International Journal of Computer Vision
https://doi.org/10.1007/s11263-024-01989-w

Oriented R-CNN and Beyond

Xingxing Xie[1] · Gong Cheng[1] · Jiabao Wang[1] · Ke Li[2] · Xiwen Yao[1] · Junwei Han[1]

Received: 19 February 2023 / Accepted: 1 January 2024

Abstract
Currently, two-stage oriented detectors are superior to single-stage competitors in accuracy, but the step of generating oriented proposals is still time-consuming, thus hindering the inference speed. This paper proposes an Oriented Region Proposal Network (Oriented RPN) to produce high-quality oriented proposals in a nearly cost-free manner. To this end, we present a novel representation manner of oriented objects, named midpoint offset representation, which avoids the complicated design of oriented proposal generation network. Built on Oriented RPN, we develop a simple yet effective oriented object detection framework, called Oriented R-CNN, which could accurately and efficiently detect oriented objects. Moreover, we extend Oriented R-CNN to the task of instance segmentation and realize a new proposal-based instance segmentation method, termed Oriented Mask R-CNN. Without bells and whistles, Oriented R-CNN achieves state-of-the-art accuracy on all seven commonly-used oriented object detection datasets. More importantly, our method has the fastest speed among all detectors. For instance segmentation, Oriented Mask R-CNN also achieves the top results on the large-scale aerial instance segmentation dataset, named iSAID. We hope our methods could serve as solid baselines for oriented object detection and instance segmentation. Code is available at https://github.com/jbwang1997/OBBDetection.

Keywords Oriented object detection · Oriented region proposal network · Instance segmentation

1 Introduction

Oriented object detection is a fundamental yet challenging problem in the tasks of aerial object detection (Ding et al., 2021; Xia et al., 2018) and scene text detection (Xu et al., 2021; Li et al., 2021). Currently, most advanced oriented detectors (Ma et al., 2018; Ding et al., 2019; Han et al., 2021b) are based on the two-stage pipeline: first generating oriented proposals, and then refining the proposals and classifying them into different categories. They have dominated in terms of accuracy but their proposal generation approaches are still time-consuming.

One of the pioneering methods of generating oriented proposals is Rotated Region Proposal Network (Rotated RPN for short) (Ma et al., 2018), which places 54 anchors with different angles, scales, and aspect ratios (3 scales × 3 ratios × 6 angles) on each location, as shown in Fig. 1a. The introduction of rotated anchors improves the recall and demonstrates good performance when the oriented objects distribute sparsely. However, the abundant preset anchors cause massive computation and memory footprint. To address this issue, more recently, RoI Transformer (Ding et al., 2019) learns oriented proposals from horizontal Regions of Interest (RoIs) by complex process, which involves horizontal proposal generation via RPN, RoIAlign, and oriented proposal generation (see Fig. 1b). RoI Transformer drastically reduces the number of rotated anchors, produces relatively promising oriented proposals, and improves detection results, but it also brings about expensive computation costs. Given such circumstances, some representative studies, like R3Det (Yang et al., 2021b) and S2ANet (Han et al., 2021a), draw the idea of multiple regressions in two-stage detection and propose refined detectors. They first regress rotated anchors from horizontal ones, then apply feature interpolation or deformable convolution to align the features with rotated anchors, and finally refine and classify these rotated anchors on top of this. Although such methods avoid placing dense anchors and have decent efficiency, they fall short of achieving a satisfactory level of detection accuracy.

Communicated by Jifeng Dai.

