochs is set to 24, and the learning rate is initialized at 0.01 and is divided by 10 at epochs 16 and 22. The final NMS threshold is set to 0.1 for obtaining the bounding boxes and categories of objects. The mAP_{50} is obtained by submitting the detection results to the DOTA-v2.0 server for comparing the performance of different methods.

C. Comparison With SOTA Methods

In this section, our method is compared with other SOTA methods. In these methods, some methods design a new loss function to evaluate the accuracy of predicted bounding boxes, such as ARS-DETR [54], GF-CSL [49], KLD [44], P2P [64], and PIoU [65]. Some other methods propose new approach to obtain the rotational proposals from horizontal anchors, for example, Gliding Vertex [13], Oriented R-CNN [14], and QPDet [15], while the other methods adopt some efficient strategies (e.g., label assignment, feature alignment, and DCN [66]) to obtain more accurate results of the oriented object detection, such as the DAL [67], DODet [68] and S^2ANet [69].

1) Results on the HRSC2016 Dataset: We compare the mAP_{50} results of our method on the HRSC2016 dataset with other advanced oriented object detectors in Table II. The symbol "‡" denotes multiscale training and testing to obtain the quantitative results. The $mAP_{50}(07)$ and $mAP_{50}P(12)$ denote the quantitative results under PASCAL VOC07 and PASCAL VOC12 metrics, respectively. All the quantitative results of compared detectors are sourced from the References of the "Methods" column in Table II.

TABLE II
COMPARISON OF OUR METHOD WITH SOME ADVANCED METHODS ON THE HRSC2016 DATASET

Methods	Backbones	$mAP_{50}(07)$	$mAP_{50}(12)$
S^2ANet [69]	R-101-FPN	90.17	95.01
TIR-Net [5]	R-50-FPN	90.20	-
ABFL [17]	R-50-FPN	89.98	-
GF-CSL [49]	R-50-FPN	89.80	96.28
Faster R-CNN-O [8]	R-50-FPN	83.93	-
Gliding Vertex [13]	R-101-FPN	88.20	-
Oriented R-CNN [14]	R-50-FPN	90.40	96.50
QPDet [15]	R-50-FPN	90.47	96.60
UFDet	R-50-FPN	90.47	97.50
UFDet	R-101-FPN	90.54	97.86
UFDet‡	R-50-FPN	90.48	98.25
UFDet‡	R-101-FPN	90.58	98.53

Fig. 7. Visualization of some detection results on the HRSC2016 dataset based on our method by using the backbone ResNet50 and FPN.

As shown in Table II, with single-scale training and testing, our method can achieve 90.47% (VOC 07 metric) and 97.50% mAP_{50} (VOC 12 metric) using the ResNet50 as backbone, respectively. Moreover, using the ResNet101 as backbone, our method achieves 90.54% (VOC 07 metric) and 97.86% mAP_{50} (VOC 12 metric), which shows that our method achieves higher detection accuracy compared with other SOTA methods. Specifically, compared with the typical S^2ANet and Gliding Vertex, our method improves by 2.85% mAP_{50} (VOC 12 metric) and 2.34% mAP_{50} (VOC 07 metric) under the same single-scale training and testing conditions. By utilizing multiscale training and testing, our method can obtain 90.58% mAP_{50} and 98.53% mAP_{50} under PASCAL VOC 07 and PASCAL VOC 12 metrics based on the ResNet101. Besides, some detection results of our method on the HRSC2016 dataset are shown in Fig. 7, which also illustrates that our method can achieve accurate ship detection in various scenes. For example, it can detect ships accurately in complex background and foggy scenes (first and second rows of Fig. 7).

2) Results on the DIOR-R Dataset: We compare the proposed method on the DIOR-R dataset with other SOTA methods in Table III. ARS-DETR* represents another implementation of the original method in Table III, which is retrained based on our software environment with the same data augmentation techniques as our method.

As shown in Table III, our method achieves 67.90% in mAP_{50}, 44.10% in mAP_{75} and 42.52% in $mAP_{50:95}$, which are competitive compared with other SOTA methods. In Table III,

TABLE III

COMPARISON OF OUR METHOD WITH SOME ADVANCED METHODS ON THE DIOR-R DATASET. THE VALUES WITH BOLD FONT REPRESENT THE BEST MAP RESULTS

Methods	Faster R-CNN-O [8]	S^2ANet [69]	R3Det [41]	Gliding Vertex [13]	KFIoU [45]	SASM [70]	GWD [43]	KLD [44]	FCOSR [71]	Oriented Rep [53]	RoI Trans [50]	ARS-DETR* [54]	UFDet
APL	62.89	67.98	62.55	62.67	58.03	64.78	69.68	66.52	63.07	67.80	63.10	63.57	72.15
APO	41.38	44.44	43.44	38.56	45.41	49.90	28.83	46.80	40.22	48.01	47.19	43.91	41.93
BF	71.83	71.63	71.72	71.94	69.52	74.94	74.32	71.76	71.89	77.02	71.97	72.54	79.25
BC	81.00	81.39	81.48	81.20	81.55	80.38	81.49	81.43	81.36	85.37	81.20	83.05	89.94
BR	38.01	42.66	36.49	37.73	38.82	34.52	29.62	40.81	39.67	38.55	42.49	36.24	47.61
CH	72.46	72.72	72.63	72.48	73.36	69.21	72.67	78.25	72.51	78.45	72.59	73.96	72.65
ESA	77.73	79.03	79.50	78.62	78.08	76.28	76.45	79.23	79.19	81.13	79.43	77.92	85.83
ETS	67.52	70.40	64.41	69.04	66.41	61.37	63.14	66.63	69.45	72.06	71.08	69.05	74.69
DAM	28.61	27.08	27.02	22.81	25.23	31.66	27.13	29.01	26.00	33.67	30.37	32.25	35.00
GF	74.58	75.56	77.36	77.89	79.24	72.22	77.19	78.68	77.93	76.00	78.15	73.03	77.56
GTF	77.04	81.02	77.17	82.13	78.25	77.81	78.94	80.19	82.28	79.89	82.45	77.40	84.32
HA	40.66	43.41	40.53	46.22	44.67	44.69	39.11	44.88	46.91	45.72	48.77	33.23	47.33
OP	53.92	56.45	53.33	54.76	54.45	52.08	42.18	57.23	53.90	54.27	57.82	52.55	58.94
SH	79.41	81.12	79.66	81.03	80.78	83.64	79.10	80.91	81.03	85.13	81.24	81.26	81.28
STA	66.33	68.00	69.22	74.88	68.40	62.83	70.41	74.17	75.77	76.04	76.82	72.05	75.65
STO	67.57	70.03	61.10	62.54	64.52	63.91	58.69	68.02	62.54	65.27	62.48	69.63	70.78
TC	79.88	87.07	81.54	81.41	81.49	80.79	81.52	81.48	81.42	85.38	81.45	82.93	81.51
TS	48.10	53.88	52.18	54.25	51.64	56.54	47.78	54.63	54.50	59.76	57.56	54.81	58.92
VE	46.22	51.12	43.57	43.22	46.03	43.58	44.47	47.80	43.17	48.02	43.80	46.58	50.07
WM	64.79	65.31	64.13	65.13	59.50	63.14	62.63	64.41	65.73	68.92	66.32	70.26	72.53
mAP_{50}	62.00	64.50	61.91	62.91	62.29	62.21	60.31	64.63	63.41	66.31	64.81	63.31	**67.90**
mAP_{75}	36.10	38.24	38.40	40.00	40.20	40.40	40.90	41.60	41.80	44.36	**44.59**	41.72	44.10
$mAP_{50:95}$	37.61	38.02	37.84	38.34	38.52	39.51	39.70	40.34	39.72	**42.81**	42.27	39.73	42.52
GFLOPs	252.70	257.43	412.98	296.21	266.22	237.14	266.22	266.22	296.19	237.14	352.27	396.20	296.28
Parameters	31.93M	38.62M	42.02M	41.15M	36.52M	36.61M	36.52M	35.52M	36.06M	36.61M	55.16M	41.15M	41.14M
Training time	5.91h	7.16h	8.69h	6.61h	7.59h	9.11h	6.76h	7.13h	6.45h	12.88h	7.43h	27.69h	9.35h
FPS	16.12	19.81	17.22	23.56	9.76	18.84	26.23	23.71	23.62	22.24	19.36	16.14	21.52

