DRDet: Dual-Angle Rotated Line Representation for Oriented Object Detection

Minjian Zhang, Heqian Qiu, Hefei Mei, *Graduate Student Member, IEEE*, Lanxiao Wang, Fanman Meng, *Member, IEEE*, Linfeng Xu, *Member, IEEE*, and Hongliang Li, *Senior Member, IEEE*

***Abstract*— In aerial scenes, oriented object detection is sensitive to the orientation of objects, which makes the formulation of orientation-aware object representation become a critical problem. Existing methods mostly adopt rectangle anchors or discrete points as object representation, which may lead to the feature aliasing between overlapping objects and ignore the orientation information of objects. To solve these issues, we propose a novel anchor-free oriented object detection network named DRDet, which adopts dual-angle rotated lines (DRLs) as object representation. Different from other object representations, DRL can adaptively rotate and extend to the boundary of the object according to its orientation and shape, which explicitly introduces the orientation information into the formulation of object representation. And it can adaptively cope with the geometric deformation of objects. Based on the DRLs, we design an orientation-guided feature encoder (OFE) to encode discriminant object features along each rotated line, respectively. Instead of encoding the rectangle feature, the OFE module adopts line features for orientation-guided feature encoding, which can alleviate the feature aliasing between neighboring objects or backgrounds. To further enhance the flexibility of DRLs, we design a dual-angle decoder (DD) that predicts two angle offsets according to the orientation-guided feature and converts the angle offsets and regression offsets into DRL representation, which can help to guide the adaptive rotation of each rotated line, respectively. Our proposed method achieves consistent improvement on both DOTA and HRSC2016 datasets. Extensive experiment results verify the effectiveness of our method in oriented object detection.**

***Index Terms*— Aerial scenes, feature extraction, object representation, oriented object detection.**

I. Introduction

ORIENTED object detection usually relies on the learning of object representation for accurate detection, which makes the modeling of object representation become a key issue. In particular, there are densely packed objects in the aerial scenarios, which can result in overlaps between objects and aggravate the difficulty in modeling adaptive representations of oriented objects. Existing oriented object detectors mainly utilize two types of object representations: anchor-based object representation and point-based object representation.

Mainstream anchor-based oriented object detectors [1], [2], [3], [4], [5], [6] follow the region-proposal paradigm regions with convolutional neural network (R-CNN), which can be divided into two steps: the generation of oriented candidate regions and the refinement of regression and classification on these proposals. The generation of oriented candidate regions often requires a series of predefined anchor boxes and the oriented proposals are obtained through a well-designed transformation process. Then, the refinement of regression and classification are performed on these oriented proposals based on the rectangle proposal features. Besides, some anchor-based oriented object detectors [7], [8] directly predict the oriented bounding boxes and categories according to the rectangle anchor boxes without the generation of proposals, and require feature alignment through specific feature operators. These anchor-based methods model the rectangular anchors or proposals as object representation and perform object feature extraction under the guidance of rectangle object representation. However, when there is overlap between objects, rectangular object representation is prone to introduce background noise, making the network prediction vulnerable to background interference. As shown in Fig. 1(a), the predictions of the harbor are easily disturbed by other objects (e.g., ships) within them.

Existing point-based oriented object detectors [9], [10], [11], [12], [13], [14] discard a series of predefined anchor boxes, and adopt key points or adaptive point sets as object representation. In a general detection pipeline, these point-based oriented object detectors usually use key points or adaptive point sets to guide the object feature extraction, and then generate offset points to be transformed into rotated bounding boxes. However, key points or adaptive point sets lack the ability to explicitly model object representation by utilizing the geometric information (e.g., orientation) from objects. As shown in Fig. 1(b) and (c), discrete points are difficult to be aware of the global information and may fail in introducing the orientation information into object representation, which can lead to incomplete regression prediction and inaccurate angle predictions. In addition, converting discrete points into polygons for calculating regression loss may result in suboptimal results.

Oriented object detection relies on object representations for precise classification and regression. However, existing

Manuscript received 5 December 2022; revised 12 April 2023 and 20 June 2023; accepted 17 August 2023. Date of publication 4 September 2023; date of current version 13 September 2023. This work was supported in part by thep National Natural Science Foundation of China under Grant 61831005, Grant 62271119, and Grant 62071086. *(Corresponding authors: Hongliang Li; Heqian Qiu; Lanxiao Wang.)*

The authors are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: mjzhang_ivip@std.uestc.edu.cn; hqqiu@std.uestc.edu.cn; hfmei@std.uestc.edu.cn; lanxiao.wang@std.uestc.edu.cn; fmmeng@uestc.edu.cn; lfxu@uestc.edu.cn; hlli@uestc.edu.cn).

Digital Object Identifier 10.1109/TGRS.2023.3311870

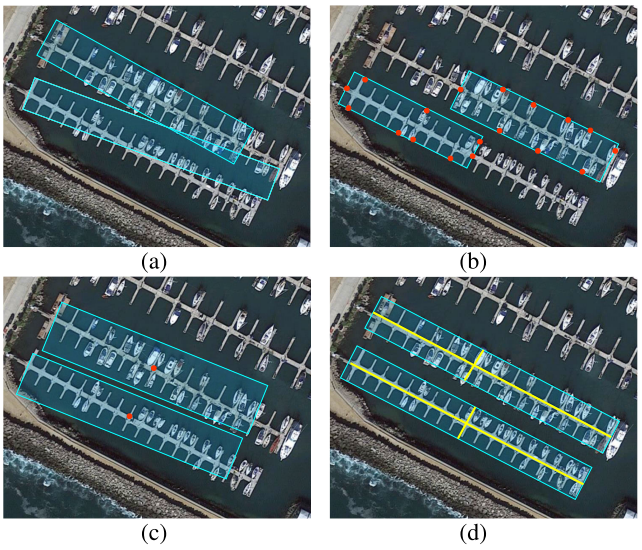


Fig. 1. Different object representation of existing methods. (a) Anchor-based object representation, which adopts the rectangle bounding boxes to represent objects. (b) Adopts a set of points to represent objects and guide feature extraction. (c) Uses a single center point as the object representation. (d) Our DRL representation is denoted by yellow lines. The red points denote the feature sampling points of objects, and the coral blue rectangles represent the predicted bounding boxes.

oriented object detectors using anchors or points as object representations usually suffer heavily from several drawbacks.

1) Since overlaps between objects are common in aerial scenarios, anchor-based oriented object detectors that adopt rectangular anchor boxes as object representation are subject to background noise. Besides, these anchor-based methods often rely on manually set horizontal anchors for object regression.
2) The point-based oriented object detectors that adopt discrete points as object representation cannot explicitly model the orientation information from objects and lack the ability to be aware of the global information of objects.

To ease the above issues in both anchor-based and point-based methods, we propose a novel anchor-free oriented object detection network named DRDet, which discards the preset anchors for regressing objects. To enhance the perception of orientation and global information from objects, the proposed DRDet adopts a pair of rotated lines to represent oriented objects named DRL representation. Different from existing methods, DRL representation can adaptively rotate and extend to the boundary of the object, which explicitly utilizes the orientation information to model object representation. Besides, compared with point-based representation, the DRL representation can provide more continuity information within objects. To alleviate the background noise caused by overlapping objects, DRDet obtains an orientation-guided feature encoder (OFE), which encodes the discriminant object feature along each rotated line separately, for the generation of orientation-guided feature maps. Instead of rectangular feature regions, the OFE module adopts line features for orientation-guided feature encoding, which can introduce less background noise when there is an overlap between objects. Remote sensing images (RSIs) are more complex than text content because of the weak discrimination and overlapping of objects in RSIs. Under such characteristics of RSIs, our DRL presents a discriminative object representation, and OFE can alleviate noise interference caused by overlapping objects. To accommodate the deformation of oriented objects, we propose the dual-angle decoder (DD) to provide a more flexible DRL representation. Based on the encoded feature maps, the DD module is adopted to predict two angle offsets and decode the predicted angle offsets and regression offsets into DRL representation. Then, the DD module can guide the adaptive rotation of each rotated line separately, thus enhancing the flexibility of DRLs.

We summarize the main contributions of the proposed approach as follows.

1) We propose a novel anchor-free oriented object detector named DRDet, which adopts DRLs to adaptively represent oriented objects. To the best of our knowledge, DRL is the first attempt to adopt two orientation-adaptive rotated lines for introducing the orientation information into an oriented object representation.
2) We propose a novel orientation-guided feature encoder to encode the discriminant object feature along each rotated line, which can alleviate feature aliasing by adopting the line feature. In addition, we propose a DD to enhance the flexibility of DRL representation and output accurate angle predictions.
3) To verify the effectiveness of our method, we conduct extensive experiments on DOTA [15] and HRSC2016 [16] datasets. The results demonstrate the effectiveness of our method on oriented object detection.

The remaining content of this article is organized as follows. Section II introduces and analyzes the related works. Section III introduces the proposed DRDet in detail. Extensive experiment results are reported and analyzed in Section IV. Finally, the conclusion is summarized in Section V.

