Orientation-First Strategy With Angle Attention Module for Rotated Object Detection in Remote Sensing Images

Yuxi Zhang, Yongcheng Wang, Ning Zhang, Zheng Li, Zhikang Zhao, Yunxiao Gao, Dongdong Xu, and Guangli Ben

***Abstract*—Recently, object detection in remote sensing images (RSIs) have received extensive attention and made significant progress. Nonetheless, the arbitrary orientations of objects in RSIs make their detection a challenging task. Most of the existing detection methods are difficult to extract the orientation features of objects due to the lack of directionality of conventional convolutions. In addition, the boundary discontinuity in angle regression affects the detection of object orientations. In response to these problems, this article proposes an orientation-first refinement detector (OFRDet), which is based on a strategy that enables the detector to detect the angle of an object ahead of others and presets oriented anchors. In OFRDet, we propose an angle encoding regression module (AERM) and an angle channel attention module (ACAM). AERM transforms angle detection into multiparameter regression, which eliminates boundary discontinuities. ACAM uses convolution kernels with different angles to extract directional features purposefully according to the preset oriented anchors. After these two modules, more accurate bounding boxes are generated and sent to the refined stage to obtain the final detection results. We evaluate our method and demonstrate the effectiveness of it by conducting experiments on two challenging and credible datasets, DOTA, HRSC2016. OFRDet achieves competitive results 79.56%, 96.29% mAP on the two datasets, respectively.**

***Index Terms*—Angle channel attention, angle encoding, remote sensing images, rotated object detection.**

I. Introduction

OBJECT detection is a technique in computer vision that requires locating and identifying the certain object in the image. Remote sensing images (RSIs) are more challenging to be detected since the scale of RSIs is larger and the content is more complex than that of ordinary natural images [1]. In addition, objects are unevenly distributed on RSIs and are generally small. With the continuous development of deep learning technology, neural networks are widely used in image processing. Meanwhile the object detection based on convolutional neural networks (CNNs) have made great progress. Numerous CNN-based object detection methods aimed at addressing the abovementioned challenges in RSIs have been proposed in recent years [2], [3], [4], [5], [6], [7], [8]. These methods have achieved pretty good results and solved some of the challenges to a certain extent.

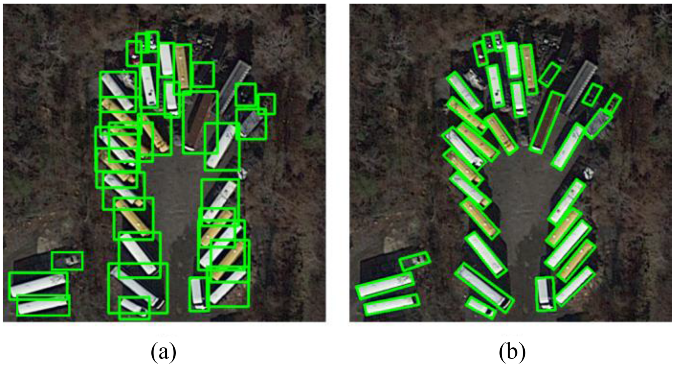


Fig. 1. (a) HBBs of objects with arbitrary orientations. (b) OBBs of objects with arbitrary orientations.

The object detection method based on neural network uses the smallest rectangular boxes that can contain the objects to locate the objects. Generally, the horizontal bounding boxes (HBBs) are quite good at representing the objects in natural images but not the ground objects in RSIs because the ground objects have arbitrary orientation in the overhead view used in RSIs. As illustrated in Fig. 1(a), HBBs representing rotated objects may contain a lot of undesirable contents such as a large amount of background for narrow objects with large aspect ratios and parts of other objects for densely distributed objects. For better localization of rotated objects, oriented bounding boxes (OBBs) are widely used in RSIs objects detection [9], [10], [11], [12], [13], [14], [15]. As can be seen in Fig. 1(b), the OBBs better enclose the objects themselves and has almost none of the problems described above in the horizontal boxes. The angle value is required as well as the position and side length of the box when defining an orientation box. There are various ways to represent the angle of object, the most common is to use the

Manuscript received 15 June 2022; revised 6 August 2022; accepted 14 September 2022. Date of publication 28 September 2022; date of current version 11 October 2022. This work was supported by the Chinese Academy of Sciences. *(Corresponding author: Yongcheng Wang.)*

Yuxi Zhang, Ning Zhang, Zheng Li, Zhikang Zhao, and Yunxiao Gao are with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: zhangyuxi18@mails.ucas.ac.cn; zhangning171@mails.ucas.ac.cn; lizheng20@mails.ucas.ac.cn; zhaozhikang20@mails.ucas.ac.cn; gaoyunxiao19@mails.ucas.ac.cn).

Yongcheng Wang, Dongdong Xu, and Guangli Ben are with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China (e-mail: wangyc@ciomp.ac.cn; xudongdong@ciomp.ac.cn; benguangli@ciomp.ac.cn).

Digital Object Identifier 10.1109/JSTARS.2022.3209349

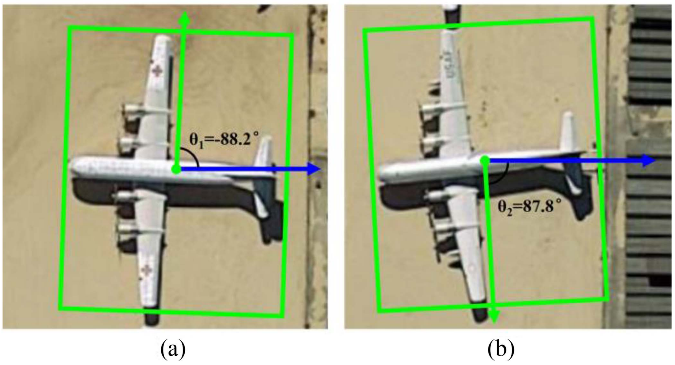


Fig. 2. Angle boundary discontinuity in rotated object detection. The angle between the long side of the bounding box and the positive *x*-axis of the image is taken as the angle of the box, which is limited to the interval [−90°, 90°). (a) OBB with an angle value of −88.2° near the lower boundary. (b) OBB with an angle value of 87.8° near the upper boundary. Two objects of (a) and (b) with similar orientations have very different angle values.

angle between one side of the bounding box (e.g., the long side) and the *x*-axis of the image as the angle value of that object. Based on the structure of the HBB object detector, the detection of rotated objects is achieved by adding the angle prediction module. Normally, angle prediction can be implemented by increasing a channel in the location regression module.

Although the accuracy and efficiency of detection are getting better as many methods with different network structures for rotated object detection are proposed, there are still several non-negligible problems in most rotated object detectors that have not been perfectly solved. List three challenges as follows.

1) In the anchor-based detectors, a large number of anchors with different angles are preset in the network in order to make the anchors match the rotated objects as much as possible, which causes a serious redundancy of the anchors and greatly increases the computational complexity.
2) The problem of discontinuity in the upper and lower boundaries of the angle values occurs when the angle is expressed in common rotated object detectors. The angle values near the two boundaries represent similar directions and have similar visual features on the image, while they are numerically jumpy and far apart, as sketched in Fig. 2. This makes the angle learning of anchors in the network somewhat confusing.
3) The structure of network has yet to be improved in terms of orientation-sensitive features extraction because the capability of extracting orientation-sensitive features is the key in rotated object detectors. In addition, the shape of the convolutional kernel in traditional CNN is generally horizontal and square, which also has a certain adverse impact on the extraction of orientation-sensitive features.

To deal with the abovementioned problems of rotated objects detection, we propose an orientation-first refinement detector based on orientation-first strategy in this article. The orientation-first strategy instructs the network to predict the orientation of the object first, and then preset the high-quality anchor based on the angle value. In this case, a large amount of redundancy in the initial anchor is avoided and the accuracy of the network for detecting rotated objects can be improved. An angle encoding regression module (AERM) is proposed in which the angle values are encoded as multiple parameters and the network predicts the object angle by learning multiple parameter values. The upper and lower boundaries of the angle values in this representation, such as −90° and 90°, correspond to the same encoded values, which solves the problem of discontinuity in the boundaries of the angle values. An angle channel attention module (ACAM) that uses the encoding parameters from the abovementioned angle representation method is also constructed in our network architecture. We use convolutional kernels with different angles in this module to extract features, and then utilize the abovementioned encoding parameters as weights to fuse multiple feature maps to generate a new feature map.

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

1) An orientation-first strategy for rotated object detection is proposed. This method avoids a large amount of redundancy in the preset anchor by predicting the object angle first, while the high-quality anchors improve the detection network for rotated objects.
2) We propose a new angle representation method that encodes the angle values into multiple parameters. This method can well solve the problem of discontinuous angle boundary and improve the learning ability of the network for object orientation.
3) An orientation feature extraction module based on multi-angle channel attention that fuses feature maps generated by different convolution kernels is introduced to more effectively extract orientation-sensitive features, enabling the detector to detect rotated objects more accurately.

The proposed rotated objects detection framework in this article achieves 79.56% and 96.29% accuracy in two challenging datasets, DOTA and HRSC2016, respectively.

II. Related Works

In recent years, object detection based on deep learning have gained great progress. Rectangular boxes are able to locate objects accurately and can be easily defined using a few parameters. While HBBs achieve excellent performance in most cases, the increased difficulty of object detection in specific environments has caused OBBs with arbitrary angles to be emphasized in research.