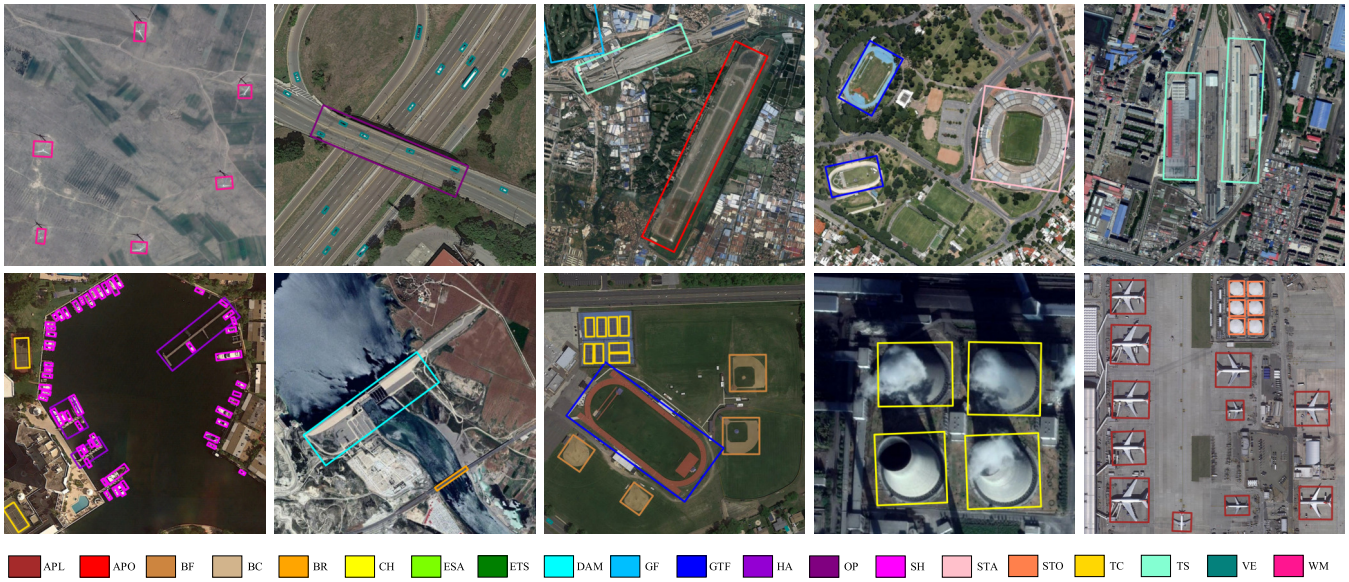

Fig. 8. Visualization of some detection results on the DIOR-R dataset based on our method by using the backbone ResNet50 and FPN. The final bounding boxes of objects need NMS with IoU greater than 0.1 and scores greater than 0.05.

the mAP_{75} of our method is slightly lower than that of ROI Trans, mainly because the ROI Trans employs a more complex network with an RROI Learner module. The $mAP_{50:95}$ of our method is lower than that of Oriented Rep, since Oriented Rep uses a more complex approach (nine points) to represent bounding boxes.

We also show some detection results of our method on the DIOR-R dataset in Fig. 8, which also illustrates that our method can achieve accurate object detection for different categories of objects. In summary, our method can achieve accurate object detection for the DIOR-R dataset in remote sensing images.