II. Related Works

The object representation of both generic object detection methods and oriented object detection methods can be divided into two types: anchor-based representation and point-based representation. Anchor-based generic object detectors [17], [18], [19], [20] heuristically preset a variety of rectangle anchors to represent possible objects, and perform regression and classification in a single-stage or multistage pipeline according to the rectangle object features. In contrast, point-based generic object detectors [21], [22], [23], [24], [25], [26] adopt single point or a set of points as the object representation, and perform predictions based on the point-wise features. However, the direction of an oriented object varies greatly in the aerial scene and directly applying the generic object detectors to the oriented object detection may produce inaccurate results. To address this issue, some anchor-based oriented object detectors [1], [2], [3], [4], [5], [6] follow the generic R-CNN detection paradigm and usually require the specific transformation process for oriented proposals. Besides, other anchor-based

oriented object detectors [7], [8] adopt the generic one-stage detection paradigm for direct regression and classification predictions, and often achieve feature alignment through specific feature operators. Furthermore, point-based oriented object detectors [9], [10], [11], [12], [13], [14] usually use key points or adaptive point sets to represent oriented objects and guide the feature extraction, and then group these discrete points into rotated bounding boxes.

A. Anchor-Based Object Representation

The mainstream anchor-based generic object detection methods [18], [27] [28] usually rely on the region proposal mechanism. Specifically, these methods adopt an region proposal network (RPN) to predict a series of rectangle proposals according to the predefined anchors, and then extract features from proposals for subsequent refining predictions. However, these methods perform poorly in detecting multiscale objects. Lin et al. [29] proposed feature pyramid network (FPN) for multiscale feature fusion, which improves the ability of multiscale object recognition. As for the anchor-based single-stage detector, it relies on the densely arranged anchors and performs dense regression and classification predictions on these anchors. However, such a method often suffers from the problem of class imbalance between foreground and background. To address this problem, Lin et al. [30] proposed focal loss to down-weight the loss of well-classified samples and make the network focus on the training of difficult samples.

In aerial scenes, the scale, orientation, and shape of instances vary enormously, posing a great challenge to oriented object detection [15]. Some oriented object detectors [31], [32], [33] cope with multiscale and arbitrarily rotated objects by placing multiple anchors at each position, but this incurs significant computational overhead. To address this issue, Ding et al. [4] proposed the region of interest (RoI) transformer that serves as the transformation module from horizontal RoIs to oriented RoIs, which can reduce a large number of anchors. However, the complex transformation process in the RoI transformer increases the computation cost of the network. To balance the quality of proposals and the complexity of the network, Xie et al. [6] proposed oriented R-CNN based on the midpoint offset representation, which converts the horizontal anchors into rotated RoIs based on four anchor regression parameters and two extra midpoint offsets. Xu et al. [5] proposed an oriented bounding box representation called gliding vertex, which represents oriented objects by gliding four vertices on each side of the horizontal box. Without the generation of candidate region, some work [7], [8] directly perform dense classification and regression prediction, but the misalignment between anchors and the features of objects remains a problem to be solved. To address the problem, Yang et al. [7] built a refined single-stage detector R^3Det that encodes positional information in the form of the feature interpolation at four vertices and center of the predicted bounding boxes, and sums up these features to reconstruct the feature map. S^2A-Net [8] tries to solve the inconsistency between classification and regression from the perspective of feature alignment. It designs an alignment convolution to adaptively extract features according to anchor shapes and adopts an active rotating filter [34] to obtain orientation-invariant features for classification features. Although the above methods can achieve superior performance, there are common defects in anchor-based object representation. Anchor-based oriented object detectors rely on the heuristically defined anchor boxes for object regression, and there may be aliasing in the extracted features in the case of densely packed oriented objects. Different from the above methods, our method proposes an orientation-guided feature encoder to encode the object features within the adaptive rotated lines, which can avoid background noise caused by overlapping objects.

B. Point-Based Object Representation

To avoid the defects of anchor, point-based generic object detectors [21], [22], [23], [24], [25], [26] usually adopt single point or a set of points as the object representation, and perform predictions based on the point-wise features. Recently, Law and Deng [21] proposed an anchor-free detector called CornerNet, which formulates the objects as pairs of corners, that is, the top-left corner and the bottom-right corner. Besides, CornerNet introduces corner pooling to accurately locate corners. Based on the corner representation, CenterNet [22] introduces the center keypoint to represent the object as a triplet, which introduces more recognizable information within the object. However, both CornerNet [21] and CenterNet [22] need to group these keypoints to make sure they belong to the same object. Zhou et al. [23] model object as the center point of its bounding box, and directly predicts the size, depth, orientation, and pose of the object through the center-point features. Fully convolutional one-stage object detection (FCOS) [24] treats the points inside the target as positive samples, predicts the category of the object, and regresses the bounding box with the single-point feature. However, single-point features often lack the boundary information of objects, Reppoints [25] adopts a set of representative points as the object representation, and extracts features based on the location of point sets for refined predictions. CrossDet [26] points out that discrete point sets make it easy to lose the continuous information inside the objects, and use horizontal crossline as the object representation.

These point-based generic object detection methods achieve superior performance in common scenarios but struggle to detect densely arranged oriented objects in aerial scenarios. To cope with the diverse shapes and directions of oriented objects, dynamic refinement network (DRN) [10] contains a feature selection module (FSM) to adaptively adjust the receptive field, and adopts a dynamic refinement head (DRH) based on the center feature for refined predictions. Wei et al. [11] presented a one-stage and NMS-free network called O^2-DNet, which first locates the intersection points of object midlines and directly regresses the corresponding endpoints of each middle line. Zand et al. [35] presented a classification-based oriented object detector, which outputs confidence scores for each image location at multiple scales. The bounding boxes are formed by finding the minimum surrounding box of the connected neighborhood that shares the same category. Yang et al. [36] proposed AR^2Det for real-time ship detection, which consists of a feature extraction module (FEM), a ship detector (SDet), and a center detector (CDet).

For accurate and efficient ship detection, the CDet adjusts the scores of bounding boxes predicted by SDet to select the final bounding boxes. Liu et al. [37] delivered a refined anchor-free oriented object detector named R2YOLOX, which proposes a Gaussian distribution sampling optimal transport assignment (GSOTA) method to simplify the label assignment optimal transport problem. Cheng et al. [38] delivered an anchor-free oriented proposal generator called AOPG, which abandons horizontal boxes and predicts high-quality oriented proposals in an anchor-free scheme. Guo et al. [12] proposed a convex-hull representation to model-oriented objects, and adopted a convex-hull feature adaptation approach to guide the feature extraction of objects. Li et al. [13] proposed an adaptive points assessment and assignment (APAA) strategy to assign the representative points as positive samples, which evaluates the quality of discrete points in terms of the classification, regression, orientation, point-wise correlation measurement. The APAA scheme can help the learning of point sets in the training stage, but this strategy cannot be applied in the inference stage without supervision information. The above point-based methods, which adopt key points or point sets as object representation, are weak in introducing the orientation information into object representation and lack the ability to capture the global information of objects. Different from the above methods, our proposed DRL representation can explicitly model objects according to the orientation information and provide continuous information within objects. Besides, our DD can guide the rotation of each rotated line to enhance the flexibility of DRL representation, which can provide more combinations to represent objects with various deformations. Compared with the CrossDet [26], our method explicitly introduces the orientation information into object representation, which can further guide the orientation-aware feature extraction in the OFE module.

III. Proposed Method

The overview of the proposed DRDet is shown in Fig. 2. In our method, the proposed DRDet builds on a multistage detection framework with a backbone of ResNet [39] and FPN [29]. In the initial detection stage, the OFE encodes the object features according to initialized rotated lines. Based on the encoded features, the DD predicts two angle offsets and converts them into rotated lines with regression offsets. In the refined detection stage, the OFE module separately encodes the discriminative features along each rotated line. Then, the DD module is adopted for dual-angle prediction and the predicted angles are averaged as the final angle output. At last, the refined regression prediction is performed according to the encoded feature from the OFE module. In Sections III-A–III-D, these components of DRDet are described in detail.

A. DRL Representation

Object representation is the core of object detection, which determines the ability of regression and classification. Different from anchor-based and point-based representation, the proposed DRL can adaptively rotate and extend to the boundary of the object, which explicitly utilizes the orientation information to model object representation. The DRL representation consists of two orientation-adaptive lines shown in Fig. 3. And we specify that both rotated lines $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ rotate around their respective midpoints $\boldsymbol{C}_H$ and $\boldsymbol{C}_V$, and the corresponding expressions are as follows:

$$\begin{aligned} \boldsymbol{S}_1 &= (x_l, x_r, y_1, \theta_1) \\ \boldsymbol{S}_2 &= \left(x_2, y_t, y_b, \theta_2 + \frac{\pi}{2}\right) \end{aligned} \tag{1}$$

where $\boldsymbol{S}_1$ is obtained by rotating a horizontal line by θ_1 and $\boldsymbol{S}_2$ is obtained by rotating a vertical line by θ_2. Note that we adopt radian as the unit for angle prediction offset. (x_l, y_1) and (x_r, y_1) denote the left and right endpoints of the horizontal line, and (x_2, y_t) and (x_2, y_b) denote the top and bottom endpoints of the vertical line, respectively. Besides, (x_2, y_1) denotes the intersection point of the horizontal and vertical lines, respectively. The DRL representation can flexibly model objects in a variety of combinations, as shown in Fig. 3(c) and (d). Compared with the point-based methods, our DRL representation can explicitly utilize the orientation information to model objects. Based on the DRL representation, our method does not require the conversion process from point sets to polygons, thus avoiding suboptimal network optimization.

In the initial stage, we first initialize two rotated lines to guide the feature extraction in an OFE, with the preset angles of 0 and $(\pi/2)$, respectively. The initialized rotated lines can be expressed as follows:

$$\begin{aligned} \boldsymbol{S}_1 &= (x_0 - \alpha s, x_0 + \alpha s, y_0, 0) \\ \boldsymbol{S}_2 &= \left(x_0, y_0 - \alpha s, y_0 + \alpha s, \frac{\pi}{2}\right) \end{aligned} \tag{2}$$

where s denotes the stride of the feature map. In the refinement stage, the OFE module encodes the discriminant object features based on the obtained DRL representation, so as to improve the regression quality of objects.