A. Object Detection Based on Deep Learning

The network architecture of object detection based on deep learning can be broadly classified into two categories, namely single-stage detector and two-stage detector, with the difference between the two types of detectors being whether the proposed regions are extracted. The two-stage detector will detect each proposed region separately after extracting the proposed regions, while the single-stage detector can detect all objects in an image end-to-end. Generally speaking, the two-stage detector has a higher accuracy rate, but its efficiency is reduced due to the extraction of proposed regions, whereas the single-stage detector has higher efficiency and lower accuracy rate. The R-CNN series [16], [17], [18] are representative algorithms of two-stage detectors, and plenty of improved algorithms based on Faster-RCNN [18] have emerged in recent years. SSD [19], YOLO series [20], [21], [22] are classic single-stage detection algorithms, among which YOLOv3 [22] has achieved brilliant results in various practical scenarios. In order to improve the accuracy of the single-stage detector, He et al. [23] proposed RetinaNet, which uses focal loss to deal with the problem of unbalanced positive and negative samples during the training of the network. This problem is considered to be an important reason why the accuracy of the single-stage detector is inferior to that of the two-stage detector. The most important role of the methods for extracting proposed region in the two-stage detector, such as RoI Pooling [17] and RoI Align [24], is to generate the refined feature maps corresponding to the proposed regions derived from the first stage, namely feature alignment. Inspired by this, Zhang et al. [25] proposed to use ordinary convolution operation to achieve feature alignment in a single-stage detector, and the potential is huge. With the introduction of the deformable convolution [26], [27], the method to achieve feature alignment is used by a variety of single-stage detectors and is called alignment convolution [28], [29], [30], [31], [32], [33]. The accuracy of single-stage detectors is gradually improved with the use of abovementioned methods.

Since anchor was proposed in Faster-RCNN, anchor-based algorithm have been extensively used in object detection because of its high accuracy. This type of algorithm has been fully developed and achieved great success in recent years, and it is still the mainstream method in the field of object detection. Anchors are rectangular boxes preset on the feature maps before the network detects the object. During the training process, the anchors are matched with the most similar bounding boxes of the ground-truth objects, and then corrected by the network to make the anchors as close as possible to the matched boxes. One problem of the anchor-based approach is that it is impossible for the anchors to match all the objects, especially the small objects with very few pixels. In order to match objects as much as possible, a lot of unnecessary anchors have to be preset, resulting in a waste of computational resources. In addition to the anchor-based approaches, the anchor-free models that does not require preset anchors are proposed and become popular [34], [35], [36], [37]. Law and Deng [34] proposed CornerNet based on corner point detection, and Duan et al. [36] proposed CenterNet based on center point detection. This type of method locates objects by detecting the key points, and then predicts other information at the positions of these key points. Subsequently, the application of anchor-free for object detection in RSIs has been extensively studied [38], [39], [40], [41], [42], [43].

B. Rotated Object Detection in RSIs

Unlike natural images, where objects are mostly arranged vertically under the effect of gravity, the objects on the ground may have arbitrary orientations in the overhead view of RSIs. If HBBs are used to represent these objects, the rectangular box will contain a lot of irrelevant content, especially when the aspect ratios of objects are large. In addition, the huge scale differences in RSIs, the complex and diverse earth surface, and the uneven objects distribution make detection more challenging. The abovementioned problem of HBB can seriously affect the accuracy of objects detection in RSIs. Therefore, more and more object detection methods use OBBs to locate objects in RSIs. In general, it is possible to convert from a horizontal box to an oriented box by simply adding one or few parameters to represent the orientation. The convenient conversion allows most classical object detection algorithms to detect rotated objects with minor modifications. At the same time, new methods and network structures have been proposed in order to further improve the accuracy.

The detection of angle is an important part in the rotated object detector and there is a difficult problem to deal with, i.e., the discontinuity of the angle boundary. This problem arises from the contradiction between the continuity of the directions and the discontinuity of the angle values. The DCL [44] and CSL [45] methods proposed by Yang et al. deal with this problem by converting the angle detection from a regression problem to a classification problem, however, the number of classifications limits the angle detection accuracy. The sliding vertex method proposed by Xu et al. [46] determined the oriented box by the minimum external horizontal box and the distance from the vertexes of the direction box to the vertexes of that horizontal box. Song et al. [47] designed a detector that first extracts proposals containing rotated objects and then predicts the endpoints of objects, avoiding the regression of angles. The AProNet [48] proposed by Zheng et al. determines the oriented box by the center point, the length and width of the object, and its mapping length in the horizontal and vertical directions. Besides the problem of angular boundary discontinuity, there is another problem in anchor-based rotated object detection, which is high difficulty of matching anchors with rotated objects. In [49], [50], [51], and [52], in order to match the rotated objects as much as possible, anchors with different angles are added at the same position. This method further increases the redundancy of the anchors, which greatly wastes computing resources. Zhong et al. [53] proposed an anchor matching method that matches a horizontal anchor to a horizontal box, which is obtained by decoupling the oriented box. Another solution is used in [54], [55], and [56], i.e., the horizontal anchor is still preset, but it is matched with the smallest outer HBBs of the ground-truth objects to increase the matching rate, and then let the anchor learn the oriented box in the subsequent network. In this article, the angle representation method of multiparameter regression is proposed, which has a periodicity consistent with the direction of object and fundamentally solves the problem of discontinuous angle boundaries. In this method, the angle is first detected before other information, and then the anchor with the angle is preset so that the objects can be matched more accurately with a small number of anchors.

III. Propose Method

In this section, we detailed the OFRDet based on the orientation-first strategy proposed in this article. First, each

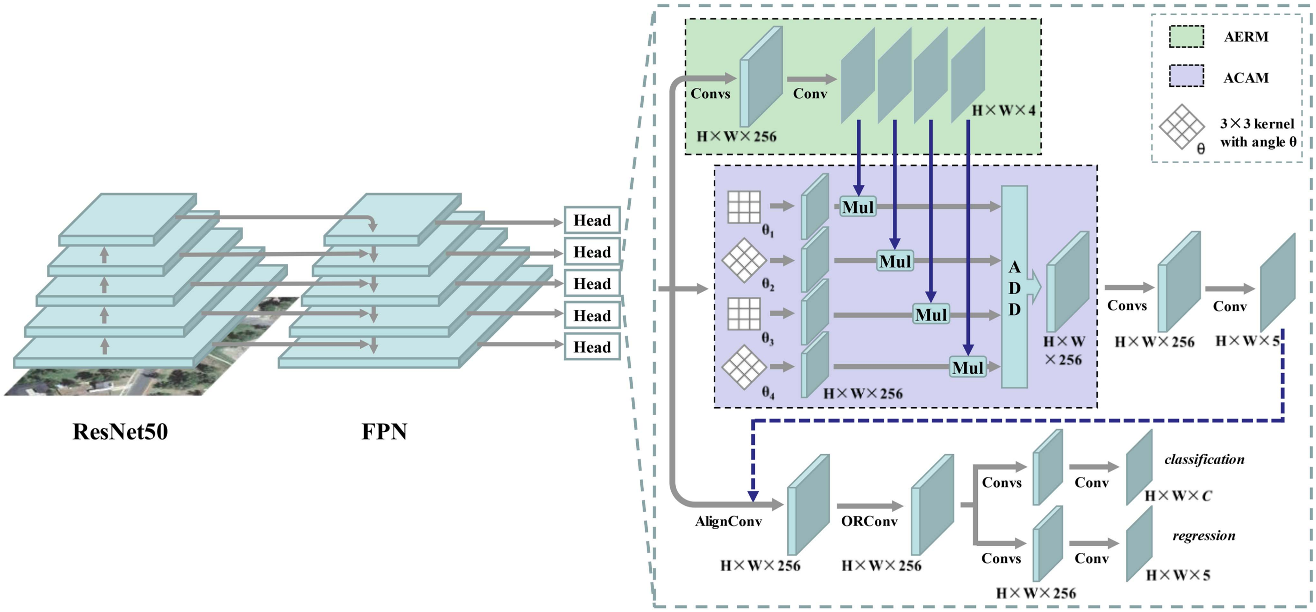


Fig. 3. Overall framework of the proposed OFRDet. OFRDet is a refinement detector with ResNet50-FPN as backbone to generate multiscale features. On each scale of the feature map we apply a detection head with orientation-first strategy to predict objects. AERM first predicts the orientation information to preset the oriented anchors. ACAM takes the outputs of AERM as attention maps and extracts directional features in multiple angle channels. The refined stage consists of AlignConv, ORConv, classification branch, and regression branch.

component of OFRDet and the overall implementation process are described in Section III-A. The overall network structure of the detector is shown in Fig. 3. Then, the baseline we adopted is introduced in Section III-B. Next, the designed AERM and ACAM is introduced in Sections III-C and III-D, respectively. Finally, our designed loss function of the overall network is shown in Section III-E.

A. Overall Design Structure of OFRDet

OFRDet is a refinement detector proposed in this article that can detect rotated objects in RSIs. In order to obtain the feature information of objects with large scale differences in RSIs and achieve correct detection, OFRDet uses ResNet50 [57] and feature pyramid network (FPN) [58] as the backbone to generate multiscale feature maps, and sets detection heads on the feature maps of multiple scales. In each detection head we adopt the orientation-first strategy, which makes the head first detect the direction information among all the information of objects. The strategy is to enable the network to preset anchor boxes with angles to better match objects, and to extract directional features based on the initial angle information to better regress bounding boxes. We design AERM to implement the priority detection of angle, which predicts the angle values of the objects at the positions of all feature points within the object bounding boxes. And uses the multiparameter angle encoding method to deal with the discontinuity of angle boundaries. Then, the encoded values obtained on each feature point is decoded into an angle value, and oriented anchor is preset on each feature point according to the angle value. The angle of the anchor is similar to the angle of the object to which the feature point belongs, so the anchor can better match the object and be further adjusted. There are two detection stages in the detection head, the coarse stage, and the refinement stage, which detect objects by adjusting the anchor boxes. In order to better distinguish and extract features in different directions, in the coarse stage, ACAM is designed to fuse feature maps of multiple angle channels to generate a new orientation-sensitive feature map, and the attention mechanism is adopted to give them different weights during fusion. Subsequently, the convolution operation is performed on the new feature map to complete the detection at this stage. The refined stage adjusts the detection results of the coarse stage through a series of convolution operations to obtain the final detection results.