TABLE IV
COMPARISON OF OUR METHOD WITH SOME ADVANCED METHODS ON THE DOTA-V1.0 DATASET

Methods	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP_{50}
single-scale:																	
RetinaNet-O [11]	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 [72]	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
PIoU [65]	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 [73]	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 [67]	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
R3Det [41]	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
KLD [44]	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 [64]	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 [70]	R-50-FPN	86.42	78.97	52.47	69.84	77.30	75.99	86.72	90.89	82.63	85.66	60.13	68.25	73.98	72.22	62.37	74.92
S^2ANet [69]	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
Faster R-CNN-O [8]	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
RoI Trans [50]	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
Mask OBB [74]	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 [13]	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
ReDet [47]	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 R-CNN [14]	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
DODet [68]	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
Oriented Rep [53]	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
ARS-DETR [54]	R-50-FPN	86.97	75.56	48.32	69.20	77.92	77.94	87.69	90.50	77.31	82.86	60.28	64.58	74.88	71.76	66.62	74.16
AOPG [52]	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
QPDet [15]	R-50-FPN	89.55	83.66	54.06	73.93	78.93	83.08	88.29	90.89	86.60	84.80	62.03	65.55	74.16	70.09	58.16	76.24
multi-scale:																	
TIR-Net‡ [5]	R-50-FPN	89.56	86.99	58.63	82.60	80.52	85.64	88.90	90.88	86.86	88.20	72.67	70.09	78.61	80.81	68.48	80.63
S^2ANet‡ [69]	R-50-FPN	88.89	83.60	57.74	81.95	79.94	83.19	89.11	90.78	84.87	87.81	70.30	68.25	78.30	77.01	69.58	79.42
KLD‡ [44]	R-50-FPN	88.91	85.23	53.64	81.23	78.20	76.99	84.58	89.50	86.84	86.38	71.69	68.06	75.95	72.23	75.42	78.32
GWD‡ [43]	R-50-FPN	89.06	84.32	55.33	77.53	76.95	70.28	83.95	89.75	84.51	86.06	73.47	67.77	72.60	75.76	74.17	77.43
TIOE‡ [46]	R-50-FPN	89.76	85.23	56.32	76.17	80.17	85.58	88.41	90.81	85.93	87.27	68.32	70.32	68.93	78.33	68.87	78.69
FPNFormer‡ [48]	R-50-FPN	89.10	82.41	56.15	81.24	82.05	85.16	88.74	90.89	85.78	87.07	68.93	68.47	77.93	78.65	68.35	79.39
GF-CSL‡ [49]	R-50-FPN	89.77	86.15	49.85	73.01	78.56	81.77	87.95	90.84	87.91	86.23	63.04	66.24	77.14	79.27	65.32	77.54
ABFL‡ [17]	R-50-FPN	88.85	83.02	56.03	70.08	81.46	84.47	88.11	90.84	84.68	86.58	67.97	73.19	75.55	82.48	69.86	78.88
AOPG‡ [52]	R-50-FPN	89.88	85.57	60.90	81.51	78.70	85.29	88.85	90.89	87.60	87.65	71.66	68.69	82.31	77.32	73.10	80.66
ReDet‡ [47]	ReR50-FPN	88.81	82.48	60.83	80.82	78.34	86.06	88.31	90.87	88.77	87.03	68.65	66.90	79.26	79.71	74.67	80.10
UFDet	R-50-FPN	89.40	83.63	53.03	72.81	78.98	81.44	87.91	90.90	87.00	84.40	64.21	69.10	70.87	71.33	63.61	**76.57**
UFDet	R-101-FPN	89.44	84.34	54.55	73.39	78.58	82.50	87.78	90.89	85.78	85.78	64.39	64.77	73.06	70.21	57.82	76.22
UFDet‡	R-50-FPN	89.95	83.78	59.46	80.99	79.69	85.16	88.62	90.89	86.16	88.38	71.38	70.76	80.10	78.16	78.08	**80.77**
UFDet‡	R-101-FPN	90.04	83.15	60.89	80.78	79.28	84.90	88.70	90.85	86.77	87.85	71.85	71.13	82.11	79.09	69.47	80.46

3) Results on the DOTA-v1.0 Dataset: We compare the proposed method on the DOTA-v1.0 dataset with other SOTA methods in Table IV, which includes 15 AP_{50} (average precision) for each categories of objects and the mAP_{50} for the overall detection results. In Table IV, the symbol '‡' represents multiscale training and testing to obtain the quantitative results. The value with bold font represents the best mAP result for single-scale testing, and the value with bold font and underline represents the best mAP results for multiscale testing.

As shown in Table IV, our method achieves 76.57% mAP_{50} under single-scale training and testing, and 80.77% mAP_{50} under multiscale training and testing, which can outperform the most SOTA methods. By using the backbone ResNet101, our method can achieve 76.22% mAP_{50} under single-scale training and testing, and 80.46% mAP_{50} under multiscale training and testing. More specifically, our method surpasses the advanced Gliding Vertex, AOPG, and Oriented R-CNN under the same backbone with the mAP_{50} gains of 1.20%, 0.83%, and 0.70%, respectively. Some detection results of our method on the DOTA-v1.0 dataset are visualized in Fig. 9, which demonstrate that our method can achieve accurate object detection for different categories of objects in remote sensing images. For example, it accurately detects LVs with different orientations in row #1, column #5, SPs in row #1, column #1, and densely packed STs in row #2, column #5 of Fig. 9.

In addition, we show some detection results for the same samples on the DOTA-v1.0 dataset using some advanced methods in Fig. 10. As shown in Fig. 10, compared with Faster R-CNN-O, QPDet, S^2ANet, and Oriented R-CNN, our method achieves more accurate object detection in remote sensing images. This can be seen from the detection results for densely packed objects (SVs in row #1, SHs in row #3, LVs in row #4, and STs in row #6) and slightly blurry objects (BD and SVs in row #1, LVs in row #4, and HAs in row #5). The localization errors of our method are also lower than those of Faster R-CNN-O, QPDet, S^2ANet, and Oriented R-CNN. Therefore, our method reduces both missed detections (row #4 and row #6) and false detections (row #1 and row #2), improving object detection performance compared with other methods.

4) Results on the DOTA-v2.0 Dataset: We compare the proposed method on the DOTA-v2.0 dataset with other SOTA methods in Table V.

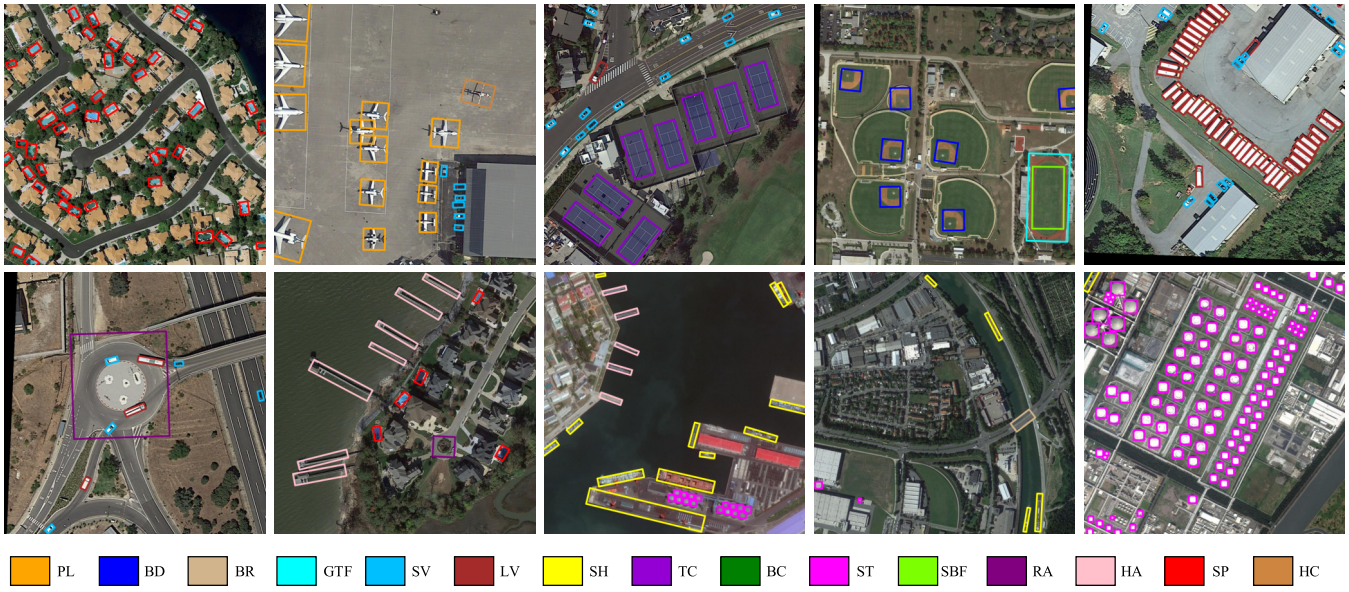

Fig. 9. Visualization of some detection results on the DOTA-v1.0 dataset based on our method by using the backbone ResNet50 and FPN. The final bounding boxes of objects need NMS with an IoU greater than 0.1 and scores greater than 0.05.