B. Orientation-Guided Feature Encoder

The OFE serves as the FEM for both initial and refined detection stages, which is shown in Fig. 2. According to the DRL representation, the OFE module can explicitly aggregate the discriminant object features along two directions of rotated lines while avoiding feature aliasing. Hence, the OFE module can encode the orientation-aware information into object features for subsequent dual-angle prediction and regression.

Specifically, the OFE module first converts rotated lines into consecutive sampling points for feature interpolation, according to the obtained DRL representation $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$. Taking $\boldsymbol{S}_1$ shown in Fig. 4(c) as an example, the red dots on the rotated line $\boldsymbol{S}_1$ represent the feature sampling points $\boldsymbol{P}_S$. As mentioned in Section III-A, these sampling points can be obtained by rotating the consecutive points of the horizontal line $\boldsymbol{L}_H$ around center $\boldsymbol{C}_H = (x_H, y_H)^{\mathrm{T}}$. Thus, we define a collection of consecutive points on the horizontal line $\boldsymbol{L}_H$ as $\{\boldsymbol{P}_H = (x_H - (w/2) + k, y_H) | k < \lfloor w \rfloor\}$ with k denoting an integer. The parameter k is determined by the width w of the horizontal line $\boldsymbol{L}_H$, and it ranges from 0 to $\lfloor w \rfloor$. The process

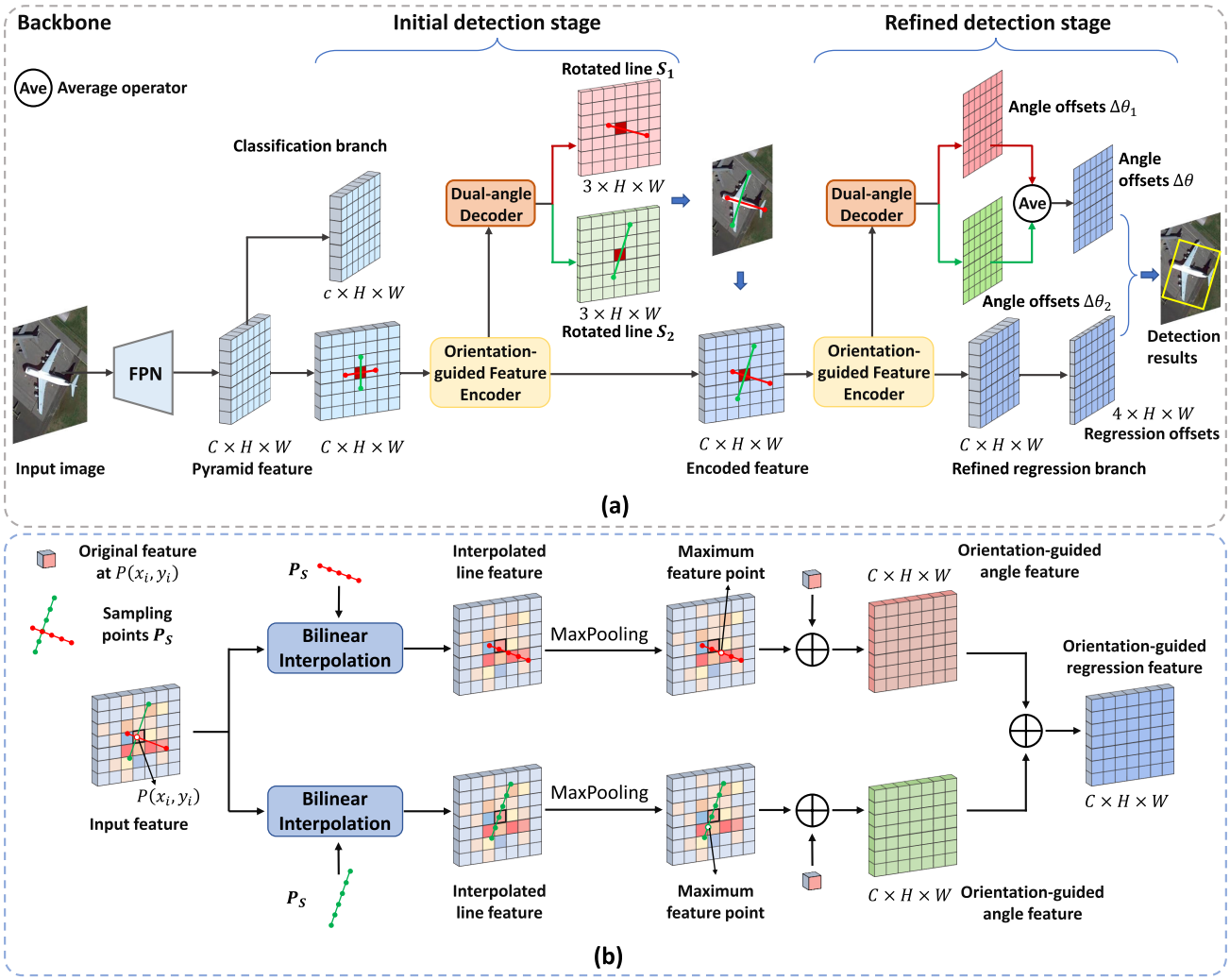


Fig. 2. (a) Overall framework of proposed DRDet. DRDet contains a backbone network with FPN and two detection stages. In the initial detection stage, the classification branch directly outputs the category results based on the pyramid features. Then, the OFE encodes the pyramid features for initial regression according to initialized rotated lines. With the encoded features, DD predicts two angle offsets and decodes these angle offsets into DRLs with regression offsets. In the refined detection stage, the OFE module takes the DRLs and the encoded features as input, and separately aggregates the object features along each rotated line. Then, the output of the encoded feature by the OFE module is applied for dual-angle prediction and refined regression prediction. At last, the dual-angle prediction is averaged as the final angle output. For brevity, the center-ness branch of each detection stage and initial regression branch are omitted. (b) Orientation-guided feature encoder. It takes the feature map and DRL representation as input, and then the OFE module can separately aggregate and fuse discriminant object features.

of rotation transformation from $\boldsymbol{P}_H$ to $\boldsymbol{P}_S$ can be represented as follows:

$$\boldsymbol{d}_S = \boldsymbol{P}_H - \boldsymbol{C}_H \tag{3}$$

$$\boldsymbol{P}_S = \mathbf{R} \otimes \boldsymbol{d}_S + \boldsymbol{C}_H. \tag{4}$$

Based on the coordinate of sampling points $\boldsymbol{P}_S$, we adopt the bilinear interpolation (BI) method to calculate the feature value of each sampling point and obtain the interpolated line feature, as shown in Fig. 2(b). The red line corresponds to the rotated line $\boldsymbol{S}_1$ while the green line represents the rotated line $\boldsymbol{S}_2$, and the colored points on each line represent its corresponding feature sampling points. To encode the orientation-aware information, we use max-pooling to capture the important object features within the corresponding rotated line, and then separately add the captured maximum feature point with the original feature at location $\boldsymbol{P}$ as shown in Fig. 2(b). The maximum feature point of each rotate line is represented by a white dot, shown in Fig. 2(b). By performing the above operations at each position of the original feature map, we can obtain two feature maps guided by the corresponding rotated lines called the orientation-guided angle feature, which is used for dual-angle prediction in subsequent DD. These orientation-guided angle features are fused together to obtain the orientation-guided regression feature, which are adopted for the following regression prediction.

C. Dual-Angle Decoder

To cope with the shape variation of objects, we propose a DD to enhance the flexibility of DRL representation. DD module can decode the predicted angle and regression offsets into DRLs and further guide the rotation of each rotated line separately. DD module contains two angle prediction branches, which separately predict angle offsets based on the encoded angle features from the OFE module, as shown in Fig. 5.

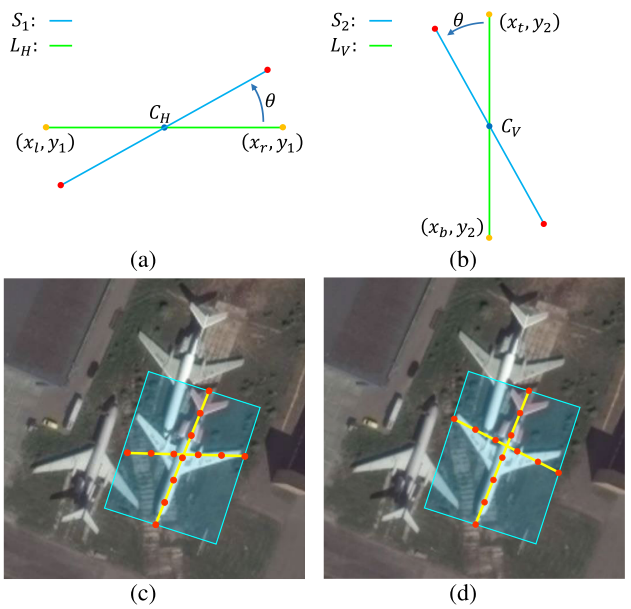


Fig. 3. DRL representation consists of two rotated lines, namely, $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, which are represented by blue line segments. In (a), the rotated line $\boldsymbol{S}_1$ is obtained through rotating the horizontal line $\boldsymbol{L}_H$ by θ around the midpoint $\boldsymbol{C}_H$. Similarly, the rotated line $\boldsymbol{S}_2$ in (b) is obtained through rotating the horizontal line $\boldsymbol{L}_V$ by θ around the midpoint $\boldsymbol{C}_V$. Yellow dots and red dots denote the endpoints of corresponding lines. (c) and (d) Two different DRL representations.