B. Refined Detector Based on RetinaNet as Baseline

We add the refined stage to RetinaNet and use it as our baseline. ResNet50 and FPN are used as the backbone of the network to extract features. The residual module in ResNet well solves the problem of gradient disappearance in deep networks, so it can better extract deep features containing rich semantic information. FPN fuses deep and shallow features to generate multiscale feature maps, so that objects with different scales can be detected on the feature maps with the corresponding scales. Both the object classification branch and the bounding box regression branch of the detection head consist of ordinary full convolutional networks. In addition, focal loss is used as classification loss to solve the problem of imbalance of positive and negative samples during training.

In our baseline, the regression branch of detection head adds a channel to predict the angle θ. The OBB is represented by five parameters (x, y, w, h, θ), where (x, y) is the position of the center point of the bounding box in the image, w is the length of the long side, h is the length of the short side, $\theta \in [-\pi/2, \pi/2)$ represents the angle from the positive *x*-axis to the direction of the long side *w*.

In refined detector, the detection in the refined stage is adjusted bounding boxes according to the result of the coarse stage, so it is necessary to perform feature alignment according to the result of the coarse stage before the detection in the refined stage. We use AlignConv [33] to complete feature alignment. AlignConv calculates the offsets of the anchor adjustments in the coarse stage, applies the offsets to the convolution kernel, and uses the deformable convolution to perform the convolution operation on the feature map to achieve feature alignment. Furthermore, ORConv [59] is used after feature alignment. ORConv captures features in different directions by rotating the same convolution kernel *N* times (we set *N* to 8) and using them to perform convolution operations separately, with 1/*N* of the original number of channels in each direction and the total number of channels in the feature map unchanged.

C. Angle Encoding Regression Module

We propose a multiparameter encoding and decoding method that encodes an angle value into multiple regressable parameters (here, we use four parameters to introduce the method, and in the following, if not specifically stated, all four parameters are used as examples). The whole angular range *T* was divided into four intervals all bounded by θ_1, θ_2, θ_3, θ_4, and θ_5. Particularly, θ_5 and θ_1 are the upper and lower limits of the angular range, which differ by *T* and represent the same direction. In direction detection, θ and $\theta+nT$ (*n* is an arbitrary integer) represent the same direction due to periodicity. Considering this periodicity and subsequent decoding operations, the four parameters $x_{\theta 1}$, $x_{\theta 2}$, $x_{\theta 3}$, $x_{\theta 4}$ correspond to θ_1+nT, θ_2+nT, θ_3+nT, and θ_4+nT, respectively. When encoding an angle value θ, the interval in which the angle is located needs to be determined first, i.e., $\theta \in [\theta_a, \theta_b)$, where (a, b) ∈ {(1, 2), (2, 3), (3, 4), (4, 5)}. And then values of the two corresponding parameters $x_{\theta a}$ and $x_{\theta b}$ were determined according to the difference between θ and the two bounding angles θ_a and θ_b in that interval. The larger the angle difference, the smaller the corresponding parameter, and the sum of two parameters is 1. The parameters $x_{\theta a}$ and $x_{\theta b}$ is given by

$$x_{\theta_a} = \frac{|\theta - \theta_b|}{T/4}, x_{\theta_b} = \frac{|\theta - \theta_a|}{T/4}. \tag{1}$$

Finally, the other two parameters are set to 0, and the encoding of the angle θ is completed. The coding example is shown in Fig. 4, the interval $[-\pi/2, \pi/2)$ with the range *T* of π is divided into 4 parts, bounded by $-\pi/2$, $-\pi/4$, 0, $\pi/4$, and $\pi/2$. The angle θ to be encoded lies between $[-\pi/2, -\pi/4)$, and its difference from the boundary angle $-\pi/2$, $-\pi/4$ is α and β, respectively. Its encoding result is

$$E(\theta) = \left(\frac{\beta}{\pi/4}, \frac{\alpha}{\pi/4}, 0, 0\right). \tag{2}$$

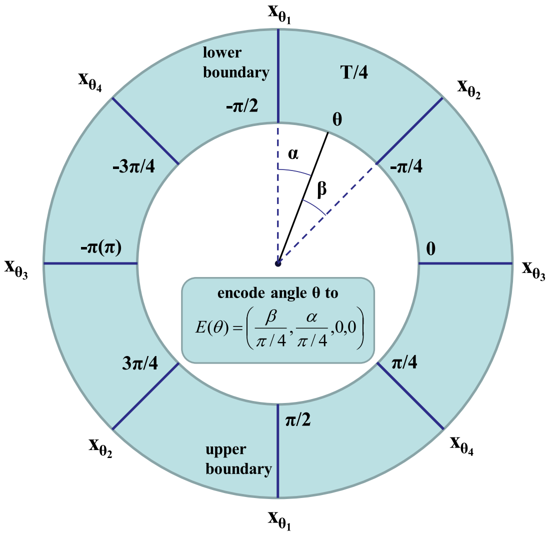


Fig. 4. Example of four-parameter angle coding method. The range of angle detection is $[-\pi/2, \pi/2)$, which is divided into four intervals, bounded by $-\pi/2$, $-\pi/4$, 0, $\pi/4$, and $\pi/2$. The angle θ is encoded as $E(\theta)$ consisting of four parameters $x_{\theta 1}$, $x_{\theta 2}$, $x_{\theta 3}$, and $x_{\theta 4}$, each corresponding to $-\pi/2+n\pi$, $-\pi/4+n\pi$, $0+n\pi$, $\pi/4+n\pi$.

Algorithm 1: angle decoding with four parameters.

```
Input: Parameters $x_1, x_2, x_3, x_4$ are acquired from
       regression module, $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$ are angle
       interval bounds, $T$ is the whole angular range of
       detection.
Output: $\theta$ is decoded angle value.
begin
  $i \leftarrow$ index of maximum $(x_1+x_2, x_2+x_3, x_3+x_4, x_4+x_1)$
  if $i$ is equal to 0 then $\theta$ is in interval $[\theta_1, \theta_2)$
      $\theta = x_4(\theta_4 - T) + x_1\theta_1 + x_2\theta_2 + x_3\theta_3$
  else if $i$ is equal to 1 then $\theta$ is in interval $[\theta_2, \theta_3)$
      $\theta = x_1\theta_1 + x_2\theta_2 + x_3\theta_3 + x_4\theta_4$
  else if $i$ is equal to 2 then $\theta$ is in interval $[\theta_3, \theta_4)$
      $\theta = x_2\theta_2 + x_3\theta_3 + x_4\theta_4 + x_1\theta_5$
  else if $i$ is equal to 2 then $\theta$ is in interval $[\theta_4, \theta_5)$
      $\theta = x_3\theta_3 + x_4\theta_4 + x_1\theta_5 + x_2(\theta_2 + T)$
  end if
  return $\theta$
end
```

Based on the abovementioned encoding principle, it can be inferred that the upper and lower limits of angle range θ_1 and θ_5 have the same encoding value, and the encoding values of angles slightly larger than θ_1 and slightly smaller than θ_5 are approximate and continuous at θ_1 and θ_5. From the above, it can be seen that the encoding value has the same continuity and periodicity as the direction to be detected, so the problem of discontinuity of the angle boundary is solved.

The multiparameter angle encoding method makes angle prediction a multiparameter regression problem. As shown in Fig. 5(a), we designed the AERM to predict angle values. The module consists of three convolutional layers and two activation layers following the first two convolutional layers. The input is a 256-channel feature map, and the output is a four-channel

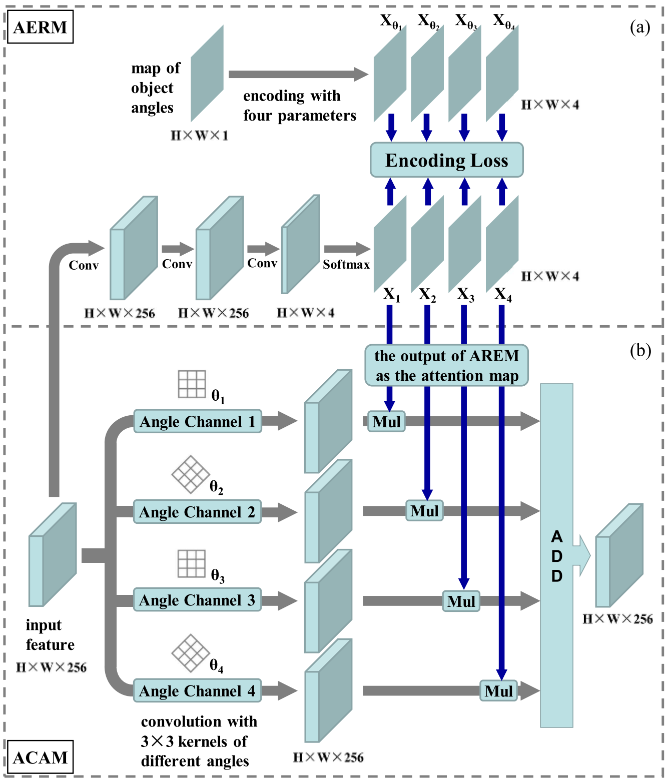


Fig. 5. Structure of AERM and ACAM. (a) AERM. It takes the 256-channel feature map as input and outputs a four-channel map after convolution and Softmax. The angle map after four-parameter encoding is used as the label to calculate loss with the output. (b) ACAM. It has four angle channels, each of which generates feature map using rotated kernels. The output is obtained by weighted summation of the four feature maps according to the attention maps gained from AERM.

angle encoding map that can be divided into four maps X_1, X_2, X_3, X_4. Each of the four maps corresponds to an encoding parameter. Since the sum of the four target encoded values is 1, the network performs a Softmax operation on these four channels for more efficient regression and decoding. The target encoding maps $X_{\theta 1}$, $X_{\theta 2}$, $X_{\theta 3}$, and $X_{\theta 4}$ is obtained from the angle map through four-parameter encoding. The value of each point on the angle map is the angle value of the object to which the point belongs (if a point does not belong to any object, then no loss is calculated at that point). After the prediction of the angles in this module, the angle values that need to be adjusted in the next coarse stage and refined stage are not randomly distributed in the entire detection range, but are clustered around 0°, and the angle discontinuity problem basically disappears. Therefore, the encoded values obtained by this module is decoded into an angle value and provided to the coarse stage for further adjustment.