TABLE V

COMPARISON OF OUR METHOD WITH SOME ADVANCED METHODS ON THE DOTA-V2.0 DATASET. THE VALUE WITH BOLD FONT REPRESENTS THE BEST MAP RESULT

Methods	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	CC	AP	HP	mAP_{50}	FPS
RetinaNet-O [11]	R-50-FPN	70.63	47.26	39.12	55.02	38.10	40.52	47.16	77.74	56.86	52.12	37.22	51.75	44.15	53.19	51.06	6.58	64.28	7.45	46.68	17.24
Faster R-CNN-O [8]	R-50-FPN	71.61	47.20	39.28	58.70	35.55	48.88	51.51	78.97	58.36	58.55	36.11	51.73	43.57	55.33	57.07	3.51	52.94	2.79	47.31	16.52
Mask R-CNN-O [9]	R-50-FPN	76.20	49.91	41.61	60.00	41.08	50.77	56.24	78.01	55.85	57.48	36.62	51.67	47.39	55.79	59.06	3.64	60.26	8.95	49.47	7.01
RoI Trans [50]	R-50-FPN	71.81	48.39	45.88	64.02	42.09	54.39	59.92	82.70	63.29	58.71	41.04	52.82	53.32	56.18	57.94	25.71	63.72	8.70	52.81	13.28
S^2ANet [69]	R-50-FPN	77.83	54.57	44.22	62.99	47.38	50.87	57.96	79.72	59.25	65.36	39.26	53.35	45.72	52.45	34.08	0.67	67.21	5.03	49.88	16.88
ReDet [47]	Re-50-FPN	79.61	54.02	50.36	62.29	43.64	50.59	60.94	79.76	58.78	59.97	45.23	54.79	55.96	55.21	55.86	21.26	69.36	11.31	53.83	10.32
GWD [43]	R-50-FPN	72.64	51.84	36.22	59.88	40.03	40.25	46.92	77.49	56.31	58.95	37.83	53.90	39.70	48.28	36.17	0.01	68.15	0.22	45.82	14.86
KLD [44]	R-50-FPN	73.50	47.22	38.58	60.96	39.57	43.46	47.24	76.31	56.60	57.72	38.96	52.64	40.74	47.61	41.19	0.41	68.54	10.59	46.77	14.82
Oriented R-CNN [14]	R-50-FPN	78.65	51.80	47.15	65.78	43.35	58.29	60.89	82.83	63.51	59.50	43.40	55.79	52.90	56.18	54.13	27.55	66.24	5.22	54.06	16.04
OAN [75]	R-50-FPN	71.90	50.48	43.55	65.76	42.51	54.90	60.35	79.09	63.45	60.03	46.59	52.91	54.30	57.25	60.33	23.59	66.39	9.72	53.51	16.42
UFDet	R-50-FPN	78.63	51.80	46.53	65.47	43.40	58.85	60.96	82.98	63.59	59.53	45.65	53.94	53.38	57.92	55.78	29.21	69.67	12.96	**55.02**	16.16

As shown in Table V, our method can achieve 55.02% mAP_{50} under the ResNet50 backbone with FPN, which also illustrates the superiority of our method compared with other SOTA methods. Moreover, the detection speed of our method is 16.1 FPS, which is also competitive with these methods, achieving higher accuracy in object detection. Some detection results based on the DOTA-v2.0 dataset are visualized in Fig. 11.

In summary, compared with other SOTA methods, our method achieves higher performance of object detection on the HRSC2016, DIOR-R, DOTA-v1.0, and DOTA-v2.0 datasets. For example, our method obtains higher detection accuracy for harbors and planes, as shown by the quantitative results ranking in the top three in Tables III–V, and the visualization results in Figs. 8–11. Besides, the proposed method does not perform sufficiently well on some samples (as shown in Fig. 12), which represents a future direction for this work. These cases are primarily due to the difficulty of regressing objects with extremely small scales or large aspect ratios from anchors with fixed aspect ratios.

D. Ablation Studies

1) Different Settings of Our Method: We conduct a series of ablation experiments to verify the superiority of each module in our method on the DOTA-v1.0 dataset. The quantitative results are listed in Table VI. In Table VI, the symbol "✓" denotes using the module. On the contrary, the symbol "✗" represents not using the module.

TABLE VI

COMPARISON OF DIFFERENT SETTINGS OF OUR METHOD ON THE DOTA-V1.0 DATASET

Modules	Baseline	Different Settings of our method				
UF-RPN	✗	✓	✗	✓	✗	✓
FD-Head	✗	✗	✓	✗	✓	✓
SC-Loss	✗	✗	✗	✓	✓	✓
mAP_{50}/(%)	69.05	75.72	72.70	76.11	73.82	76.57

As shown in Table VI, our method is based on the two-stage Faster R-CNN-O (baseline method), which mainly constructs two modules: UF-RPN, FD-Head. The SC-Loss in our method is primarily designed to accelerate the network's convergence. Specifically, compared with the baseline method, the UF-RPN and FD-Head alone achieve mAP_{50} improvements of 6.67% and 3.65% based on the unified five-distance OBB representation, respectively. In addition, our method achieves 76.57% mAP_{50} with a 7.52% mAP_{50} improvement after the UF-RPN, FD-Head, and SC-Loss were applied, which further verifies the effectiveness of our method.