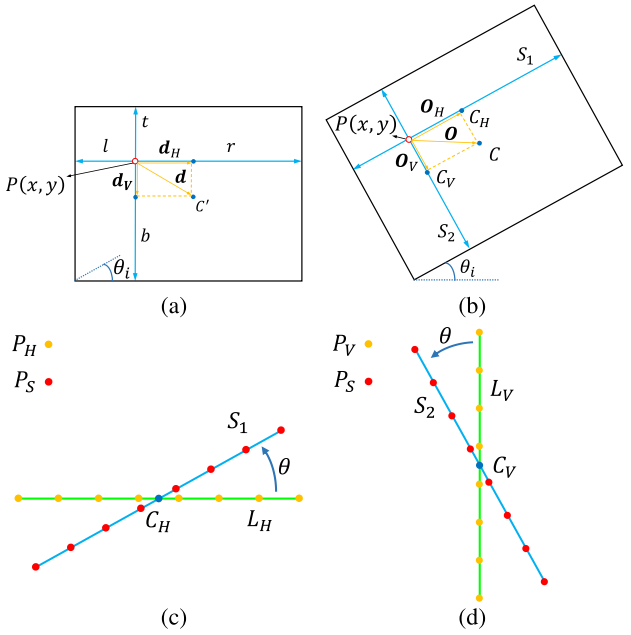


Fig. 4. Decoding transformation in DD and feature sampling in orientation-guided feature encoder. (a) Prediction offsets (l, t, r, b, θ_i) $(i = 1, 2)$ of location $\boldsymbol{P}$, and $\boldsymbol{d}_H$, $\boldsymbol{d}_V$, $\boldsymbol{d}$ represents the distance vectors in horizontal bounding box. (b) Oriented bounding box with rotated lines $\boldsymbol{S}_1$, $\boldsymbol{S}_2$ and offset vectors $\boldsymbol{O}_H$, $\boldsymbol{O}_V$, $\boldsymbol{O}$. (c) and (d) Feature sampling on rotated lines $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, respectively.

Compared with a pair of rotated lines with consistent angle offsets (e.g., $\theta_1 = \theta_2$), rotated lines with different angle offsets have more combinations for representing objects with various deformations.

In the initial detection stage, the DRL representation R_{line} : $\{\boldsymbol{S}_1 = (x_l, x_r, y_1, \theta_1), \boldsymbol{S}_2 = (x_2, y_t, y_b, \theta_2)\}$ is decoded from

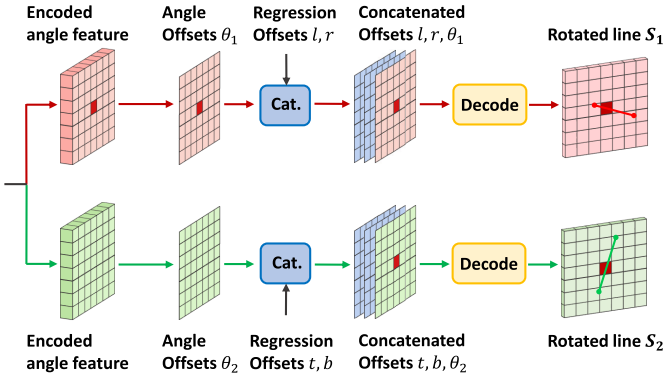


Fig. 5. DD in the first detection stage. Given the orientation-guided feature as input, the DD module separately outputs two angle offset maps. By concatenating the predicted angle maps with the regression offset map, we can obtain two offset maps that can be decoded into the rotated lines, respectively.

the prediction offsets. Given the feature maps $I \in \mathbb{R}^{H\times W\times C}$ with H, W, and C denoting the height, width, and channel of feature maps, respectively, the prediction offsets of location $\boldsymbol{P} = (x, y)^{\mathrm{T}}$ can be represented as $(l, t, r, b, \theta_1, \theta_2)$, as shown in Fig. 4(a). Note that the offset vector $\boldsymbol{O}$ can be decomposed into $\boldsymbol{O}_H$ and $\boldsymbol{O}_V$ as shown in Fig. 4(b). Thus, the decoding process from $\boldsymbol{P}$ to the center point of each rotated line can be derived from the transformation of $\boldsymbol{P}$ to the center $\boldsymbol{C}$ of the oriented bounding box, which can be represented as follows:

$$\mathbf{R}(\theta_i) = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \tag{5}$$

$$\boldsymbol{d} = \left(\frac{l-r}{2}, \frac{t-b}{2}\right)^{\mathrm{T}} \tag{6}$$

$$\boldsymbol{O} = \mathbf{R}(\theta_i) \otimes \boldsymbol{d} \tag{7}$$

$$\boldsymbol{C} = \boldsymbol{P} + \boldsymbol{O} \tag{8}$$

where $\mathbf{R}_i$ denotes the rotation matrix of corresponding angle θ_i $(i = 1, 2)$, and $\boldsymbol{d}$ denotes the distance vector in the horizontal bounding box, as shown in Fig. 4(a). And the offset $\boldsymbol{O}$ between location $\boldsymbol{P}$ and the center $\boldsymbol{C}$ is obtained by the matrix multiplication $\otimes$ in (7). Furthermore, the coordinates of center $\boldsymbol{C}$ are calculated by (8).

Based on above transformation, the center points $\boldsymbol{C}_H, \boldsymbol{C}_V$ of corresponding rotated lines from location $\boldsymbol{P}$ can be expressed as follows:

$$\boldsymbol{d}_H = \left(\frac{l-r}{2}, 0\right)^{\mathrm{T}}, \boldsymbol{d}_V = \left(0, \frac{t-b}{2}\right)^{\mathrm{T}} \tag{9}$$

$$\begin{aligned} \boldsymbol{O}_H &= \mathbf{R}(\theta_1) \otimes \boldsymbol{d}_H \\ \boldsymbol{O}_V &= \mathbf{R}(\theta_2) \otimes \boldsymbol{d}_V \end{aligned} \tag{10}$$

$$\begin{aligned} \boldsymbol{C}_H &= \boldsymbol{P} + \boldsymbol{O}_H = (x_H, y_H) \\ \boldsymbol{C}_V &= \boldsymbol{P} + \boldsymbol{O}_V = (x_V, y_V) \end{aligned} \tag{11}$$

where $\boldsymbol{d}_H$ and $\boldsymbol{d}_V$ represent the distance vectors in horizontal bounding box, and $\boldsymbol{O}_H$ and $\boldsymbol{O}_V$ represent the offsets from $\boldsymbol{P}$ to the center points $\boldsymbol{C}_H, \boldsymbol{C}_V$ of rotated lines $\boldsymbol{S}_1$, $\boldsymbol{S}_2$, as shown in Fig. 4(a) and (b). As mentioned above, $\boldsymbol{d}$ can be decomposed into $\boldsymbol{d}_H$ and $\boldsymbol{d}_V$ as calculated in (9). Similarly, the offsets $\boldsymbol{O}_H$ and $\boldsymbol{O}_V$ are calculated by the matrix multiplication in (10).

Then, the center points $\boldsymbol{C}_H$ and $\boldsymbol{C}_V$ of corresponding rotated lines are obtained by (11).

With the obtained center points, we can formulate the $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ as follows:

$$w = l + r, h = t + b, \tag{12}$$

$$\boldsymbol{S}_1 = \left(x_H - \frac{w}{2}, x_H + \frac{w}{2}, y_H, \theta_1\right)$$

$$\boldsymbol{S}_2 = \left(x_V, y_V - \frac{h}{2}, y_V + \frac{h}{2}, \theta_2\right) \tag{13}$$

where (x_H, y_H) denotes the center of rotated line $\boldsymbol{S}_1$, and (x_V, y_V) denotes the center of rotated line $\boldsymbol{S}_2$. Based on the obtained representation, objects can be adaptively represented by the combination of a pair of rotated lines. Then, the OFE module can further encode the orientation-guided features for the following predictions. Note that angle and regression offsets are decoded into DRL representation only in the first DD module. In the second detection stage, the DD module also predicts two angle offsets, and they are averaged to obtain the final angle predictions as output.

D. Dual-Angle Oriented Detector

In the subsequent contents, we elaborate on the DRDet in terms of network architecture, training targets, loss function, and inference.

1) Network Architecture: Our DRDet builds on the generic object detector FCOS [24] and performs oriented object detection in a two-stage pipeline based on the DRL representation and proposed OFE and DD modules. The initial detection stage includes the regression branch, classification branch, and center-ness branch, and the refined detection stage includes the regression branch and center-ness branch. FPN is applied in the backbone network for multiscale feature extraction. In addition, we adopt focal loss [30] to mitigate the imbalance between foreground and background classes. The regression output in FCOS can be formulated as the distances from the location (x, y) to the left, top, right, and bottom side of the bounding box, as shown below

$$\{(l, t, r, b)\}.$$

In order to accommodate oriented object detection, we reformulate the regression output into a rotated bounding box as follows:

$$\{(l, t, r, b, \theta_1, \theta_2)\}$$

where θ_i $(i = 1, 2)$ denotes the predicted dual-angle offsets. For an angular definition, we adopt a long-edge paradigm, with an angle range of $[-(\pi/2), (\pi/2))$. To suppress the low-quality detected boxes, the center-ness branch is preserved in both detection stages.