It can be known from the abovementioned encoding method that θ_b is larger than θ_a by *T*/4, and then according to (1), θ can be obtained by

$$\theta = x_{\theta_a}\theta_a + x_{\theta_b}\theta_b. \tag{3}$$

Therefore, in order to decode the angle θ from the encoded value, it is necessary to determine, which interval the angle θ lies in. Ideally, only two or one of the encoded values are nonzero, and it is easy to find the corresponding angle interval. However, the four values are all nonzero in practice because of the deviation of the network regression results. From this, we design the decoding process in Algorithm 1. The adjacent two of the four coded values are added in turn, and the two with the largest sum are considered ideal nonzero values. Its corresponding angle interval can be subsequently obtained. To make the decoding more robust, four boundary angle values are obtained by expanding two intervals outward from this interval, and then weighting and summing them using their corresponding coding parameters. In this case, the four encoded values obtained by the network regression can all participate in the calculation of decoding the angle.

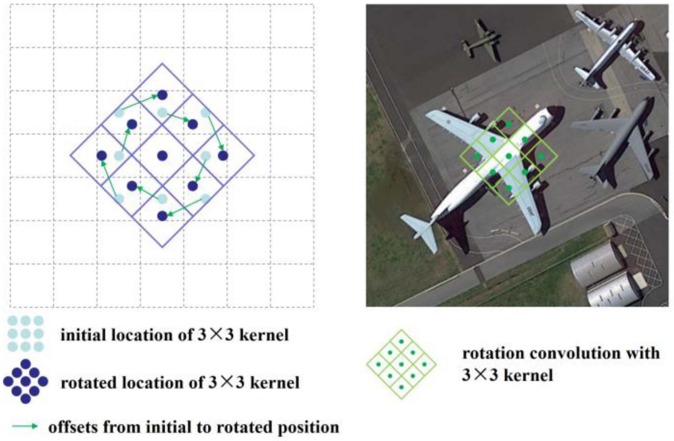
D. Angle Channel Attention Module

In the coarse stage, we designed the ACAM to extract orientation-sensitive features, whose structure is shown in Fig. 5(b). In this module, we design an attention mechanism based on the angle channel, while other attention mechanisms have gained special interest in the field of remote sensing in recent years [60], [61]. The input of this module is the feature map extracted by the backbone network, and the output is a new feature map with the same shape as the input. In ACAM, the four angle channels are designed to perform convolution operations using rotated kernels with angles θ_1, θ_2, θ_3, and θ_4 to extract features in corresponding directions, and generate four feature maps $f_{\theta 1}(x)$,$f_{\theta 2}(x)$,$f_{\theta 3}(x)$,$f_{\theta 4}(x)$, respectively. In oriented object detection, objects with arbitrary directions to be detected have various directional features. When detecting various objects, the degree of attention to the four angle channels should be different. From this, the angular channel attention mechanism is proposed, which assigns different weights to the four channels on each feature point, and obtains the final feature map after the weighted summation of the feature maps generated by the four angle channels. The output feature map $fo(x)$ can be given by

$$\begin{aligned} f_o(x) = {} & X_1 f_{\theta_1}(x) + X_2 f_{\theta_2}(x) \\ & + X_3 f_{\theta_3}(x) + X_4 f_{\theta_4}(x). \end{aligned} \tag{4}$$

In the above formula, X_1, X_2, X_3, and X_4 are the weight maps corresponding to the four angle channels, which are cascaded together to form an attention map with the shape of $h \times w \times 4$. The angle of the object has been predicted in AERM, so the four weight values on each feature point here can be determined by the angle of the object to which the point belongs. The closer the angle of the convolution kernel used by the channel is to the angle of the object, the greater the weight of the channel. And the sum of the four weights is 1. Ingeniously, the setting principle of the weight values is the same as the setting principle of the encoding parameters in AERM. Therefore, provided that the angles of the convolution kernels used by the four channels are the same as the first four boundary angles of the intervals in AERM, the four weights can correspond one-to-one with the four encoding parameters. In this case, as shown in Fig. 5, the output of AERM can be directly used as the attention map.

As illustrated in Fig. 6(a), the rotated convolution kernel is obtained by rotating the regular convolution kernel around the


initial location of 3×3 kernel
rotated location of 3×3 kernel
offsets from initial to rotated position
rotation convolution with 3×3 kernel

(a) (b)

Fig. 6. (a) Rotation convolution from regular convolution with kernel size 3 × 3. Since the rotation causes the change of convolution location, offsets are determined by the rotation angle. (b) Rotation convolution with kernel size 3 × 3 on a map. The kernel after adjusting the sampling position can specifically extract features in this direction.

center point by a certain angle. The offsets from the horizontal regular kernel to the rotated kernel can be calculated by the size of the kernel and the rotation angle value. As shown in Fig. 6(b), the convolution operation on the map using this rotated kernel can sufficiently extract features in a certain direction. And the rotated convolution with different angles in different angle channels can perform feature extraction in different directions. In addition, according to the angle of the preset anchors, the features in certain directions can be extracted more purposefully, and the generated feature maps are beneficial to the detection of rotated objects.

E. Loss Function

The loss of the whole network consists of three components, including the angle encoding regression loss, the coarse stage detection loss, and the refinement stage detection loss. The loss function is defined as

$$\begin{aligned} L = & \frac{\lambda_1}{N_E}\sum_m\sum_n L_r\left(p^E_{mn}, p^*_{mn}\right) \\ & + \frac{\lambda_2}{N_C}\left(\sum_i L_c\left(c^C_i, l^*_i\right) + \sum_i [l^*_i \geq 1] L_r\left(x^C_i, g^*_i\right)\right) \\ & + \frac{\lambda_3}{N_R}\left(\sum_i L_c\left(c^R_i, l^*_i\right) + \sum_i [l^*_i \geq 1] L_r\left(x^R_i, g^*_i\right)\right). \end{aligned} \tag{5}$$

In the first term of (5), the angle encoding regression loss, λ_1 is the balance coefficient, N_E is the total number of encoded values that have a target, *m* represents each feature point that has target encoded values, *n* represents each encoding parameter on a feature point, L_r is the regression loss, where smoothed $L1$ loss is used, *pE mn* represents the encoded value obtained by the network, and p^*_{mn} represents the target encoded value. The detection loss consists of object classification loss and bounding box regression loss, where the bounding box regression loss is obtained from positive samples only. In the second and third terms of (5), λ_2 and λ_3 are the balance coefficients, N_C and N_R are the number of positive samples in the coarse and refined stages, respectively, *i* represents each sample, L_c is the classification loss, where focal loss is used, *cC i* and *cR i* are the category predictions for sample *i* in the two stages, l^*_i is the ground-truth label of that, $[l^*_i{>}1]$ is the Iverson bracket indicating equation, i.e., the value is 1 when *i* is a positive sample, *xC i* and *xR i* are the location predictions for sample *i* in the two stages and g^*_i is the ground truth of that.

IV. Experiments and Analysis

A. Data Sets

*1) DOTA-*v*1.0 [1]:* This is a large-scale aerial remote sensing dataset made for object detection, which contains 2806 aerial images collected from satellites such as Google Earth, satellite JL-1, and 188282 ground objects on them. All the objects are grouped into 15 common categories, which are 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). Instances in this dataset are annotated with HBBs and OBBs, and we use the OBB annotation in it for experiments. The entire dataset is randomly divided into three parts, where 1/2 is used as the training set, 1/6 as the validation set, and 1/3 as the test set.

The image sizes in DOTA vary widely, ranging in size from 800 × 800 to 4000 × 4000 pixels. We crop the original image into a series of 1024 × 1024 patches with a stride of 824. In the experiments with multiscale data augmentation, the original images were resized using three scales (0.5, 1.0, and 1.5) and change the cropping step size to 512. If instances are segmented during cropping, we decide whether to adopt them or not according to the method in [1]. In testing, the cropped images were fed into the network for detection and merge the results into the original size image.

2) HRSC2016 [62]: This is a high-resolution image dataset for ship detection containing arbitrarily oriented ships from open sea or coast side. The images in the dataset are collected from Google Earth with resolutions ranging from 2 to 0.4 m and image sizes ranging from 300 × 300 to 1500 × 900 pixels. There are 1061 images in the dataset, including 436 images in the training set, 181 images in the validation set, and 444 images in the test set. We use the OBB annotations in the dataset for experiments. And all the images are resized to the range (512, 800) without changing their aspect ratio, i.e., each image has a short side of 512 pixels and a long side of up to 800 pixels.