✉ Gong Cheng
gcheng@nwpu.edu.cn

[1] Northwestern Polytechnical University, Xi'an, China

[2] Zhengzhou Institute of Surveying and Mapping, Zhengzhou, China

Published online: 29 January 2024

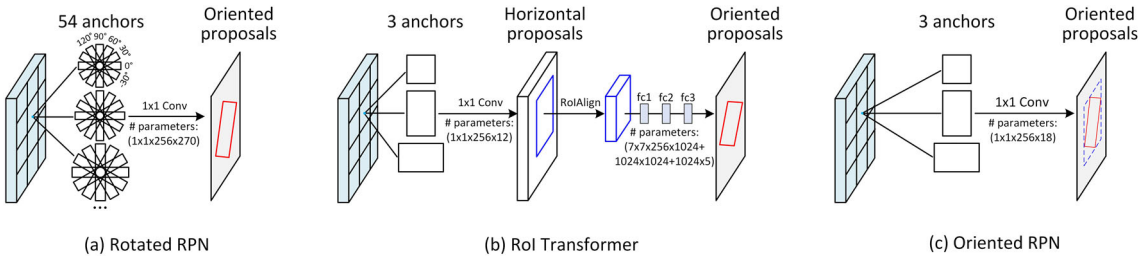

Fig. 1 Comparisons of different methods of oriented proposal generation. **a** Rotated RPN places 54 rotated anchors (3 scales × 3 ratios × 6 angles) on each location of feature maps. **b** RoI Transformer learns oriented proposals by the complex process, which contains horizontal proposal generation, RoIAlign, and oriented proposal generation. **c** Oriented RPN generates high-quality proposals in a nearly cost-free manner

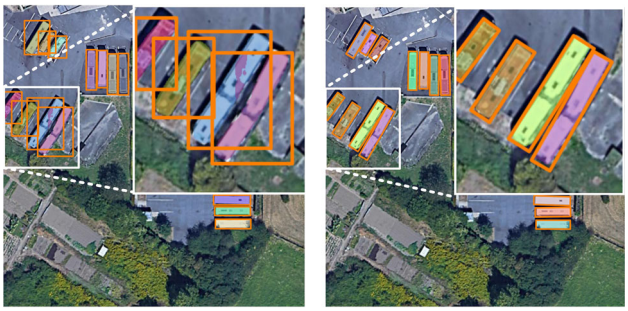

Fig. 2 **a** Region-based instance segmentation methods take horizontal proposals as the input of mask head, e.g., the popular Mask R-CNN. Thus, they face serious feature misalignment between the proposals and oriented objects because many instances are densely arranged with arbitrary orientations, degenerating the instance segmentation accuracy. **b** Our Oriented Mask R-CNN uses oriented proposals for instance segmentation. It could provide more accurate features for the mask head, thus leading to high-quality segmentation results. Here, white boxes denote the zoomed regions for better visibility

A natural and intuitive question to ask is: can we design an extremely simple proposal generation network, as the popular RPN (Ren et al., 2017), to efficiently generate high-quality oriented proposals? Motivated by this question, in the paper, we propose an elegant Oriented Region Proposal Network (Oriented RPN for short), as shown in Fig. 1c, which is a light-weight, fully-convolutional network with much fewer parameters than Rotated RPN and RoI Transformer, thus generating high-quality oriented proposals in a nearly cost-free manner. The ingenious design of Oriented RPN benefits from our proposed representation manner of oriented objects. In brief, the representation, called midpoint offset representation, uses six parameters to represent an oriented object. It, on the one hand, avoids the complicated design of oriented proposal generation network, and on the other hand works in cooperation with horizontal bounding boxes to provide bounded offset ranges for oriented proposal generation. Built on Oriented RPN, we develop an effective two-stage oriented object detector, termed Oriented R-CNN, which can not only improve the detection accuracy but also drastically accelerate the detection speed.

We further move our study to the task of aerial instance segmentation (Waqas Zamir et al., 2019; Follmann & König, 2019). As we know, most of the current region-based instance segmentation methods (He et al., 2017; Liu et al., 2018; Chen et al., 2019), such as the popular Mask R-CNN (He et al., 2017), are based on horizontal proposals. However, when dealing with aerial images, these methods face serious feature misalignment between the horizontal proposals and oriented objects because many instances are densely arranged with arbitrary orientations, as shown in Fig. 2. This challenge severely degenerates the instance segmentation accuracy. To address this issue, we propose a new aerial instance segmentation framework, named Oriented Mask R-CNN by adding a mask head to the Oriented R-CNN. To the best of our knowledge, Oriented Mask R-CNN is the first work to use oriented proposals for instance segmentation.

Without tricks, Oriented R-CNN achieves state-of-the-art detection accuracy on all commonly-used datasets for oriented object detection. More importantly, our method has the fastest detection speed among all detectors. For instance segmentation task, Oriented Mask R-CNN also achieves the top results on the large-scale aerial instance segmentation dataset iSAID. We hope our simple and effective methods will serve as strong baselines for oriented object detection and instance segmentation.