2) Training Targets: Following the setting in FCOS, the training target of DRDet consists of three parts: the classification part, the regression part, and the center-ness part. For location $\boldsymbol{P} = (x, y)^{\mathrm{T}}$, it is considered as a foreground object when it is within a single ground-truth (GT) box, and the category label c of the GT box is assigned to $\boldsymbol{P}$ as the classification targets. If it falls into the overlaps of multiple GT boxes, the smallest GT box is selected as a training target. Besides, the regression target $t = (\Delta l, \Delta t, \Delta r, \Delta b, \Delta\theta)$ and center-ness target at location $\boldsymbol{P}$ can be formulated as follows:

$$\boldsymbol{d} = \left(x - x_g, y - y_g\right)$$

$$\boldsymbol{O} = \mathbf{R}(\theta_g) \otimes \boldsymbol{d} = \left(O_x, O_y\right) \tag{14}$$

$$\Delta l = \frac{w_g}{2} + O_x, \Delta r = \frac{w_g}{2} - O_x$$

$$\Delta t = \frac{h_g}{2} + O_y, \Delta b = \frac{h_g}{2} - O_y, \Delta\theta_i = \theta_g \tag{15}$$

$$\mathrm{ctr}(t) = \sqrt{\frac{\min(\Delta l, \Delta r)}{\max(\Delta l, \Delta r)} \times \frac{\min(\Delta t, \Delta b)}{\max(\Delta t, \Delta b)}} \tag{16}$$

where $\{(x_g, y_g, w_g, h_g, \theta_g)\}$ indicates the parameters of GT box, and $\boldsymbol{d}$ indicates the distance from position (x, y) to the center of GT box (x_g, y_g). The offset $\boldsymbol{O}$ is calculated by matrix multiplication between $\mathbf{R}(\theta_g)$ and $\boldsymbol{d}$ in (14). Then, the regression targets can be formulated by (15). Based on the regression targets, the center-ness targets can be represented as (16).

3) Loss Function: The overall loss of DRDet consists of two stages and follows the multitask manner, which can be defined as follows:

$$\mathcal{L} = \sum_{k=1}^{K} \lambda_{\mathrm{cls}}^k \mathcal{L}_{\mathrm{cls}}^k + \lambda_{\mathrm{reg}}^k \mathcal{L}_{\mathrm{reg}}^k + \lambda_{\mathrm{ctr}}^k \mathcal{L}_{\mathrm{ctr}}^k \tag{17}$$

where k indicates the stage number of each loss, and classification loss, regression loss, and center-ness loss are represented as $\mathcal{L}_{\mathrm{cls}}^k$, $\mathcal{L}_{\mathrm{reg}}^k$, and $\mathcal{L}_{\mathrm{ctr}}^k$, respectively. λ_{cls}^k, λ_{reg}^k, and λ_{ctr}^k are adopted to weight the contribution of different tasks in multiple stages. In the initial stage, we conduct each loss to optimize the generation of rotated lines. Specifically, we adopt two regression losses $\mathcal{L}_{\mathrm{reg}}^{1,j}$ $(j = 1, 2)$ to optimize the corresponding rotated lines $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$. In the second stage, we only utilize regression loss $\mathcal{L}_{\mathrm{reg}}^2$ and center-ness loss $\mathcal{L}_{\mathrm{ctr}}^2$ to optimize the refinement of network. In our method, λ_{cls}^1, λ_{reg}^2, and λ_{ctr}^k are set to 1, and $\lambda_{\mathrm{reg}}^{1,j}$ $(j = 1, 2)$ are set to 0.25. Due to the consistency of losses in each stage, we omit the superscript in each loss, and the classification loss $\mathcal{L}_{\mathrm{cls}}$, regression loss $\mathcal{L}_{\mathrm{reg}}$, and center-ness loss $\mathcal{L}_{\mathrm{ctr}}$ can be detailed as follows:

$$\mathcal{L}_{\mathrm{cls}} = \frac{1}{N_{\mathrm{pos}}} \sum_{x,y} \mathcal{L}_c\left(\hat{c}_{x,y}, c_{x,y}\right) \tag{18}$$

$$\mathcal{L}_{\mathrm{reg}} = \frac{1}{N_{\mathrm{ctr}}} \sum_{x,y} \mathbb{1}_{\{c_{x,y}>0\}} \mathrm{ctr}\left(t_{x,y}\right) \cdot \mathcal{L}_r\left(\hat{t}_{x,y}, t_{x,y}\right)$$

$$N_{\mathrm{ctr}} = \sum_{x,y} \mathbb{1}_{\{c_{x,y}>0\}} \mathrm{ctr}\left(t_{x,y}\right) \tag{19}$$

$$\mathcal{L}_{\mathrm{ctr}} = \frac{1}{N_{\mathrm{pos}}} \sum_{x,y} \mathbb{1}_{\{c_{x,y}>0\}} \mathcal{L}_b\left(\mathrm{ctr}\left(\hat{t}_{x,y}\right), \mathrm{ctr}\left(t_{x,y}\right)\right) \tag{20}$$

where $\mathcal{L}_c$ indicates the focal loss for classification, $\mathcal{L}_r$ indicates the rotated IoU loss for oriented box regression, $\mathcal{L}_b$ indicates the binary cross entropy loss, and $\mathbb{1}_{\{c_{x,y}>0\}}$ is an indicator function. $c_{x,y}$ and $t_{x,y}$ denote the class label and regression target of the GT box at location (x, y). $\hat{c}_{x,y}$ and $\hat{t}_{x,y}$ represent the predicted category and regression offsets at location (x, y). N_{pos} is a number of positive locations and N_{ctr} is the sum of the center-ness targets at each positive location.

4) Inference: Our DRDet performs oriented object detection without the generation of RoIs. First, the backbone network extracts pyramid features from an image. And OFE module takes the pyramid features to encode object features for initial regression. Then, the DD module performs the dual-angle predictions and outputs the DRLs, which are adopted to guide the OFE module in the refined stage. With the encoded feature from OFE, the refined regression branch makes final regression predictions, and the DD module takes the averaged angles as final angle predictions. At last, we adopt NMS to remove low-quality overlapping bounding boxes.

IV. Experiments and Analysis

A. Datasets

1) DOTA [15]: It is a large-scale oriented object detection dataset for aerial scenes, where instances are highly diverse in scale, ratio, and orientation. The DOTA dataset consists of 2806 aerial images, including 188 282 annotated instances, which can be classified into 15 categories, namely, plane (PL), baseball diamond (BD), bridge (BR), ground track field (GTF), small vehicle (SV), large vehicle (LV), ship (SH), tennis court (TC), basketball court (BC), storage tank (ST), soccer-ball field (SBF), roundabout (RA), harbor (HA), swimming pool (SP), and helicopter (HC). The scale of images varies from 800 × 800 to 4000 × 4000. We crop the original images into patches of 1024 × 1024 in the form of a sliding window, with a stride of 824. And we adopt random rotation in training to prevent overfitting. For a multiscale setting, we resize the original image into three scales (0.5, 1.0, 1.5) and crop the image to a size of 1024 × 1024 with a stride of 512. We apply both the training set and the validation set for model training and submit the results on the test set to the DOTA server for evaluation.

2) HRSC2016 [16]: It is a high-resolution remote sensing dataset for ship recognition, which contains ship instances at sea and offshore. This dataset includes 436 training set images, 181 validation set images, and 444 test set images, with 1207 instances, 541 instances, and 1228 instances, respectively. The scale of images varies from 300 × 300 to 1500 × 900. Our method applies both the training set and validation set for model training and tests the trained models on the test set. All pictures are resized to a scale of 800 × 800, and random flipping and random rotation are applied in model training.

B. Implementation Details

In our approach, we adopt ResNet50 and ResNet101 [39] with FPN [29] as the backbone, and ResNet50 with FPN is applied as the default backbone in experiments if not specified. We select pyramid features from P_3 to P_7 as input to the detection head and adopt FCOS [24] as the baseline and maintain the consistent parameters in experiments. The settings for focal loss are: $\alpha = 0.25$ and $\gamma = 2.0$. For the DOTA dataset, the SGD optimizer is adopted for network training with an initial learning rate of 0.01 and batch size of 8. The momentum and weight decay are set to 0.9 and 0.0001, respectively. We train our models in 40 epochs for the DOTA dataset and the learning rate is divided by 10 at epochs 32 and 35. For the HRSC2016 dataset, we train the models in 72 epochs with an initial learning rate of 0.00125 and batchsize of 4. Model training and testing are performed on a server with 8 TITAN V. We reference the open source code in [40].

TABLE I
Comparison of Different Object Representation on DOTA. We Compare the Proposed DRL Representation With Baseline Method (FCOS) and Other Methods Based on Single Point, Point Sets, and Anchor Box

Methods	Schedule	mAP	mAP_{50}	mAP_{75}
Baseline	40e	40.33	72.36	39.49
Single Point	40e	43.54	73.30	44.97
Point Sets	40e	43.26	73.80	44.34
Anchor box	40e	42.11	73.69	42.25
DRL	40e	**44.91**	**74.85**	**46.38**

C. Ablation Studies

To validate the effectiveness of our method, we conduct a series of ablation experiments on DRL representation, orientation-guided feature encoder, and DD. These experiments are conducted on the DOTA dataset with ResNet50 FPN as the backbone.