B. Implementation Details

This article uses ResNet50 FPN as the backbone network in the following experiments. The ResNet50 is initialized using the parameters pretrained on ImageNet. In the pyramidal feature

TABLE I
RESULTS OF ABLATION EXPERIMENTS FOR ACAM, AERM IN OFRDET ON DOTA DATASET

baseline	OFRDet		PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
	ACAM	AERM																
√			90.15	81.54	55.86	81.53	70.98	86.70	90.08	90.77	73.91	89.47	74.52	70.05	77.08	68.23	63.93	77.65
√	√		90.35	84.01	54.81	83.87	72.97	86.53	90.15	90.75	74.18	89.42	76.63	71.26	77.76	70.70	66.63	78.67
√	√	√	90.28	85.39	55.30	84.31	70.33	86.73	90.18	90.70	74.58	89.20	81.63	68.72	77.96	69.01	79.15	79.56

maps generated by FPN, (P3, P4, P5, P6, P7) are selected to preset the anchors of different scales. An anchor box with an aspect ratio of 1:1 is set on each feature point, whose side length is four times the stride of the feature map (i.e., 32, 64, 128, 256, 512) and whose angle is determined by the network prediction. In the loss function, the balance parameter of the angle encoded regression loss is set to 0.1, and other balance parameters are set to 1. The hyperparameters α and γ in Focal loss are set to 0.25 and 2.0, respectively. For the matching strategy, the Intersection over Union (IoU) threshold of foreground and background are set as 0.5 and 0.4 in both the coarse stage and the refined stage. In the training phase, a single NVIDIA 3080Ti GPU is used for the experiments with the batch size set to 4. SGD optimizer is used to update the parameters of the model, in which the initial learning rate is set to 0.005, the learning rate is reduced to 1/10 of the previous one each time it decays, and the momentum and weight decays are set to 0.9 and 0.0001, respectively. When using the DOTA dataset, the network is trained with 18 epoches, compared to 36 epoches when using the HRSC2016 dataset. To prevent overfitting, we use horizontal flipping to increase the complexity of the dataset, and we also use zero-padding random rotation and multiscale data augmentation when employing a data enhancement strategy. In the testing phase, we also use a single 3080Ti GPU for inference. We keep bounding boxes with classification scores greater than 0.05, and set the IOU threshold in rotated nonmaximum suppression to 0.1. At the same time, considering that an image contains a limited number of objects, we set the upper limit of the number of objects in each image to 2000.

C. Ablation Studies

We conduct ablation experiments on the DOTA dataset to verify the effectiveness of our method, using mAP as a criterion for evaluating method performance. To compare the best results achieved by various architectures, all ablation experiments below are performed using the data augmentation strategy described in Section IV-B.

1) Baseline: As a classical object detection network, RetinaNet can fit the detection tasks in most scenarios and achieve good results. In our baseline, the refined stage is added to RetinaNet to pursue better results. We add an angle prediction channel to the regression branch of the detection head so that it can be used to detect OBBs. The training and test parameters of each part in the baseline are exactly the same as the parameters of the network structure in other ablation experiments that follow. In the refined stage, we use AlignConv to realign the feature map. When using an oriented box to locate an object, the object can mostly occupy a higher proportion inside the box compared to using a horizontal box. Therefore, feature alignment within the oriented boxes can play an important role and significantly promote the feature representation of the object in the box. From the detection results, the network based on RetinaNet with the addition of refined stage has good performance in rotated object detection. As shown in the first row of Table I, the mAP of baseline network is 77.65% for 15 types of objects on the DOTA dataset.

2) Effectiveness of ACAM in the Orientation-First Strategy: To evaluate the effectiveness of the ACAM in OFRDet, the experiment is conducted by adding this module in the coarse stage based on the baseline method. At the same time, in order to make the attention mechanism work by obtaining effective angle channel weights and not let AERM affect our judgment on the test results, we change the AERM that implements the priority angle detection to the angle detection method in the baseline, i.e., the conventional angle value regression. As shown in the second row of Table I, with these settings, the mAP of the detector for 15 categories of rotated objects in the DOTA dataset is 78.67%, which is 1.02% higher than the baseline method. This enhancement is due to the fact that the ACAM is able to extract directional features more purposefully under the guidance of the angles prior predicted, and the oriented anchors preset by the orientation-first strategy can be more easily adjusted to the ground bounding boxes. In addition, the detection results of different categories of objects are presented in Table I, and it can be found that the APs of categories such as GTF, SV, SBF, HC have been increased significantly. These objects have various features in different directions, and this performance indicates that ACAM is more effective in extracting directional features.

3) Effectiveness of AERM: The complete OFRDet is used in this ablation experiment to evaluate the effectiveness of AERM. Compared with the previous experimental settings to verify the effectiveness of the ACAM, OFRDet only changes the part that implements the priority detection of angles to AERM, and the other parts remain unchanged. As shown in the third row of Table I, the mAP obtained by OFRDet is 79.56%, which is about 0.89% higher than the previous experimental result without AERM, and about 1.91% higher than the baseline method, with a significant improvement. The main reasons for this enhancement are, first, that the angle representation method using multiparameter encoding is easier to be learned by the network when performing orientation-first detection, and second, the multiparameter encoding method fundamentally eliminates the boundary discontinuity problem of angle regression. The qualitative visualization detection results of the two networks for rotated objects with angle values close to the upper and lower limit are shown in Fig. 7. It can be seen that the performance of OFRDet using AERM is significantly better, and the bounding boxes in the detection results have smaller

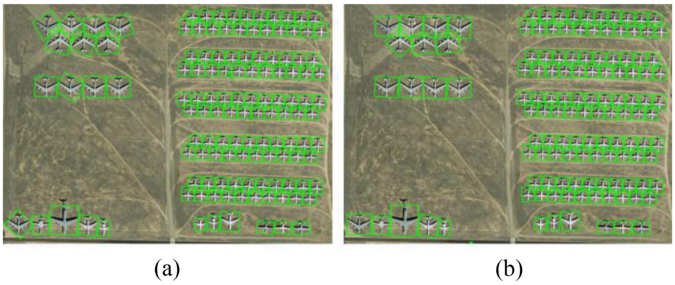


Fig. 7. Visual detection examples of plane on DOTA. (a) Detection results of OFRDet without AERM. (b) Detection results of OFRDet with AERM.

TABLE II
EXPERIMENTS WITH DIFFERENT MODULE SETTINGS

channel/parameter	3	4	6
mAP	79.22%	79.56%	79.18%

errors and more suitable orientation, which confirms that AERM effectively handles the boundary discontinuity problem of angle regression.

4) Setting of the Number of Angle Channels and Encoding Parameters: In OFRDet, as described in Section III-D, the multiple parameters obtained by AERM to encode the angle values can be directly used as weights for angle channels in ACAM. There is a one-to-one correspondence between angle channels and encoding parameters, and the number of these is the same. In the abovementioned ablation experiments, the number of angle channels and angle encoding parameters were set to 4. The influence of the number of angle channels and angle encoding parameters on the detection results is explored in the following experiments. In addition to four-channel four-parameter, three-channel three-parameter, and six-channel six-parameter are also set in the network for experimentation. The experimental results are shown in Table II, where the mAPs under the three settings are 79.22%, 79.56%, and 79.18%, respectively. These results show that the effect of the number of channels and parameters is relatively small compared to the enhancement effect of the ACAM and the AERM on the detection results. Moreover, the four-channel four-parameter has a better detection effect than the three-channel three-parameter and six-channel six-parameter, which shows that when extracting directional features through ACAM, the moderate angle interval can maximize its extraction ability. If the angle interval is too large, the extracted directional features are incomplete, and if it is too small, the extracted directional features are redundant.

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

In this section, we compare the proposed OFRDet with other state-of-the-art detection methods on two datasets, namely DOTA and HRSC2016. Their introduction and experimental details are in Sections IV-A and IV-B, respectively.

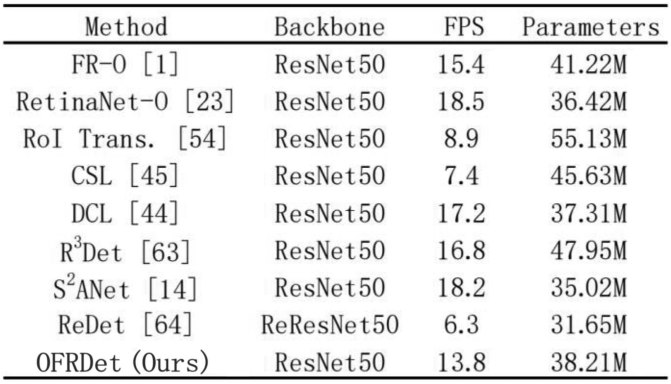
1) Complexity and Speed Comparison: We compare our method with other methods in terms of speed and complexity, and the comparison results are shown in Table III. The comparative experiments are carried out on the DOTA dataset, and the cropped image patches of size 1024 × 1024 are detected.

We reflect the speed and complexity of the detector by the number of frames per second (FPS) and the amount of model parameters, respectively. The FPS shown here is average FPS obtained after detecting the entire validation set of 5297 split images. For fairness, all the methods are inferred with batch size of 1 on a single RTX 3080Ti. The detection speed of our method is 13.8 FPS and the model size is 38.21M. It can be seen from Table III that both the detection speed and the model size of our method are in the middle level among the compared methods.