A preliminary version of this work was published in Xie et al. (2021). In this extended version, we introduce the following three major improvements: (i) To comprehensively verify the generality and advantages of our Oriented R-CNN, we present more extensive evaluation in the tasks of oriented object detection from two datasets to seven widely-used benchmarks. Our Oriented R-CNN achieves new state-of-the-art accuracy on all seven datasets. (ii) A new method, termed Oriented Mask R-CNN, is introduced by extending Oriented R-CNN to aerial instance segmentation. In contrast to prevalent methods, Oriented Mask R-CNN is based on oriented proposals. It surpasses the popular Mask R-

CNN with a large margin on the large-scale aerial instance segmentation dataset. (iii) We conduct detailed ablation studies to investigate the quality of proposal, the runtime of proposal generation, the midpoint offset representation, the balance between detection speed and detection accuracy, the performance of Oriented R-CNN in combination with other advanced components, and the contribution of oriented proposals to aerial instance segmentation. Furthermore, to facilitate the future research on oriented object detection and instance segmentation, we have released a codebase, named OBBDetection, which includes nine popular oriented object detection algorithms and two aerial instance segmentation methods, while having faster training speed than AerialDetection. For more detail, readers can refer to https://github.com/jbwang1997/OBBDetection.

2 Related Work

2.1 Oriented Object Detection

Driven by the availability of GPUs with very high computational capability and advanced deep learning algorithms (Xie et al., 2017; Liu et al., 2020; Hu et al., 2018; Huang et al., 2017; Gao et al., 2021; Cheng et al., 2023a), oriented object detection has achieved remarkable breakthroughs (Xu et al., 2021; Pan et al., 2020; Cheng et al., 2023c; Xie et al., 2023a; Cheng et al., 2023d; Li et al., 2023).

Two-stage oriented object detection methods have been dominating oriented object detection for years due to the high detection accuracy. Rotated RPN (Ma et al., 2018), as one of the early works devoted to oriented proposal generation, achieves oriented object detection by densely placing a large number of rotated anchors on the stage of proposal generation. To alleviate the computational burden caused by densely setting anchors, RoI Transformer (Ding et al., 2019) was proposed by designing a proposal transformation learner. Its core idea is to generate oriented proposals from horizontal ones produced by the conventional RPN, but the complex transformation process makes detection still slow. Recently, built on RoI Transformer, ReDet (Han et al., 2021b) introduces a rotation-equivariant backbone for oriented object detection. It improves the detection accuracy to some extent but the specifically designed backbone contains many time-consuming operations, thus making the detection inefficient.

Furthermore, there also exist some two-stage works (Yang et al., 2019; Xu et al., 2021) that achieve oriented object detection by directly using horizontal proposals as the input of the detection head. For instance, to deal with the small, clustered and rotated object, SCRDet (Yang et al., 2019) extends the Faster R-CNN (Ren et al., 2017) by adding an angle regression parameter on the detection head as well as introducing a feature fusion module and an Intersection over Union (IoU) constant factor. Xu et al. (2021) presented a representation of oriented objects, termed gliding vertex, for oriented object detection. With the representation, it realizes oriented object detection by predicting the offsets of each vertex on its corresponding side. The above approaches depending on horizontal proposals are quite simple to implement. However, the horizontal proposals usually contain background and even multiple instances, leading to severe feature misalignment between region proposals and oriented objects.