1) DRL Representation: Our DRL representation can guide the feature encoding in the refined stage, thus we compare the DRL representation with other object representations in a two-stage pipeline. We adopt FCOS [24] as a baseline and modify it into a two-stage structure for a fair comparison with single-based methods. To compare with point-set-based methods, we adopt the convex hull [12] as the object representation to guide the object feature extraction. For the comparison with the anchor-based methods, we adopt AlignConv [8] to extract object features in the refined stage. As shown in Table I, the proposed DRL representation achieves a better performance of 74.85% mAP_{50}, outperforming baseline method, single-point method, point-set method and anchor-based method by 2.49%, 1.55%, 1.05%, 1.16% in the metric of mAP_{50}, respectively. The results show that the DRL representation can improve the ability to detect oriented objects.

2) Orientation-Guided Feature Encoder: To validate the effectiveness of the OFE, we investigate the effects of the OFE module in different detection stages, as shown in Table II. We first remove the OFE module from two detection stages to obtain the performance of 73.91% mAP_{50}. When the OFE module is applied only in the initial stage or refined stage, the performance is increased to 74.51% and 74.31% mAP_{50}, respectively. As the OFE module is applied in both detection stages, our method achieves a best performance of 74.85% mAP_{50}. These results demonstrate that the proposed Orientation-guided Feature Encoder can effectively use the orientation information to encode discriminant object features.

3) Dual-Angle Decoder: The results of our DD in different settings are shown in Table III. Specifically, we apply the DD module in different detection stages and compare the effects of the DD module. When the DD module is implemented in

TABLE II

EFFECTS OF OFE IN DIFFERENT STAGES. ✓ INDICATES THAT THE OFE MODULE IS APPLIED TO THIS STAGE

Module	Initial Stage	Refined Stage	mAP	mAP_{50}	mAP_{75}
OFE	-	-	44.66	73.91	45.79
	✓	-	**45.25**	74.51	**47.30**
	-	✓	44.68	74.31	46.51
	✓	✓	44.91	**74.85**	46.38

TABLE III

EFFECTS OF DD IN DIFFERENT SETTINGS. ✓ INDICATES THAT THE DD IS APPLIED TO THIS STAGE

Module	Initial Stage	Refined Stage	mAP	mAP_{50}	mAP_{75}
DD	-	-	44.26	74.08	45.42
	✓	-	44.88	74.56	**46.92**
	-	✓	44.75	73.85	46.51
	✓	✓	**44.91**	**74.85**	46.38

TABLE IV

EFFECT OF DIFFERENT $\lambda_{\text{REG}}^{1,j}$ $(j = 1, 2)$ IN LOSS FUNCTION ON DOTA DATASET

$\lambda_{reg}^{1,j}$	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
mAP_{50}	73.63	74.71	74.77	**74.85**	74.66	74.07	74.39	73.33	73.91

the initial stage, the results show that the DD module can improve the detection performance from 74.08% to 74.56% mAP_{50}. This is mainly because the DD module can enhance the flexibility of rotated lines, so as to improve the feature extraction of the following OFE module. When the DD module is implemented in the refined stage, our method achieves 73.85% mAP_{50}, which demonstrates that simply repeating the angle prediction and averaging is not helpful for oriented object detection. When the DD module is implemented in both initial and refined stages, our method obtains an improvement from 74.08% to 74.85% mAP_{50}, showing that DD can enhance the flexibility of DRLs and help to output more accurate predictions.

We discuss the effects of different parameters $\lambda_{\text{reg}}^{1,j}$ $(j = 1, 2)$ on network performance in Table IV. With the increase of $\lambda_{\text{reg}}^{1,j}$ $(j = 1, 2)$, the network will pay more attention to the generation of rotated lines $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$. And our method achieves the best performance of 74.85 % mAP_{50} when $\lambda_{\text{reg}}^{1,j}$ $(j = 1, 2)$ are set to 0.25. When $\lambda_{\text{reg}}^{1,j}$ $(j = 1, 2)$ are set to 0.45 or 0.5, the performance of the network output is reduced. The possible reason is that the network is focused too much on the generation of $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$. Based on the experiment results, we choose $\lambda_{\text{reg}}^{1,j} = 0.25$ $(j = 1, 2)$ as the final hyperparameter in (17).

At last, the results of ablation experiments on each proposed module are shown in Table V. With the proposed DRL representation and OFE module, our DRDet achieves 74.08% mAP_{50}, an improvement of 1.72% over baseline. Applying the proposed DD module to our method, it obtains the performance of 73.91% mAP_{50}, with an increase of 1.55% compared to baseline. When the proposed DRL representation, OFE, and DD modules are adopted in our method, the DRDet achieves

TABLE V

ABLATION EXPERIMENTS ON EACH PROPOSED MODULE. WE CHOOSE VANILLA FCOS AS THE BASELINE, AND DRL, OFE, AND DD DENOTE THE DRL REPRESENTATION, ORIENTATION-GUIDED FEATURE ENCODER, AND DUAL-ANGLE DECODER, RESPECTIVELY

	DRL	OFE	DD	mAP	mAP_{50}	mAP_{75}
Baseline	-	-	-	40.33	72.36	39.49
DRDet	✓	✓	-	44.26	74.08	45.42
DRDet	-	-	✓	44.66	73.91	45.79
DRDet	✓	✓	✓	**44.91**	**74.85**	**46.38**

the best performance of 74.85% mAP_{50}, outperforming the baseline by 2.49%. Extensive experiment results demonstrate the effectiveness of our proposed method in oriented object detection.

D. Comparisons With the State-of-the-Art

We compare the proposed DRDet with state-of-the-art methods on DOTA and HRSC2016 in this section. Due to the different image scales, augmentation strategies, and training strategies adopted by various methods, it is difficult to achieve a completely fair comparison. Therefore, our method performs comparative experiments under both single-scale and multi-scale settings.

1) Results on DOTA [15]: Under single-scale settings, our method adopts random rotation for data augmentation and achieves 76.40% mAP_{50} and 76.67% mAP_{50} with ResNet-50-FPN and ResNet-101-FPN backbone, respectively. Without extra data augmentation (e.g., random rotation), DRDet achieves 74.85% mAP_{50} with ResNet-50-FPN backbone, as shown in Table VI. When adopting the multi-scale images for training and testing, DRDet can achieve 79.34% mAP_{50} and 79.22% mAP_{50} on ResNet-50-FPN and ResNet-101-FPN backbone, respectively. Based on the advanced backbone Convnext-tiny-FPN [47], our DRDet achieves the performance of 77.63 % mAP_{50} and 76.54 % mAP_{50} with and without random rotation in single-scale training and testing. When adopting the multiscale images for training and testing, DRDet can achieve 79.85 % mAP_{50} on the Convnext-tiny-FPN backbone. Table VI includes the results of the comparative experiments, and our method achieves superior detection performance on 11 categories (e.g., small vehicles, large vehicles, swimming pools, harbors, and helicopters). The visualization results of our method and baseline FCOS are shown in Fig. 6. From the visualization results, we can observe that our method can accurately detect dense small objects (e.g., small vehicles, large vehicles, and ships) while locating large-scale targets (e.g., baseball diamond and soccer ball field) precisely. We test the inference time of the proposed DRDet on a single TITAN V GPU. With a ResNet-50-FPN backbone, DRDet spends about 0.091 s to process an image with 1024 × 1024 resolution on the DOTA dataset.

2) Results on HRSC2016 [16]: To compare with other methods, our DRDet is evaluated under both VOC2007 and VOC2012 metrics on the HRSC2016 dataset. With the ResNet-50-FPN backbone, our method achieves 90.05% and

TABLE VI

COMPARISON WITH THE STATE-OF-THE-ART METHODS ON DOTA-V1.0. THE RESULTS IN BOLD DENOTE THE BEST RESULTS IN EACH COLUMN. † MEANS TRAINING AND TESTING WITHOUT DATA AUGMENTATION. * DENOTES THE MULTISCALE IMAGES FOR TRAINING AND TESTING. CONVX DENOTES THE CONVNEXT BACKBONE NETWORK