2) Results on DOTA: On the DOTA dataset, we compare with a variety of advanced or classical methods at single-scale or multiscale, and the results are shown in Table IV. Among these methods, FR-O and RetinaNet-O are implemented by adding angle prediction channels in the bounding box regression branch of the classical computer vision algorithms Faster-RCNN [18] and RetinaNet [23], respectively. Other methods are specially proposed to detect rotating objects in remote sensing images. CAD-Net [10] learns global and local contextual information of objects by computing their correlations with the global scene and local adjacent features. DAL [13] is a dynamic anchor learning method that uses a new matching mechanism to evaluate anchors and assign them more efficient labels. S^2A-Net [14] uses a new alignment convolution, which can adaptively align convolution features according to anchors. FoRDet [15] leverages the information of foreground regions from the perspectives of feature and optimization. Different from compared methods, our method proposes a new multiparameter angle coding and angle channel attention mechanism to enhance the angle regression and direction feature extraction of the network, so as to improve the detection ability of rotated objects. Our proposed OFRDet achieves a mAP of 74.19% on the single-scale dataset and 79.56% on the multiscale dataset. We achieve state-of-the-art results in 7/15 categories among the methods of comparison, and it is worth noting that our results have a large lead in the detection of GTF, SBF, and HC. The directionality of these classes of objects is obvious, indicating that our detector has a strong ability in direction detection. The qualitative visual test results produced by our method in detecting some images of the DOTA dataset are shown in Fig. 8. Although we preset only one anchor on each feature point, we finally obtained excellent detection boxes, which can closely surround objects, even for objects with large scale differences or densely arranged, which shows the effectiveness of the preset rotated anchor. For objects with arbitrary orientations in complex environments, our method can assign bounding boxes a suitable orientation with fewer errors to complete the detection.

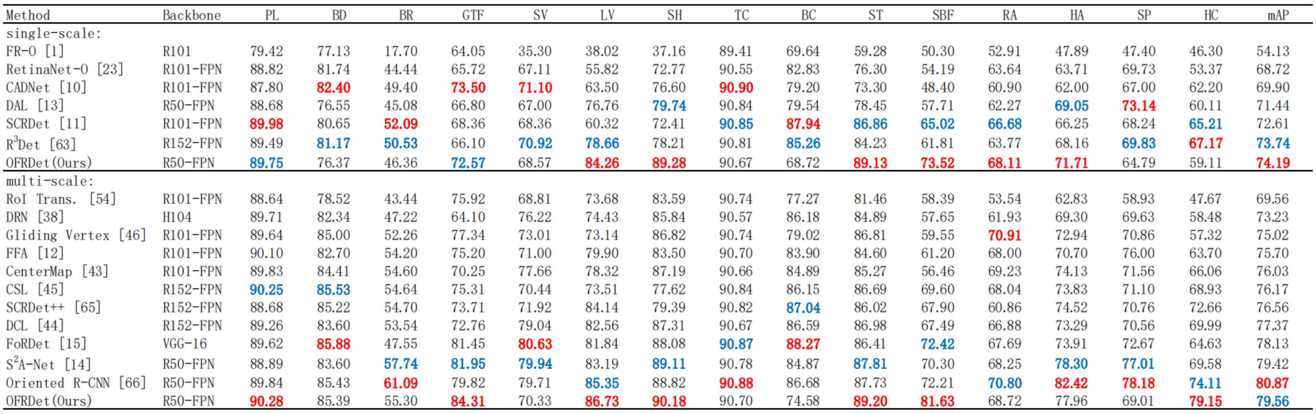
3) Results on HRSC2016: OFRDet is compared with various methods on HRSC2016 dataset, and the comparison results are shown in Table V. Among these methods, R2CNN [67], Rotated RPN [68], and SBD [69] are proposed in the field of computer vision to detect slanted text with angles. Other methods are proposed to detect rotated objects in RSIs. It is worth noting that we use the PASCAL VOC2012 metric to calculate mAP for

TABLE III
COMPARISON OF SPEED AND COMPLEXITY OF DIFFERENT METHODS



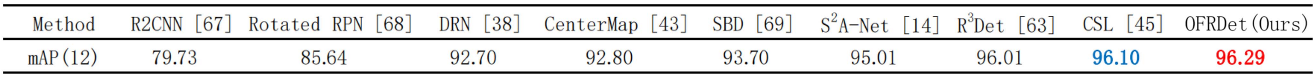
Method	Backbone	FPS	Parameters
FR-O [1]	ResNet50	15.4	41.22M
RetinaNet-O [23]	ResNet50	18.5	36.42M
RoI Trans. [54]	ResNet50	8.9	55.13M
CSL [45]	ResNet50	7.4	45.63M
DCL [44]	ResNet50	17.2	37.31M
R^3Det [63]	ResNet50	16.8	47.95M
S^2ANet [14]	ResNet50	18.2	35.02M
ReDet [64]	ReResNet50	6.3	31.65M
OFRDet (Ours)	ResNet50	13.8	38.21M

TABLE IV
COMPARISON WITH STATE-OF-THE-ART METHODS ON DOTA DATASET. R50-FPN REPRESENTS RESNET 101 WITH FPN (LIKEWISE R101-FPN, R152-FPN), H-104 REPRESENTS HOURGLASS 104, AND VGG-16 REPRESENTS VGGNET 16. THE RESULTS MARKED IN RED AND BLUE ARE THE BEST AND SECOND BEST IN EACH COLUMN, RESPECTIVELY



Method	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
single-scale:																	
FR-O [1]	R101	79.42	77.13	17.70	64.05	35.30	38.02	37.16	89.41	69.64	59.28	50.30	52.91	47.89	47.40	46.30	54.13
RetinaNet-O [23]	R101-FPN	88.82	81.74	44.44	65.72	67.11	55.82	72.77	90.55	82.83	76.30	54.19	63.64	63.71	69.73	53.37	68.72
CADNet [10]	R101-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
DAL [13]	R50-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
SCRDet [11]	R101-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 [63]	R152-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**
OFRDet (Ours)	R50-FPN	**89.75**	76.37	46.36	**72.57**	68.57	**84.26**	**89.28**	90.67	68.72	**89.13**	**73.52**	**68.11**	**71.71**	64.79	59.11	**74.19**
multi-scale:																	
RoI Trans. [54]	R101-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
DRN [38]	H104	89.71	82.34	47.22	64.10	76.22	74.43	85.84	90.57	86.18	84.89	57.65	61.93	69.30	69.63	58.48	73.23
Gliding Vertex [46]	R101-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
FFA [12]	R101-FPN	90.10	82.70	54.20	75.20	71.00	79.90	83.50	90.70	83.90	84.60	61.20	68.00	70.70	76.00	63.70	75.70
CenterMap [43]	R101-FPN	89.83	84.41	54.60	70.25	77.66	78.32	87.19	90.66	84.89	85.27	56.46	69.23	74.13	71.56	66.06	76.03
CSL [45]	R152-FPN	**90.25**	**85.53**	54.64	75.31	70.44	73.51	77.62	90.84	86.15	86.69	69.60	68.04	73.83	71.10	68.93	76.17
SCRDet++ [65]	R152-FPN	88.68	85.22	54.70	73.71	71.92	84.14	79.39	90.82	**87.04**	86.02	67.90	60.86	74.52	70.76	72.66	76.56
DCL [44]	R152-FPN	89.26	83.60	53.54	72.76	79.04	82.56	87.31	90.67	86.59	86.98	67.49	66.88	73.29	70.56	69.99	77.37
FoRDet [15]	VGG-16	89.62	**85.88**	47.55	81.45	**80.63**	81.84	88.08	**90.87**	**88.27**	86.41	**72.42**	67.69	73.91	72.67	64.63	78.13
S^2A-Net [14]	R50-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
Oriented R-CNN [66]	R50-FPN	89.84	85.43	**61.09**	79.82	79.71	**85.35**	88.82	**90.88**	86.68	87.73	72.21	**70.80**	**82.42**	**78.18**	**74.11**	**80.87**
OFRDet (Ours)	R50-FPN	**90.28**	85.39	55.30	**84.31**	70.33	**86.73**	**90.18**	90.70	74.58	**89.20**	**81.63**	68.72	77.96	69.01	**79.15**	**79.56**

TABLE V
COMPARISON WITH STATE-OF-THE-ART METHODS ON HRSC2016 DATASET. (12) MEANS THAT PASCAL VOC2012 EVALUATION METRIC IS USED TO CALCULATE THE RESULT



Method	R2CNN [67]	Rotated RPN [68]	DRN [38]	CenterMap [43]	SBD [69]	S^2A-Net [14]	R^3Det [63]	CSL [45]	OFRDet (Ours)
mAP (12)	79.73	85.64	92.70	92.80	93.70	95.01	96.01	**96.10**	**96.29**

the detection results, and the mAPs of other methods compared are also calculated under this metric. OFRDet achieves 96.29% mAP on HRSC2016 dataset, outperforming all other methods compared. OFRDet only use one scale of oriented anchor to achieve the current results. Part of the visual detection results of OFRDet on the HRSC2016 dataset are shown in the Fig. 9, from which it can be seen that OFRDet can always give a suitable OBB to tightly enclose ships with arbitrary orientations, although some ships have the characteristics of large differences in scale and dense arrangement. Even in the different environments such as harbor, coast, and sea, the method can complete the detection with high quality.