2) Effectiveness of the UF-RPN: As shown in Table VII, we compare different RPN approaches to verify the superiority of

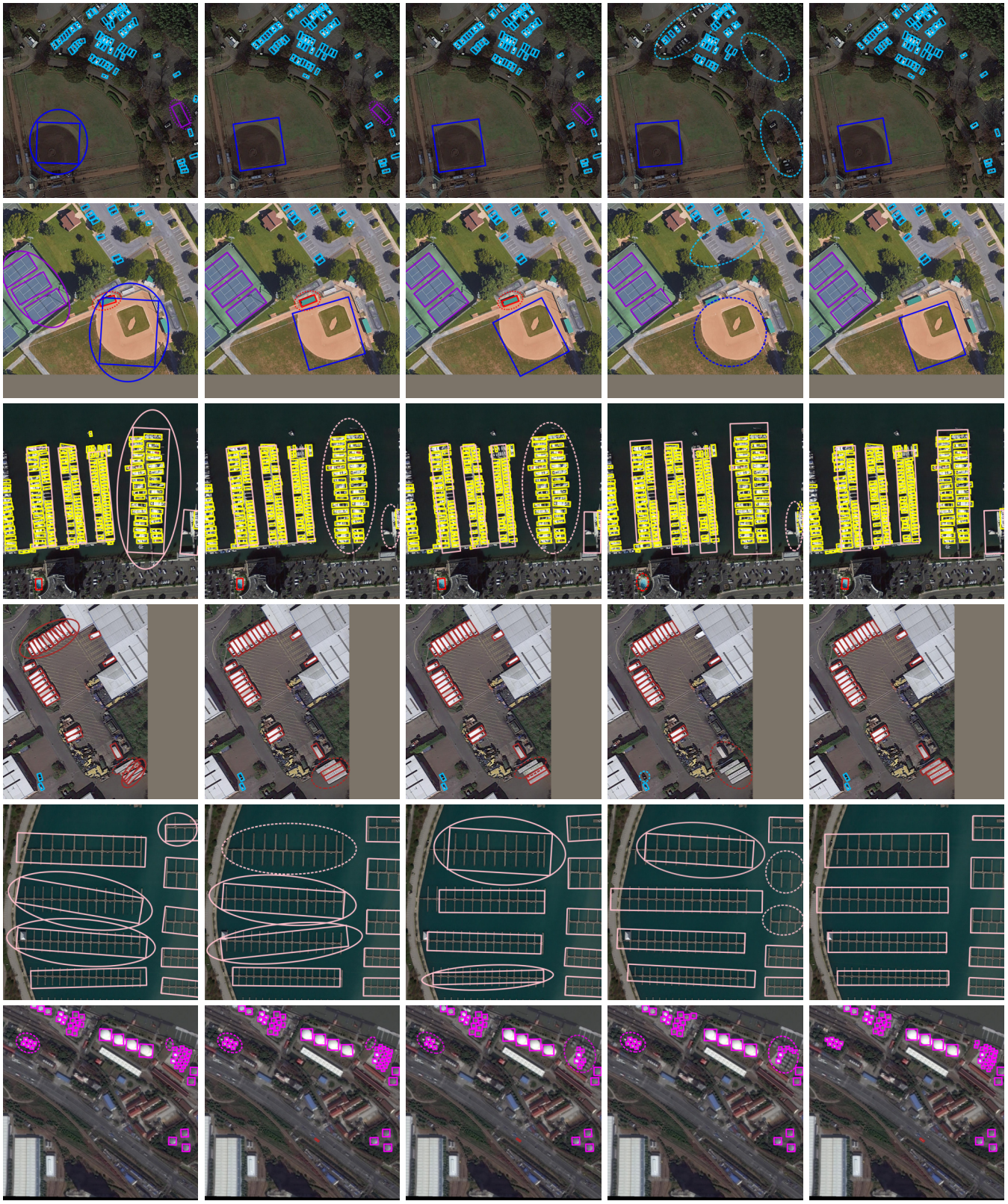

Fig. 10. Detection results of some samples on the DOTA-v1.0 dataset using different methods. The dashed ellipses and the dotted ellipses mark the missed and the false detection results, respectively. The solid-line ellipses denote the detection results with large localization errors. The different color boxes and ellipses are the detection results of different category objects, which are similar to the bounding boxes in Fig. 9. (a) Faster R-CNN-O. (b) Oriented R-CNN. (c) QPDet. (d) S^2ANet. (e) Our method.

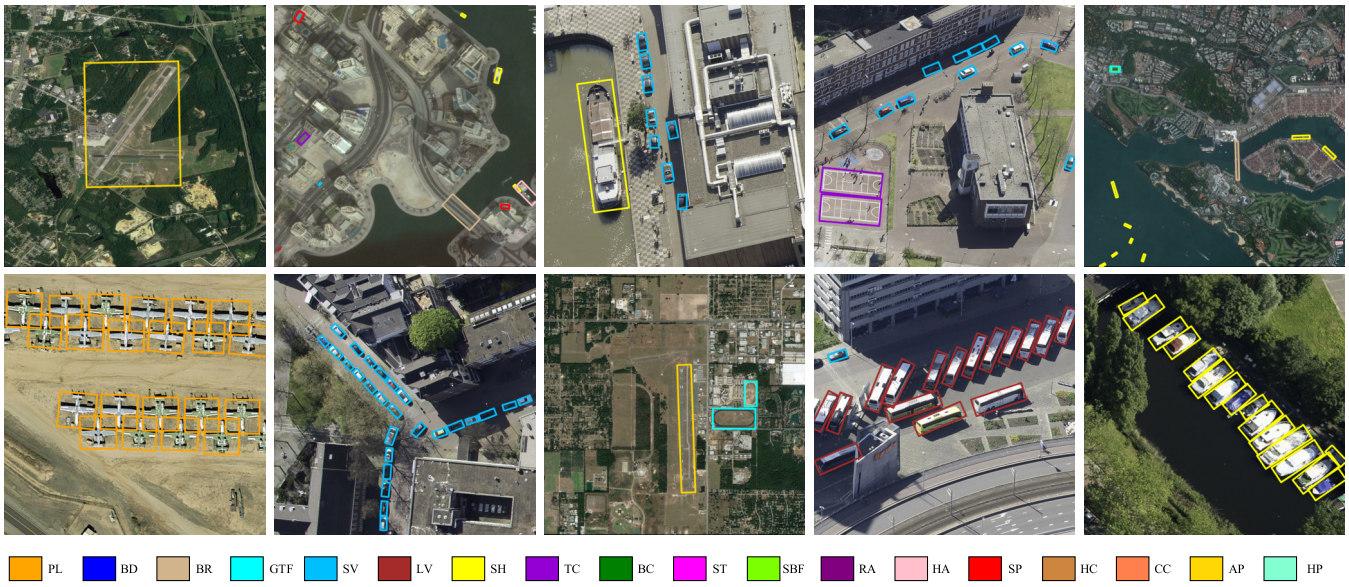

Fig. 11. Visualization of some detection results on the DOTA-v2.0 dataset based on our method by using the backbone ResNet50 and FPN. The final bounding boxes of objects need NMS with IoU greater than 0.1 and scores greater than 0.05.

Fig. 12. Some samples that our method cannot perform well.