Methods	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP_{50}
Anchor-based Representation:																	
RetinaNet [30]	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
CAD-Net [1]	R-101-FPN	87.80	82.40	49.40	73.50	71.10	63.50	76.60	90.90	79.20	73.30	48.40	60.90	62.00	67.00	62.20	69.90
CenterMap-Net [41]	R-50-FPN	88.88	81.24	53.15	60.65	78.62	66.55	78.10	88.83	77.80	83.61	49.36	66.19	72.10	72.36	58.70	71.74
DAL [42]	R-101-FPN	88.61	79.69	46.27	70.37	65.89	76.10	78.53	90.84	79.98	78.41	58.71	62.02	69.23	71.32	60.65	71.78
SCRDet [3]	R-101-FPN	**89.98**	80.65	52.09	68.36	68.36	60.32	72.41	90.85	**87.94**	86.86	65.02	66.68	66.25	68.24	65.21	72.61
R^3Det [7]	R-152-FPN	89.49	81.17	50.53	66.10	70.92	78.66	78.21	90.81	85.26	84.23	61.81	63.77	68.16	69.83	67.17	73.74
S^2A-Net [8]	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
RoI-Trans. [4]	R-101-FPN	88.65	82.60	52.53	70.87	77.93	76.67	86.87	90.71	83.83	82.51	53.95	67.61	74.67	68.75	61.03	74.61
Glinding Vertex [5]	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
Oriented R-CNN [6]	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	74.94	68.84	52.28	75.87
Point-based Representation:																	
PIoU [43]	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
DRN [10]	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
O^2-DNet [11]	H-104	89.31	82.14	47.33	61.21	71.32	74.03	78.62	90.76	82.23	81.36	60.93	60.17	58.21	66.98	61.03	71.04
CFA [12]	R-101-FPN	89.26	81.72	51.81	67.17	79.99	78.25	84.46	90.77	83.40	85.54	54.86	67.75	73.04	70.24	64.96	75.05
SASM (RepPoints-based) [14]	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
CBDA-Net [9]	DLA-34	89.17	**85.92**	50.28	65.02	77.72	82.32	87.89	90.48	86.47	85.90	66.85	66.48	67.41	71.33	62.89	75.74
Oriented RepPoints [13]	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
DarkNet-RI [35]	DarkNet-53	89.00	80.40	50.50	76.00	78.20	77.80	87.40	90.00	83.20	86.93	63.30	68.70	67.60	70.80	63.80	75.50
R2YOLOX-S [37]	CSPDarknet-S	88.18	81.66	42.31	61.69	80.31	78.65	88.22	90.88	80.68	86.22	48.21	58.59	73.53	72.51	52.00	72.25
AOPG [38]	R-50-FPN	89.27	83.49	52.50	69.97	73.51	82.31	87.95	90.89	87.64	84.71	60.01	66.12	74.19	68.30	57.80	75.24
PSCD (FCOS-based) [44]	R-50-FPN	89.06	73.61	49.03	62.34	75.18	77.69	88.00	90.85	82.63	72.97	61.48	64.20	65.77	72.88	67.88	72.90
rfpoint [45]	R-50-FPN	88.60	79.39	50.69	70.06	78.71	79.35	87.63	90.89	82.72	84.75	58.61	65.84	66.00	69.20	49.64	73.47
FCOSF [46]	R-50-FPN	88.43	78.59	48.04	65.17	79.80	80.24	87.08	90.90	82.75	84.65	58.39	66.44	64.66	71.88	57.60	73.64
Dual-angle Rotated Line Representation:																	
DRDet† (Ours)	R-50-FPN	88.25	80.93	48.66	65.17	81.06	81.98	87.87	90.87	82.20	84.83	62.77	64.56	73.39	72.46	57.78	74.85
DRDet (Ours)	R-50-FPN	88.98	82.63	49.63	64.14	81.77	83.99	88.25	**90.90**	86.86	85.89	58.80	66.65	74.88	78.14	64.58	76.40
DRDet (Ours)	R-101-FPN	88.95	83.37	49.77	62.67	81.62	83.39	88.10	90.89	86.99	85.04	65.08	67.69	75.64	81.01	59.85	76.67
DRDet* (Ours)	R-50-FPN	89.20	83.79	54.91	71.76	**82.48**	**84.84**	**89.18**	90.86	82.66	**87.87**	69.88	68.92	77.01	83.12	73.58	79.34
DRDet* (Ours)	R-101-FPN	88.94	80.84	**56.24**	74.00	82.36	84.01	89.07	90.71	83.58	87.23	**70.08**	69.02	77.51	82.99	71.71	79.22
DRDet† (Ours)	ConvX-tiny-FPN	88.51	83.57	49.07	66.56	80.65	82.90	88.09	90.90	84.90	85.26	66.00	65.17	76.08	80.55	59.92	76.54
DRDet (Ours)	ConvX-tiny-FPN	89.30	84.16	52.76	63.66	81.63	84.61	88.44	90.90	86.24	86.56	68.39	62.63	77.00	82.43	65.74	77.63
DRDet* (Ours)	ConvX-tiny-FPN	88.76	85.28	55.30	72.64	82.07	84.17	88.96	90.80	83.81	86.55	69.83	**71.61**	**77.60**	**83.60**	**76.74**	**79.85**

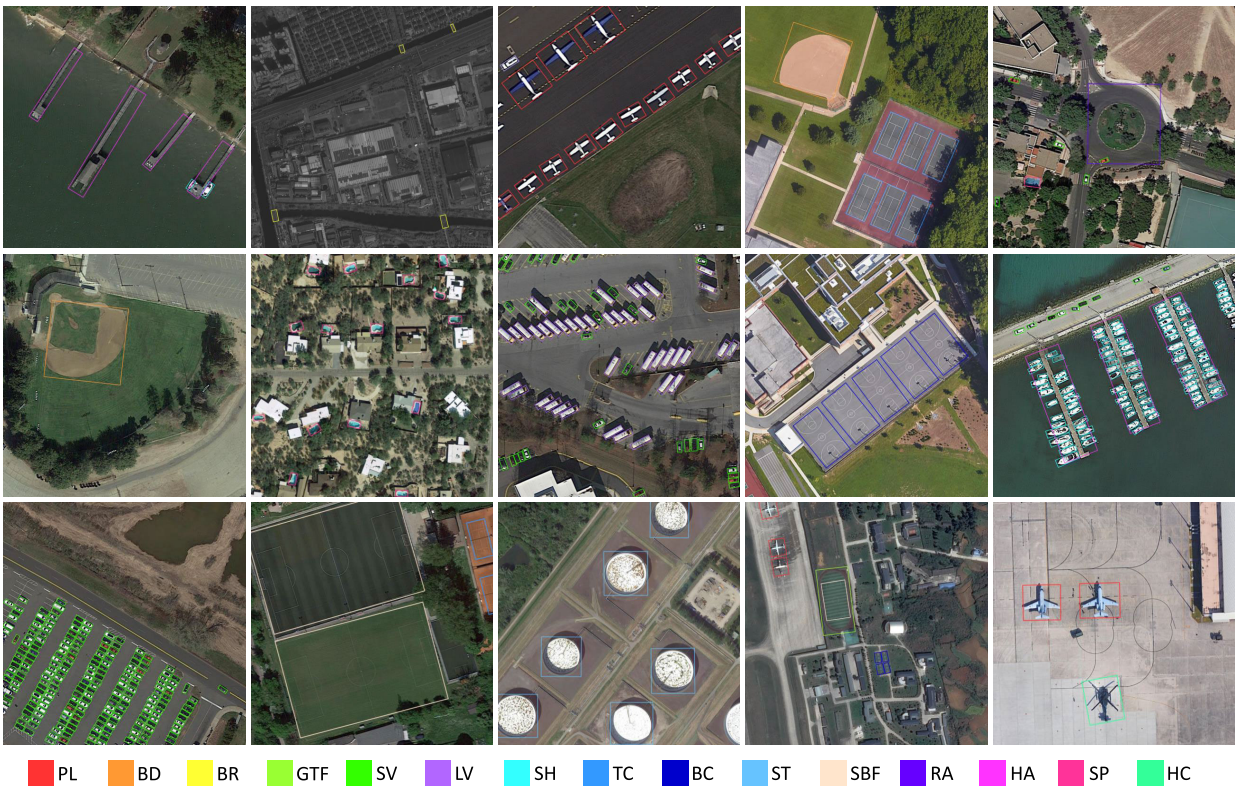


Fig. 6. Example detection results of the proposed method on DOTA [15] dataset with ResNet-50-FPN as backbone. Each category is colored accordingly.

96.70% mAP under the metrics of VOC2007 and VOC2012. With the ResNet-101-FPN backbone, our method achieves 90.10% and 96.64% mAP under VOC2007 and VOC2012 metrics. The results of the comparative experiment are shown in Table VII. The example detection results of our method are shown in Fig. 7. According to the visualized images,

TABLE VII

COMPARISON WITH THE STATE-OF-THE-ART METHODS ON HRSC2016. mAP_{50} (07) AND mAP_{50} (12) DENOTE RESULTS UNDER THE METRICS OF PASCAL VOC2007 and PASCAL VOC2012

Methods	Backbone	mAP_{50} (07)	mAP_{50} (12)
RRD [48]	VGG16	84.30	-
RoI-Trans. [4]	R-101-FPN	86.20	-
Glinding Vertex [5]	R-101-FPN	88.20	-
PIoU [43]	DLA-34	89.20	-
DRN [10]	H-104	-	92.70
R^3Det [7]	R-101-FPN	89.26	96.01
DAL [42]	R-101-FPN	89.77	-
CSL (FPN based) [49]	R-101-FPN	89.62	96.10
AR^2Det [36]	R-34-FPN	89.58	-
SASM (RepPoints-based) [14]	R-101-FPN	90.00	-
S^2A-Net [8]	R-101-FPN	90.17	95.01
AOPG [38]	R-50-FPN	**90.34**	96.22
PSCD (FCOS-based) [44]	R-50-FPN	89.91	-
rfpoint [45]	R-50-FPN	90.21	95.28
FCOSF [46]	R-50-FPN	89.99	96.15
DRDet (Ours)	R-50-FPN	90.05	**96.70**
DRDet (Ours)	R-101-FPN	90.10	96.64

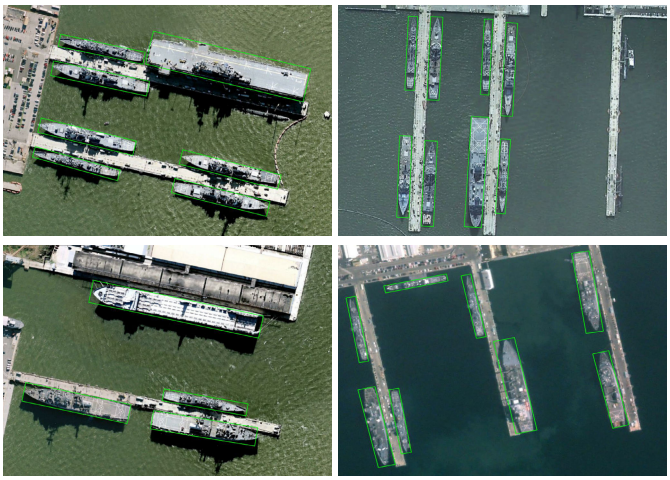


Fig. 7. Example detection results of the proposed method on HRSC [16] dataset with ResNet-101-FPN as backbone.

our method is able to accurately locate ships with various orientations and sizes.