E. Limitations of the Method

Although OFRDet has obtained competitive experimental results, it still has limitations in object detection that cannot be ignored. As shown in the Fig. 10(a), OFRDet can correctly detect the large vehicle and the small vehicle in most cases, however, in the case where the visual features of the two are very similar, the detection results will be misclassified. In addition, the feature information of objects is insufficient when they are extremely small, and then the objects are immersed in the background, resulting in the failure to be detected, which can be seen from the Fig. 10(b). The abovementioned problems are extremely

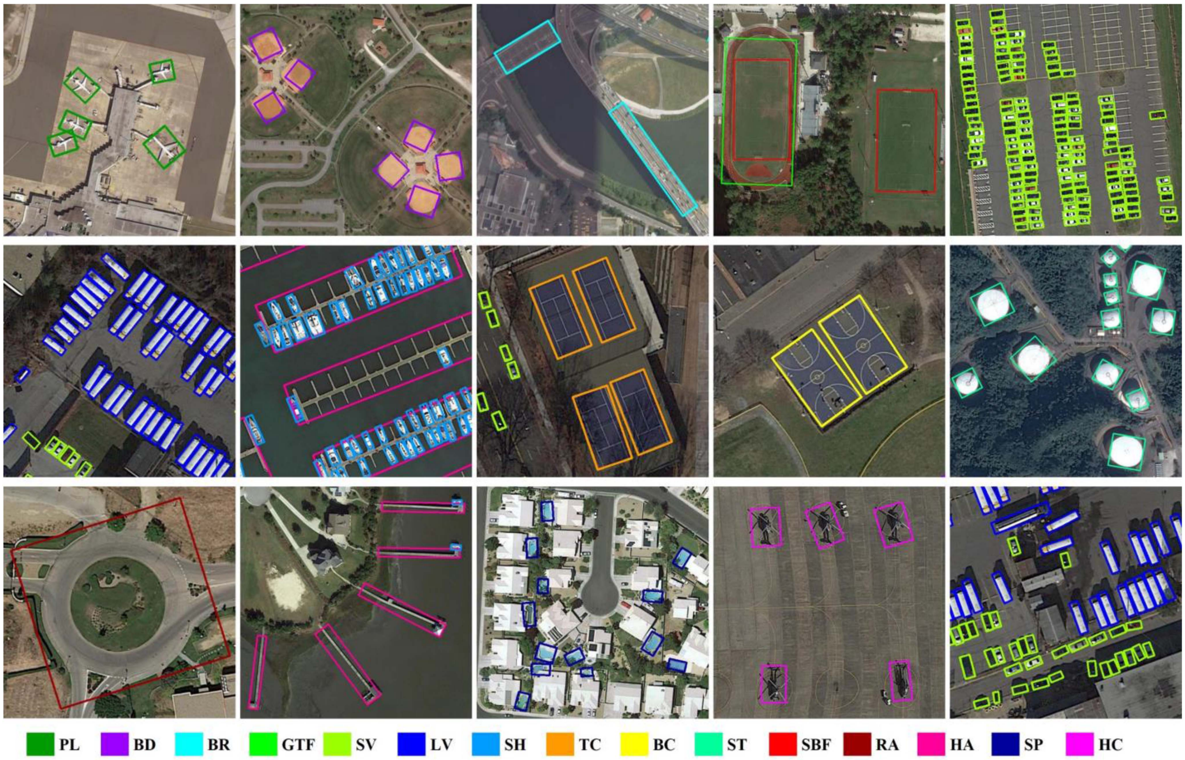


Fig. 8. Some visualization results from our proposed OFRDet on DOTA. The confidence threshold is set to 0.3. One color represents one object category.

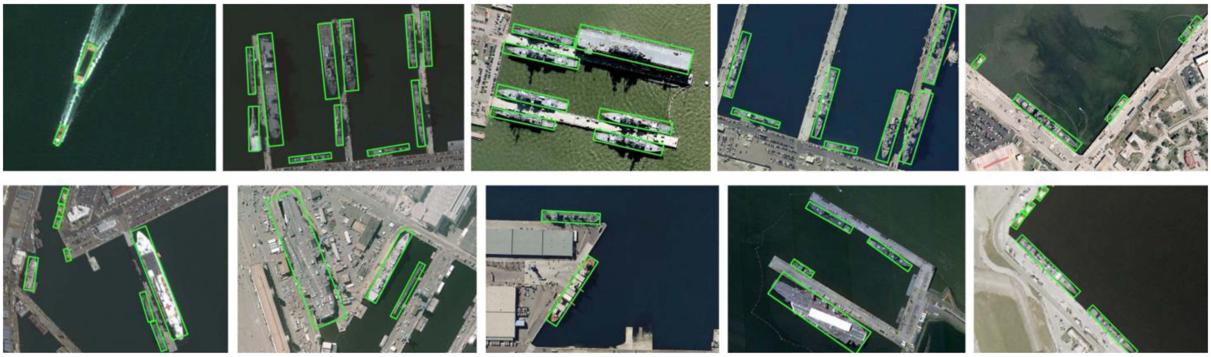
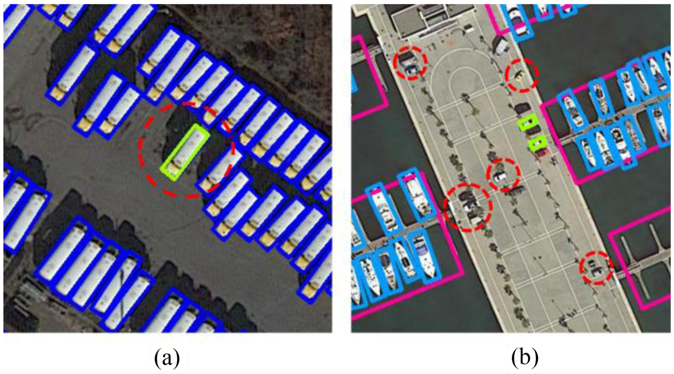


Fig. 9. Some visualization results from our proposed OFRDet on HRSC2016. The confidence threshold is set to 0.3.



(a)

(b)

Fig. 10. (a) Classification error of object detection. The blue box represents the detection results of LV, and the green box represents SV. Inside the red circle is a large vehicle that was mistakenly classified as a small vehicle. (b) Missing detection of tiny objects. The red circles are undetected small vehicles.

challenging in RSIs detection and test the feature extraction and discrimination ability of the detection network, which requires further specialized research.

V. Conclusion

This article provides a strategy for the detection of rotated objects in RSIs, namely orientation-first strategy, and OFRDet is proposed based on this strategy. In OFRDet, the ACAM is proposed to extract the orientation features of objects more accurately, thereby improving the regression accuracy of OBB. The AERM is proposed to solve the problem of discontinuous boundary in angle prediction, so as to obtain more accurate angle information of objects. We demonstrate the effectiveness of our proposed method on the DOTA and HRSC2016 datasets through extensive experiments.

REFERENCES

[1] 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.*, 2018, pp. 3974–3983, doi: 10.1109/CVPR.2018.00418.

[2] 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. 2018, doi: 10.1109/TGRS.2017.2778300.

[3] W. Liu, L. Ma, J. Wang, and H. Chen, "Detection of multiclass objects in optical remote sensing images," *IEEE Geosci. Remote Sens. Lett.,* vol. 16, no. 5, pp. 791–795, May 2019, doi: 10.1109/LGRS.2018.2882778.

[4] Y. Gong et al., "Context-aware convolutional neural network for object detection in VHR remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.,* vol. 58, no. 1, pp. 34–44, Jan. 2020, doi: 10.1109/TGRS.2019.2930246.

[5] J. Lei, X. Luo, L. Fang, M. Wang, and Y. Gu, "Region-enhanced convolutional neural network for object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.,* vol. 58, no. 8, pp. 5693–5702, Aug. 2020, doi: 10.1109/TGRS.2020.2968802.

[6] B. Cheng, Z. Li, B. Xu, X. Yao, Z. Ding, and T. Qin, "Structured object-level relational reasoning CNN-based target detection algorithm in a remote sensing image," *Remote Sens.,* vol. 13, no. 2, p. 281, 2021, doi: 10.3390/rs13020281.

[7] W. Yin, X. Sun, W. Diao, Y. Zhang, and X. Gao, "Thermal power plant detection in remote sensing images with saliency enhanced feature representation," *IEEE Access,* vol. 9, pp. 8249–8260, 2021, doi: 10.1109/ACCESS.2021.3049431.

[8] K. Zhou, Z. Zhang, C. Gao, and J. Liu, "Rotated feature network for multiorientation object detection of remote-sensing images," *IEEE Geosci. Remote Sens. Lett.,* vol. 18, no. 1, pp. 33–37, Jan. 2021, doi: 10.1109/LGRS.2020.2965629.

[9] M. Sharma et al., "YOLOrs: Object detection in multimodal remote sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.,* vol. 14, pp. 1497–1508, Nov. 2021, doi: 10.1109/JSTARS.2020.3041316.

[10] 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, doi: 10.1109/TGRS.2019.2930982.

[11] X. Yang et al., "SCRDet: Towards more robust detection for small, cluttered and rotated objects," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8231–8240, doi: 10.1109/ICCV.2019.00832.

[12] K. Fu, Z. Chang, Y. Zhang, G. Xu, K. Zhang, and X. Sun, "Rotation-aware and multi-scale convolutional neural network for object detection in remote sensing images," *ISPRS J. Photogrammetry Remote Sens.,* vol. 161, pp. 294–308, Mar. 2020, doi: 10.1016/j.isprsjprs.2020.01.025.

[13] 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.

[14] J. Han, J. Ding, J. Li, and G. S. Xia, "Align deep features for oriented object detection," *IEEE Trans. Geosci. Remote Sens.,* vol. 60, Mar. 2022, Art. no. 5602511, doi: 10.1109/TGRS.2021.3062048.

[15] T. Zhang et al., "Foreground refinement network for rotated object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.,* vol. 60, Sep. 2022, Art. no. 5610013, doi: 10.1109/TGRS.2021.3109145.

[16] 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.*, 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[17] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[18] 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, doi: 10.1109/TPAMI.2016.2577031.

[19] W. Liu et al., "SSD: Single shot multibox detector," in *Computer Vision – ECCV*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 21–37.

[20] 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.*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[21] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[22] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018, *arXiv:1804.02767*.

[23] 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.*, 2017, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.

[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988, doi: 10.1109/ICCV.2017.322.

[25] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4203–4212, doi: 10.1109/CVPR.2018.00442.

[26] J. Dai et al., "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 764–773, doi: 10.1109/ICCV.2017.89.