TABLE VII

COMPARISON OF DIFFERENT RPNS FOR ACHIEVING ORIENTED OBJECT DETECTION

FPN approaches	mAP_{50}/(%)	FPS	GFLOPs	Parameters
Faster R-CNN-O	69.05	16.52	449.69	41.14M
Gliding Vertex	72.79	14.81	450.22	41.27M
Oriented R-CNN	75.87	16.03	449.96	41.14M
QPDet^{*1}	75.32	16.04	449.83	41.14M
UF-RPN	76.11	16.12	449.83	41.14M

UF-RPN on the DOTA-v1.0 dataset. In Table VII, all methods utilize the same network and the general detection head by directly regressing the general OBBs. QPDet^{*1} denotes another implementation of the original method that uses the general OBB representation in the detection head and was retrained in our software environment.

As shown in Table VII, all these methods utilize different OBB representations in the RPN with the same network structures to show the superiority of UF-RPN. Specifically, compared with the RPN approaches in Faster R-CNN-O, Gliding Vertex, Oriented R-CNN and QPDet^{*1}, the proposed UF-RPN can achieve 76.11% mAP_{50}, increasing 7.06%, 3.32%, 0.24%, and 0.79% mAP_{50} on the DOTA-v1.0 dataset, respectively. Compared with RPN approaches in Faster R-CNN-O and QPDet^{*1}, the UF-RPN can gain significant improvement in mAP_{50} with a small increase or no change in GFLOPs. Compared with RPN approaches in Gliding Vertex and Oriented R-CNN, the UF-RPN can achieve higher performance of object detection with a decrease of GFLOPs and a little increase of detection speed (FPS). In summary, the proposed UF-RPN can generate high-quality rotational proposals and achieve higher mAP detection results compared with other RPN approaches.

TABLE VIII

COMPARISON OF DIFFERENT REPRESENTATIONS IN RPNS AND DETECTION HEADS

Methods	First stage representation	Second stage representation	mAP_{50}/(%)	mAP_{75}/(%)	$mAP_{50:95}$/(%)
Faster R-CNN-O	(x_h, y_h, w_h, h_h)	$(x_r, y_r, w_r, h_r, \theta_r)$	69.60	24.10	33.46
Faster R-CNN-O^{*1}	$(x_r, y_r, w_r, h_r, \theta_r)$	$(x_r, y_r, w_r, h_r, \theta_r)$	72.48	41.09	41.61
Faster R-CNN-O^{*2}	(x_h, y_h, w_h, h_h)	$(x_r, y_r, L_a, L_b, L_h)$	73.82	44.24	43.01
Gliding Vertex	$(x_h, y_h, w_h, h_h, \frac{\Delta a}{w_h}, \frac{\Delta a}{h_h}, \frac{\Delta b}{w_h}, \frac{\Delta b}{h_h})$	$(x_r, y_r, w_r, h_r, \theta_r)$	72.79	41.73	41.43
Gliding Vertex*	$(x_h, y_h, w_h, h_h, \frac{\Delta a}{w_h}, \frac{\Delta a}{h_h}, \frac{\Delta b}{w_h}, \frac{\Delta b}{h_h})$	$(x_r, y_r, L_a, L_b, L_h)$	73.55	46.04	44.52
QPDet	$(x_r, y_r, R, \Delta\alpha, \Delta\beta)$	$(x_r, y_r, R, \Delta\alpha, \Delta\beta)$	76.04	49.61	46.13
QPDet^{*1}	$(x_r, y_r, R, \Delta\alpha, \Delta\beta)$	$(x_r, y_r, w_r, h_r, \theta_r)$	75.32	49.57	46.16
QPDet^{*2}	$(x_r, y_r, R, \Delta\alpha, \Delta\beta)$	$(x_r, y_r, L_a, L_b, L_h)$	76.07	49.58	46.24
Oriented R-CNN	$(x_h, y_h, w_h, h_h, \Delta a, \Delta b)$	$(x_r, y_r, w_r, h_r, \theta_r)$	75.89	50.36	46.40
Oriented R-CNN^{*1}	$(x_h, y_h, w_h, h_h, \Delta a, \Delta b)$	$(x_h, y_h, w_h, h_h, \Delta a, \Delta b)$	76.03	49.56	46.11
Oriented R-CNN^{*2}	$(x_h, y_h, w_h, h_h, \Delta a, \Delta b)$	$(x_r, y_r, L_a, L_b, L_h)$	76.12	49.97	46.36
Only UF-RPN	$(x_r, y_r, L_a, L_b, L_h)$	$(x_r, y_r, w_r, h_r, \theta_r)$	76.11	50.23	46.46
UFDet	$(x_r, y_r, L_a, L_b, L_h)$	$(x_r, y_r, L_a, L_b, L_h)$	76.57	49.18	46.54

3) Effectiveness of the FD-Head: In Table VIII, we use different OBB representations in two-stage network to verify the superiority of the FD-Head based on the DOTA-v1.0 dataset. All methods were retrained to obtain mAP_{50}, mAP_{75}, and $mAP_{50:95}$ results, which leads to the mAP_{50} difference between Table VIII and other tables. The meaning of QPDet^{*1} is the same as that in Table VI. Faster R-CNN-O^{*1} and Faster R-CNN-O^{*2} denote other implementations using the general OBB representation in the first stage and the proposed OBB representation in the second stage of the network, respectively. Gliding Vertex* and QPDet^{*2} represents other implementations of the original methods using the proposed OBB representation in the detection head, and were retrained in our software environment. Similarly, Oriented R-CNN^{*1} and Oriented R-CNN^{*2} represent other implementations of the original method, which use different OBB representations $(x_h, y_h, w_h, h_h, \Delta a, \Delta b)$ and $(x_r, y_r, L_a, L_b, L_h)$ in the detection head.

TABLE IX
COMPARISON OF DIFFERENT OBB REPRESENTATIONS FOR DIFFERENT NETWORKS

OBB Representation	Baseline methods	mAP_{50}/(%)	mAP_{75}/(%)	$mAP_{50:95}$/(%)
$(x_r, y_r, w_r, h_r, \theta_r)$	RetinaNet-O	69.41	43.42	41.36
$(x_h, y_h, w_h, h_h, \frac{s_1}{w_h}, \frac{s_2}{h_h}, \frac{s_3}{w_h}, \frac{s_4}{h_h})$	RetinaNet-O	56.82	10.45	20.93
$(x_h, y_h, w_h, h_h, \Delta a, \Delta b)$	RetinaNet-O	70.98	43.58	42.24
$(x_r, y_r, R, \Delta\alpha, \Delta\beta)$	RetinaNet-O	69.92	41.74	40.64
$(x_r, y_r, L_a, L_b, L_h)$	RetinaNet-O	72.54	47.08	44.54
$(x_r, y_r, w_r, h_r, \theta_r)$	Faster R-CNN-O	69.60	24.10	33.46
$(x_h, y_h, w_h, h_h, \frac{s_1}{w_h}, \frac{s_2}{h_h}, \frac{s_3}{w_h}, \frac{s_4}{h_h})$	Faster R-CNN-O	69.75	26.48	32.14
$(x_h, y_h, w_h, h_h, \Delta a, \Delta b)$	Faster R-CNN-O	76.03	49.56	46.36
$(x_r, y_r, R, \Delta\alpha, \Delta\beta)$	Faster R-CNN-O	76.04	49.61	46.13
$(x_r, y_r, L_a, L_b, L_h)$	Faster R-CNN-O	76.41	49.28	46.50