V. CONCLUSION

In this article, we propose a novel anchor-free oriented object detection network named DRDet, which adaptively represents objects and extracts features based on DRL representation. To the best of our knowledge, our work is the first attempt to adopt adaptively growing rotated lines as oriented object representations. The DRLs can adaptively rotate and extend to the boundary of the objects according to the orientation and shape information. With the DRLs, the orientation-guide feature encoder is designed to encode orientation-aware features along each rotated line. Besides, the DD is proposed to perform accurate angle predictions and decode the predicted offsets into DRL representation, which can enhance the flexibility of DRL representation by guiding the rotation of each rotated line separately. Extensive experiments verify the effectiveness of our DRDet, which achieves superior performance on the DOTA and HRSC2016 datasets.

REFERENCES

[1] G. Zhang, S. Lu, and W. Zhang, "CAD-Net: A context-aware detection network for objects in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 10015–10024, Dec. 2019.

[2] R. Qin, Q. Liu, G. Gao, D. Huang, and Y. Wang, "MRDet: A multihead network for accurate rotated object detection in aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022.

[3] X. Yang et al., "SCRDet: Towards more robust detection for small, cluttered and rotated objects," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 8232–8241.

[4] 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. 2849–2858.

[5] 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. 2020.

[6] X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han, "Oriented R-CNN for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3520–3529.

[7] 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.*, May 2021, vol. 35, no. 4, pp. 3163–3171.

[8] J. Han, J. Ding, J. Li, and G.-S. Xia, "Align deep features for oriented object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021.

[9] S. Liu, L. Zhang, H. Lu, and Y. He, "Center-boundary dual attention for oriented object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021.

[10] 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. 11207–11216.

[11] H. Wei, Y. Zhang, Z. Chang, H. Li, H. Wang, and X. Sun, "Oriented objects as pairs of middle lines," *ISPRS J. Photogramm. Remote Sens.*, vol. 169, pp. 268–279, Nov. 2020.

[12] Z. Guo, C. Liu, X. Zhang, J. Jiao, X. Ji, and Q. Ye, "Beyond bounding-box: Convex-hull feature adaptation for oriented and densely packed object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8792–8801.

[13] 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. 1829–1838.

[14] L. Hou, K. Lu, J. Xue, and Y. Li, "Shape-adaptive selection and measurement for oriented object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 923–932.

[15] 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.

[16] 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. Int. Conf. Pattern Recognit. Appl. Methods*, vol. 2, 2017, pp. 324–331.

[17] 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)*, 2014, pp. 580–587.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 91–99.

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

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

[21] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 734–750.

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

[23] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.

[24] 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. 9627–9636.

[25] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "RepPoints: Point set representation for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9657–9666.

[26] H. Qiu et al., "CrossDet: Crossline representation for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3175–3184.

[27] 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.

[28] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 379–387.

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

[30] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2980–2988.

[31] Z. Liu, H. Wang, L. Weng, and Y. Yang, "Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 8, pp. 1074–1078, Aug. 2016.

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

[33] K. Li, G. Cheng, S. Bu, and X. You, "Rotation-insensitive and context-augmented object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2337–2348, Apr. 2017.

[34] Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao, "Oriented response networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 519–528.

[35] M. Zand, A. Etemad, and M. Greenspan, "Oriented bounding boxes for small and freely rotated objects," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022.

[36] Y. Yang et al., "AR2Det: An accurate and real-time rotational one-stage ship detector in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022.

[37] F. Liu, R. Chen, J. Zhang, K. Xing, H. Liu, and J. Qin, "R2YOLOX: A lightweight refined anchor-free rotated detector for object detection in aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022.

[38] G. Cheng et al., "Anchor-free oriented proposal generator for object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[40] Y. Zhou et al., "MMRotate: A rotated object detection benchmark using PyTorch," 2022, *arXiv:2204.13317*.

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

[42] 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, vol. 35, no. 3, pp. 2355–2363.

[43] Z. Chen et al., "PIoU loss: Towards accurate oriented object detection in complex environments," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2020, pp. 195–211.

[44] Y. Yu and F. Da, "Phase-shifting coder: Predicting accurate orientation in oriented object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 13354–13363.

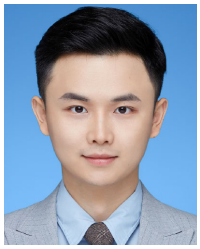
[45] K. Wang, Z. Xiao, Q. Wan, X. Tan, and D. Li, "Learnable loss balancing in anchor-free oriented detectors for aerial object," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023.

[46] C. Rao, J. Wang, G. Cheng, X. Xie, and J. Han, "Learning orientation-aware distances for oriented object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023.

[47] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 11976–11986.

[48] M. Liao, Z. Zhu, B. Shi, G.-S. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5909–5918.

[49] X. Yang and J. Yan, "Arbitrary-oriented object detection with circular smooth label," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 677–694.



Minjian Zhang received the B.E. degree in electronics information engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2018, where he is currently pursuing the Ph.D. degree in information and communication engineering under the supervision of Prof. Hongliang Li.

His research interests include computer vision and machine learning, especially the application of object detection in remote sensing and common scenarios.

Heqian Qiu received the Ph.D. degree in signal and information processing from the University of Electronic Science and Technology of China, Chengdu, China, in 2022.

She is currently a Post-Doctoral Researcher with the School of Information and Communication Engineering, University of Electronic Science and Technology of China. Her research interests include object detection, multimodal representative learning, computer vision, and machine learning.

Dr. Qiu has served as a reviewer for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *Journal of Visual Communication of Image Representation* (JVCI), Computer Vision and Pattern Recognition (CVPR), European Conference on Computer Vision (ECCV), and Association for the Advancement of Artificial Intelligence (AAAI).

Hefei Mei (Graduate Student Member, IEEE) received the B.E. degree in electronics information engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2021, where he is currently pursuing the master's degree in information and communication engineering under the supervision of Prof. Hongliang Li.

His research interests include computer vision, object detection, and few-shot learning.

Lanxiao Wang received the B.E. degree in electronics information engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2019, where she is currently pursuing the Ph.D. degree in information and communication engineering under the supervision of Prof. Li.

Her research interests include computer vision and machine learning, especially the application of deep learning on scene analysis and multimodal representation learning.

Fanman Meng (Member, IEEE) received the Ph.D. degree in signal and information processing from the University of Electronic Science and Technology of China, Chengdu, China, in 2014.

From 2013 to 2014, he was a Research Assistant with the Division of Visual and Interactive Computing, Nanyang Technological University, Singapore. He is currently a Professor with the School of Information and Communication Engineering, University of Electronic Science and Technology of China. He has authored or coauthored numerous technical articles in well-known international journals and conferences. His research interests include image segmentation and object detection.

Linfeng Xu (Member, IEEE) received the Ph.D. degree in signal and information processing from the School of Electronic Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2014.

From December 2014 to December 2015, he was with the Ubiquitous Multimedia Laboratory, The State University of New York at Buffalo, Buffalo, NY, USA, as a Visiting Scholar. He is currently an Associate Professor with the School of Information and Communication Engineering, UESTC. His research interests include machine learning, visual attention, image and video coding, visual signal processing, and multimedia communication systems.

Dr. Xu served as a Local Arrangement Chair for International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS) 2010 and Visual Communications and Image Processing (VCIP) 2016.

Hongliang Li (Senior Member, IEEE) received the Ph.D. degree in electronics and information engineering from Xi'an Jiaotong University, Xi'an, China, in 2005.

From 2005 to 2006, he joined the Visual Signal Processing and Communication Laboratory (VSPC), The Chinese University of Hong Kong (CUHK), Hong Kong, as a Research Associate. From 2006 to 2008, he was a Post-Doctoral Fellow at VSPC, CUHK. He is currently a Professor with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China. He has authored or coauthored numerous technical articles in well-known international journals and conferences. He is a coeditor of a Springer book titled *Video Segmentation and Its Applications*. He is involved in many professional activities. His research interests include image segmentation, object detection, image and video coding, visual attention, and multimedia processing.

Dr. Li was selected as the IEEE Circuits and Systems Society Distinguished Lecturer for 2022–2023. He received the 2019 and 2020 Best Associate Editor Award for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and the 2021 Best Editor Award for *Journal on Visual Communication and Image Representation*. He served as a Technical Program Chair for Visual Communications and Image Processing (VCIP) 2016 and Pacific-Rim Conference on Multimedia (PCM) 2017, the General Chair for International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS) 2017 and ISPACS 2010, the Publicity Chair for IEEE VCIP 2013, the Local Chair for IEEE International Conference on Multimedia and Expo (ICME) 2014, the Area Chair for VCIP 2022 and 2021, and a Reviewer Committee Member for IEEE International Symposium on Circuits and Systems (ISCAS) from 2018 to 2022. He served as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2018–2021). He is now an Associate Editor of *Journal on Visual Communication and Image Representation* and IEEE OPEN JOURNAL OF CIRCUITS AND SYSTEMS and an Area Editor of *Signal Processing: Image Communication* (Elsevier Science).