[27] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets V2: More deformable, better results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9300–9308, doi: 10.1109/CVPR.2019.00953.

[28] X. Chen, J. Yu, S. Kong, Z. Wu, and L. Wen, "Joint anchor-feature refinement for real-time accurate object detection in images and videos," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 31, no. 2, pp. 594–607, Feb. 2021, doi: 10.1109/TCSVT.2020.2980876.

[29] Y. Chen, C. Han, N. Wang, and Z. Zhang, "Revisiting feature alignment for one-stage object detection," 2019, *arXiv:1908.01570*.

[30] H. D. Jang, S. Woo, P. Benz, J. Park, and I. S. Kweon, "Propose-and-attend single shot detector," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2020, pp. 804–813, doi: 10.1109/WACV45572.2020.9093364.

[31] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, "Region proposal by guided anchoring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2960–2969, doi: 10.1109/CVPR.2019.00308.

[32] 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.*, 2019, pp. 9656–9665, doi: 10.1109/ICCV.2019.00975.

[33] S. Zhang, L. Wen, Z. Lei, and S. Z. Li, "RefineDet++: Single-shot refinement neural network for object detection," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 31, no. 2, pp. 674–687, Feb. 2021, doi: 10.1109/TCSVT.2020.2986402.

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

[35] X. Zhou, J. Zhuo, and P. Krähenbühl, "Bottom-up object detection by grouping extreme and center points," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 850–859, doi: 10.1109/CVPR.2019.00094.

[36] 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.*, 2019, pp. 6568–6577, doi: 10.1109/ICCV.2019.00667.

[37] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9626–9635, doi: 10.1109/ICCV.2019.00972.

[38] X. Pan et al., "Dynamic refinement network for oriented and densely packed object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11204–11213, doi: 10.1109/CVPR42600.2020.01122.

[39] H. Wei, Y. Zhang, Z. Chang, H. Li, H. Wang, and X. Sun, "Oriented objects as pairs of middle lines," *ISPRS J. Photogrammetry Remote Sens.,* vol. 169, pp. 268–279, Nov. 2020, doi: https://doi.org/10.1016/j.isprsjprs.2020.09.022.

[40] 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.*, 2021, pp. 8788–8797, doi: 10.1109/CVPR46437.2021.00868.

[41] W. Li and J. Zhu, "Oriented reppoints for aerial object detection," 2021, *arXiv:2105.11111*.

[42] B. Kim, J. Lee, S. Lee, D. Kim, and J. Kim, "TricubeNet: 2D kernel-based object representation for weakly-occluded oriented object detection," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 3421–3430, doi: 10.1109/WACV51458.2022.00348.

[43] 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, doi: 10.1109/TGRS.2020.3010051.

[44] 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.*, 2021, pp. 15814–15824, doi: 10.1109/CVPR46437.2021.01556.

[45] X. Yang and J. Yan, “Arbitrary-oriented object detection with circular smooth label,” in *Computer Vision – ECCV*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham, Switzerland: Springer, 2020, pp. 677–694.

[46] 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, doi: 10.1109/TPAMI.2020.2974745.

[47] Q. Song, F. Yang, L. Yang, C. Liu, M. Hu, and L. Xia, “Learning point-guided localization for detection in remote sensing images,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.,* vol. 14, pp. 1084–1094, Nov. 2021, doi: 10.1109/JSTARS.2020.3036685.

[48] X. Zheng, W. Zhang, L. Huan, J. Gong, and H. Zhang, “AProNet: Detecting objects with precise orientation from aerial images,” *ISPRS J. Photogrammetry Remote Sens.,* vol. 181, pp. 99–112, Nov. 2021, doi: 10.1016/j.isprsjprs.2021.08.023.

[49] S. M. Azimi, E. Vig, R. Bahmanyar, M. Körner, and P. Reinartz, “Towards multi-class object detection in unconstrained remote sensing imagery,” in *Computer Vision – ACCV*, V. Jawahar, H. Li, G. Mori, and K. Schindler, Eds. Cham, Switzerland: Springer, 2018, pp. 150–165.

[50] X. Yang et al., “Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks,” *Remote Sens.,* vol. 10, no. 1, p. 132, 2018, doi: 10.3390/rs10010132.

[51] Z. Zhang, W. Guo, S. Zhu, and W. Yu, “Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks,” *IEEE Geosci. Remote Sens. Lett.,* vol. 15, no. 11, pp. 1745–1749, Nov. 2018, doi: 10.1109/LGRS.2018.2856921.

[52] Q. Wu, W. Xiang, R. Tang, and J. Zhu, “Bounding box projection for regression uncertainty in oriented object detection,” *IEEE Access,* vol. 9, pp. 58768–58779, 2021, doi: 10.1109/ACCESS.2021.3072402.

[53] B. Zhong and K. Ao, “Single-stage rotation-decoupled detector for oriented object,” *Remote Sens.,* vol. 12, no. 19, p. 3262, 2020, doi: 10.3390/rs12193262.

[54] J. Ding, N. Xue, Y. Long, G. Xia, and Q. Lu, “Learning RoI transformer for oriented object detection in aerial images,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2844–2853, doi: 10.1109/CVPR.2019.00296.

[55] C. Xu, C. Li, Z. Cui, T. Zhang, and J. Yang, “Hierarchical semantic propagation for object detection in remote sensing imagery,” *IEEE Trans. Geosci. Remote Sens.,* vol. 58, no. 6, pp. 4353–4364, Jun. 2020, doi: 10.1109/TGRS.2019.2963243.

[56] 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, Oct. 2022, Art. no. 5608412, doi: 10.1109/TGRS.2021.3113473.

[57] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[58] 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.*, 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[59] Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao, “Oriented response networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4961–4970, doi: 10.1109/CVPR.2017.527.

[60] Q. Zhao, J. Liu, Y. Li, and H. Zhang, “Semantic segmentation with attention mechanism for remote sensing images,” *IEEE Trans. Geosci. Remote Sens.,* vol. 60, Aug. 2022, Art. no. 5403913, doi: 10.1109/TGRS.2021.3085889.

[61] M. Kaselimi, A. Voulodimos, I. Daskalopoulos, N. Doulamis, and A. Doulamis, “A vision transformer model for convolution-free multilabel classification of satellite imagery in deforestation monitoring,” *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2022.3144791.

[62] 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*, Setúbal, Portugal, SciTePress, 2017, vol. 2, pp. 324–331.

[63] 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.

[64] J. Han, J. Ding, N. Xue, and G. S. Xia, “ReDet: A rotation-equivariant detector for aerial object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2785–2794, doi: 10.1109/CVPR46437.2021.00281.

[65] X. Yang, J. Yan, X. Yang, J. Tang, W. Liao, and T. He, “Scrdet++: Detecting small, cluttered and rotated objects via instance-level feature denoising and rotation loss smoothing,” 2020, *arXiv:2004.13316*.

[66] 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.*, 2021, pp. 3500–3509, doi: 10.1109/ICCV48922.2021.00350.

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

[68] J. Ma et al., “Arbitrary-oriented scene text detection via rotation proposals,” *IEEE Trans. Multimedia,* vol. 20, no. 11, pp. 3111–3122, Nov. 2018, doi: 10.1109/TMM.2018.2818020.

[69] Y. Liu, S. Zhang, L. Jin, L. Xie, Y. Wu, and Z. Wang, “Omnidirectional scene text detection with sequential-free box discretization,” 2019, *arXiv:1906.02371*.

Yuxi Zhang received the bachelor’s degree in optoelectronic information engineering from the Harbin Institute of Technology, Weihai, China, in 2018. He is currently working toward the Ph.D. degree in optical engineering with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China.

His research interests include image processing, deep learning, and object detection in remote sensing images.

Yongcheng Wang received the bachelor’s degree from Jilin University, Changchun, China, in 2003, and the Ph.D. degree from the Chinese Academy of Sciences, Changchun, China, in 2010.

He is currently a Researcher with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. His research interests include image engineering and space payload embedded systems.

Ning Zhang received the bachelor’s degree in communication engineering from Northeastern University, Qinhuangdao, Qinhuangdao, China, in 2017. She is currently working toward the Ph.D. degree in optical engineering with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China.

And she is currently with the Technical University of Munich, Munich, Germany, as a visiting Ph.D. student. Her research interests include remote sensing image super-resolution and deep learning.

Zheng Li received the bachelor’s degree in optoelectronic information engineering from the Changchun University of Science and Technology, Changchun, China, in 2020. He is working toward the Ph.D. degree in optical engineering with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Beijing, China.

His research interests cover image processing, deep learning, and object detection in remote sensing images.

Zhikang Zhao received the bachelor's degree in optoelectronic information engineering from the Ocean University of China, Qingdao, China, in 2019. He is currently working toward the Ph.D. degree in optical engineering with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China.

His research interests include image processing, deep learning, and remote sensing image super-resolution.

Yunxiao Gao received the bachelor's degree in measurement and control technology from the Qufu Normal University, Jining, China, in 2018. He is currently working toward the Ph.D. degree in optical engineering with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science, Changchun, China.

His research interests include image processing, deep learning, and object detection in remote sensing images.

Dongdong Xu received the bachelor's degree from Shandong University, Jinan, China, in 2013, the master's degree from the Harbin Institute of Technology, Harbin, China, in 2015, and the Ph.D. degree from the Chinese Academy of Sciences, Changchun, China, in 2020.

He is currently an Assistant Researcher with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. His research interests include deep learning, image fusion, and embedded system software development.

Guangli Ben received the bachelor's and master's degrees from Harbin Engineering University, Harbin, China, in 2013 and 2016, respectively. He is currently working toward the Ph.D. degree with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China.

He is also a Research Assistant with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. His research interests include digital signal processing and space payload embedded systems.