It can be seen that the oriented object detectors usually obtain higher performance by using the special OBB representations in the detection head, as shown in Table VIII (row #2 versus row #4, row #5 versus row #6, row #7 versus row #8 or #9, and row #10 versus row #11 or #12). Besides, we found that the consistency of OBB representation in two-stage regressions is insufficient to ensure optimal object detection results in two-stage methods, as shown in Table VIII (row #7 versus row #9). Using a well-performed OBB representation is also important for the detection head. As shown in Table VIII, we can see the superiority of our OBB representation over the other representations for the other methods, regardless of whether it is used in the first or second stage. More specifically, when considering only the OBB representation in first stage, the UF-RPN adopts the five-distance OBB representation $(x_r, y_r, L_a, L_b, L_h)$ achieving 76.11% in mAP_{50}, 50.23% in mAP_{75}, and 46.46% in $mAP_{50:95}$ with improvements of at least 6% compared with the detection results of Faster R-CNN-O. By using the unified five-distance OBB representation in the FD-Head, our method achieves 76.57% in mAP_{50}, 49.18% in mAP_{75}, and 46.54% in $mAP_{50:95}$, further improving mAP_{50} and $mAP_{50:95}$, while slightly decreasing mAP_{75} compared with using UF-RPN alone.

4) Generation of the Five-Distance OBB Representation: In Table IX, we compare the mAP_{50}, mAP_{75}, and $mAP_{50:95}$ results of different OBB representations on the one-stage and two-stage networks (RetinaNet-O and Faster R-CNN-O) to evaluate the superiority of the proposed five-distance OBB representation. All these methods are retrained under the same environment and use the same data augmentation strategies in the training process.

As shown in Table IX, the single-stage RetinaNet-O using the proposed five-distance OBB representation $(x_r, y_r, L_a, L_b, L_h)$ can achieve at least 3% improvements in mAP_{50}, mAP_{75} and $mAP_{50:95}$ results on the DOTA-v1.0 dataset compared with the RetinaNet-O using the general OBB representation $(x_r, y_r, w_r, h_r, \theta_r)$. For the two-stage Faster R-CNN-O, the proposed five-distance OBB representation also brings improvements of more than 6% compared with Faster R-CNN-O using the general OBB representation. More specifically, both the one-stage and two-stage methods using the proposed five-distance OBB representation achieve more accurate detection results compared with those methods using the other OBB representations in Table IX.

5) Effectiveness of the SC-Loss: The SC-Loss, which combines the smooth-L1 loss $\mathcal{L}_1$ and the SC-Loss $\mathcal{L}_2$, $\mathcal{L}_3$, and $\mathcal{L}_4$, is used to calculate the localization losses of objects in our method. We set different weights for these losses to choose the optimal hyperparameters. The comparison results under different hyperparameter settings for the SC-Loss are shown in Table X. In Table X, all experiments use the UF-RPN to generate rotational proposals and the FD-Head to obtain detection results on the dota-v1.0 dataset. $[\beta_1, \beta_2, \beta_3]$ is different sets of hyperparameter.

TABLE X
COMPARISON OF DIFFERENT SETTINGS OF SC-LOSS

SC-Loss	$[\beta_1, \beta_2, \beta_3]$	mAP_{50}/(%)
$\beta_1\mathcal{L}_2$	[0.8, 0, 0]	75.43
$\beta_1\mathcal{L}_2$	[1.0, 0, 0]	75.50
$\beta_1\mathcal{L}_2 + \beta_2\mathcal{L}_3$	[1.0, 0.3, 0]	75.56
$\beta_1\mathcal{L}_2 + \beta_2\mathcal{L}_3$	[1.0, 0.5, 0]	75.87
$\beta_1\mathcal{L}_2 + \beta_2\mathcal{L}_3$	[1.0, 0.7, 0]	75.62
$\beta_1\mathcal{L}_2 + \beta_3\mathcal{L}_4$	[1.0, 0, 0.1]	76.12
$\beta_1\mathcal{L}_2 + \beta_3\mathcal{L}_4$	[1.0, 0, 0.2]	75.86
$\beta_1\mathcal{L}_2 + \beta_2\mathcal{L}_3 + \beta_3\mathcal{L}_4$	[1.0, 0.5. 0.2]	75.89
$\beta_1\mathcal{L}_2 + \beta_2\mathcal{L}_3 + \beta_3\mathcal{L}_4$	[1.0, 0.5, 0.1]	76.57

As shown in Table X, all these losses are based on $\mathcal{L}_1$. The losses in rows #2 and #3 include $\mathcal{L}_1$ and $\mathcal{L}_2$ with different weights. The losses from row #4 to row #6 incorporate $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$. The losses in rows #7 and #8 incorporate $\mathcal{L}_1$ and $\mathcal{L}_2$ and $\mathcal{L}_4$. Finally, the losses in rows #9 and #10 incorporate $\mathcal{L}_1$ and $\mathcal{L}_2$, $\mathcal{L}_3$ and $\mathcal{L}_4$. It can be seen that appropriately adjusting the weights of these losses is also beneficial for improving the detection results. In Table X, when β_1 is larger than 1.0, the total loss pays more attention to $\mathcal{L}_2$, and the mAP_{50} of our method becomes smaller than the best detection performance. When β_3 is smaller than 0.1, $\mathcal{L}_4$ contributes little to the total loss, resulting in a decrease in the accuracy of oriented object detection. When the weighting parameters are set to $[1.0, 0.5, 0.1]$, our method can achieve 76.57% mAP_{50} on the DOTA-v1.0 dataset, which gains an increase of 1.07% mAP_{50} compared with the combination of $\mathcal{L}_1$ and $\mathcal{L}_2$ in row #3 of Table X.

V. CONCLUSION

This article proposes a unified five-distance OBB representation, which avoids direct regression of angle parameters and uses the fewest parameters to represent the OBBs. We then establish the network UFDet with UF-RPN and FD-Head based on the new five-distance OBB representation to improve the accuracy of oriented object detection in remote sensing images. The UF-RPN can generate high-quality rotational proposals by directly transforming the horizontal anchors into OBBs in one step without additional processes based on the five-distance OBB representation. The FD-Head also utilizes the five-distance OBB representation in the detection head to further improve the accuracy of oriented object detection. We conduct extensive experiments on four remote sensing image datasets to demonstrate the superiority of our method. The experimental results show that our method achieves higher accuracy compared with other SOTA methods.

REFERENCES

[1] 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.

[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] Y. Qiao, L. Miao, Z. Zhou, and Q. Ming, "A novel object detector based on high-quality rotation proposal generation and adaptive angle optimization," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5617715.

[4] 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.

[5] H. Li et al., "TIR-net: Task integration based on rotated convolution kernel for oriented object detection in aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 4705313.

[6] 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. (CVPR)*, Jun. 2014, pp. 580–587.

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

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han, "Oriented R-CNN for object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2021, pp. 3500–3509.

[15] Y. Yao et al., "On improving bounding box representations for oriented object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5600111.

[16] X. Yang, L. Hou, Y. Zhou, W. Wang, and J. Yan, "Dense label encoding for boundary discontinuity free rotation detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15814–15824.

[17] Z. Zhao and S. Li, "ABFL: Angular boundary discontinuity free loss for arbitrary oriented object detection in aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5611411.

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

[19] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf.*, Oct. 2016, pp. 21–37.

[20] X. Zhu et al., "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. 9th Int. Conf. Learn. Represent.*, 2020, pp. 1–16.

[21] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

[22] 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.

[23] Y. Liu, Q. Li, Y. Yuan, Q. Du, and Q. Wang, "ABNet: Adaptive balanced network for multiscale object detection in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5614914.

[24] C. Zhang, K.-M. Lam, T. Liu, Y.-L. Chan, and Q. Wang, "Structured adversarial self-supervised learning for robust object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5613720.

[25] J. Liu et al., "Enhancing object detection with Fourier series," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 4, pp. 2581–2596, Apr. 2025.

[26] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

[27] 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.

[28] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Represent.*, Virtual Event, Austria, 2020, pp. 1–22.

[29] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 474–490.

[30] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 765–781.

[31] 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.

[32] Y. Liu, W. Guo, C. Yao, and L. Zhang, "Dual-perspective alignment learning for multimodal remote sensing object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, 2025, Art. no. 5404015.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

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

[40] L. Liu, Z. Pan, and B. Lei, "Learning a rotation invariant detector with rotatable bounding box," 2017, *arXiv:1711.09405*.

[41] X. Yang, J. Yan, Z. Feng, and T. He, "R3Det: Refined single-stage detector with feature refinement for rotating object," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 3163–3171.

[42] 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.

[43] 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)*, vol. 139, M. Meila and T. Zhang, Eds., Jul. 2021, pp. 11830–11841.

[44] X. Yang et al., "Learning high-precision bounding box for rotated object detection via Kullback–Leibler divergence," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 18381–18394.

[45] X. Yang et al., "The KFIoU loss for rotated object detection," in *Proc. 11th Int. Conf. Learn. Represent.*, Kigali, Rwanda, 2022, pp. 1–17.

[46] Q. Ming, L. Miao, Z. Zhou, J. Song, Y. Dong, and X. Yang, "Task interleaving and orientation estimation for high-precision oriented object detection in aerial images," *ISPRS J. Photogramm. Remote Sens.*, vol. 196, pp. 241–255, Feb. 2023.

[47] J. Han, J. Ding, N. Xue, and G. 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.

[48] Y. Tian et al., "FPNFormer: Rethink the method of processing the rotation-invariance and rotation-equivariance on arbitrary-oriented object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5605610.

[49] J. Wang, F. Li, and H. Bi, "Gaussian focal loss: Learning distribution polarized angle prediction for rotated object detection in aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4707013.

[50] 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.

[51] 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.
[52] G. Cheng et al., "Anchor-free oriented proposal generator for object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5625411.
[53] 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.
[54] Y. Zeng, Y. Chen, X. Yang, Q. Li, and J. Yan, "ARS-DETR: Aspect ratio-sensitive detection transformer for aerial oriented object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5610315.
[55] 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.
[56] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object detection in optical remote sensing images: A survey and a new benchmark," *ISPRS J. Photogramm. Remote Sens.*, vol. 159, pp. 296–307, Jan. 2020.
[57] G.-S. Xia et al., "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3974–3983.
[58] J. Ding et al., "Object detection in aerial images: A large-scale benchmark and challenges," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7778–7796, Nov. 2022.
[59] 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.
[60] 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.
[61] 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.
[62] 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.
[63] K. Chen et al., "MMDetection: Open MMLab detection toolbox and benchmark," 2019, *arXiv:1906.07155*.
[64] 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. 2022, pp. 3072–3080.
[65] Z. Chen et al., "PIoU loss: Towards accurate oriented object detection in complex environments," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 195–211.
[66] J. Dai et al., "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.
[67] Q. Ming, Z. Zhou, L. Miao, H. Zhang, and L. Li, "Dynamic anchor learning for arbitrary-oriented object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 2355–2363.
[68] G. Cheng et al., "Dual-aligned oriented detector," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5618111.
[69] 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.
[70] L. Hou, K. Lu, J. Xue, and Y. Li, "Shape-adaptive selection and measurement for oriented object detection," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 1, 2022, pp. 923–932.
[71] Z. Li, B. Hou, Z. Wu, B. Ren, and C. Yang, "FCOSR: A simple anchor-free rotated detector for aerial object detection," *Remote Sens.*, vol. 15, no. 23, p. 5499, Nov. 2023.
[72] 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.
[73] W. Qian, X. Yang, S. Peng, J. Yan, and Y. Guo, "Learning modulated loss for rotated object detection," in *Proc. AAAI Conf. Arti. Intell.*, May 2021, vol. 35, no. 3, pp. 2458–2466.
[74] 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.
[75] X. Xie et al., "Fewer is more: Efficient object detection in large aerial images," *Sci. China Inf. Sci.*, vol. 67, no. 1, pp. 1–19, Dec. 2023.

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, in 2017, and the M.S. degree in automation from the School of Automation, Beijing Institute of Technology (BIT), Beijing, China, in 2020, 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 Harbin Institute of Technology, Harbin, China, in 1986 and 1989, respectively, and the Ph.D. degree in control theory and engineering from China Academy of Launching Vehicle Technology, Beijing, China, in 2001.

Since 1992, she has been with Beijing Institute of Technology, Beijing, first as a Lecturer, since 1996 as an Associate Professor, and since 2001 as a Professor. Her main research interests include global positioning system (GPS), inertial navigation systems, inertial navigation system (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 Postdoctoral 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.

Yuhao Wang received the B.S. degree in building electricity and intelligence from Tianjin Chengjian University, Tianjin, China, in 2017, and the M.S. degree in control engineering from North China University of Technology, Beijing, China, in 2021. He is currently pursuing the Ph.D. degree with the School of Automation, Beijing Institute of Technology, Beijing.

His research interests include information fusion, pattern recognition, and autonomous driving.