One-stage detection paradigm (Lin et al., 2017b; Han et al., 2021a; Pan et al., 2020; Yang et al., 2021a; Guo et al., 2021; Ming et al., 2021; Chen et al., 2020; Qian et al., 2021; Yang et al., 2021c; Xie et al., 2023b) is famous for fast detection speed. To improve detection accuracy while maintaining the speed advantage, R3Det (Yang et al., 2021b) and S2ANet (Han et al., 2021a) introduce feature alignment and regression refinement designs within RetinaNet (Lin et al., 2017b). Feature alignment, similar in function to the rotated RoIAlign in the two-stage detectors, aims to align features with learned rotated anchors through feature interpolation or deformable convolution, and regression refinement, drawing the idea of multiple regressions in two-stage detection, is utilized to further adjust the locations of rotated anchors for more accurate regression. Also, to align features for oriented object detection, CFA (Guo et al., 2021) constructs convex-hulls and dynamically divides negative and positive convex-hulls. Recently, some studies are presented inspired by anchor-free detection paradigm. For example, EAST (Zhou et al., 2017) directly predicts the oriented bounding boxes of objects via a polygon-based representation way with a single convolutional neural network. DRN (Pan et al., 2020) designs a dynamic refine network for oriented object detection based on the anchor-free detection method CenterNet (Zhou et al., 2019). To better select high-quality training samples, DAL (Ming et al., 2021) proposes a new sample assignment strategy via dynamic anchor learning for one-stage oriented detectors. PIoU (Chen et al., 2020) devises a novel IoU loss for oriented object detection. In addition, some works (Yang & Yan, 2020; Yang et al., 2021a, c; Qian et al., 2021) devote to addressing the challenges of boundary discontinuity and angular periodicity during the training of oriented detectors. On the whole, one-stage methods have faster speed than two-stage methods, but the main limitation of one-stage detectors is the detection accuracy. Even the advanced detectors like R3Det and S2ANet, which adopt the strategy of two-stage refinement, do not achieve the accuracy comparable to two-stage methods.

Our proposed method falls within two-stage paradigm. However, different from the aforementioned two-stage detectors, which either generate oriented proposals with heavy computational burden or directly take horizontal proposals as input, the aim of this work is to design a high-efficiency

Oriented RPN to achieve oriented object detection with fast speed and high accuracy.

2.2 Instance Segmentation

Instance segmentation is another challenging task in computer vision, which predicts a pixel-level mask and the category for each object instance. This task is closely related to both the tasks of object detection and semantic segmentation (Lang et al., 2023a; Cheng et al., 2023b), but it is more difficult than object detection. Over the past few years, a lot of research efforts (He et al., 2017; Liu et al., 2018; Chen et al., 2019; Bolya et al., 2019; Cai & Vasconcelos, 2021; Lang et al., 2023b) have been made to improve the segmentation results. These approaches can be generally divided into two categories as object detection: two-stage methods and one-stage methods.

Two-stage instance segmentation methods follow the detect-then-segment pipeline. They first predict bounding boxes and then perform segmentation in the area of each bounding box. As an representative two-stage instance segmentation approach, Mask R-CNN (He et al., 2017) extends the Faster R-CNN (Ren et al., 2017) by adding a mask branch on each proposal and proposes RoIAlign to address the pixel misalignment issue. Following Mask R-CNN, PANet (Liu et al., 2018) designs bottom-up path augmentation and adaptive pooling to boost the segmentation results. Later, Cascade Mask R-CNN (Cai & Vasconcelos, 2021) and HTC (Chen et al., 2019) introduce a mask branch to cascaded object detection framework and achieve top results. To alleviate the incompatibility between the mask quality and the classification score, Mask scoring R-CNN (Huang et al., 2019) devices a Mask-IoU branch for predicting the quality of masks.

One-stage instance segmentation approaches usually generate position sensitive maps that are assembled into final masks with position-sensitive pooling. For example, YOLACT (Bolya et al., 2019) first predicts a set of prototype masks, and then linearly combines these masks to generate final results. In addition, some researchers attempt to directly predict the mask and its class for each instance without post-processing. For instance, PolarMask (Xie et al., 2020) uses a novel polar representation to encode the masks and transforms the mask prediction into distance regression. (Tian et al., 2020) proposed a new instance segmentation method, termed CondInst, to reduce the parameters and computational cost of the mask head by dynamically generating the filters. Also, SOLO (Wang et al., 2020b) directly predicts the instance masks and their corresponding classes from an input image with a single convolutional network.

To date, the above instance segmentation approaches, especially Mask R-CNN and its variants, have obtained significant improvement in the accuracy over the years for natural images. However, it is worthy of noting that, when dealing with densely arranged objects with arbitrary orientations, most of existing two-stage methods face acute feature misalignment issue because they adopt horizontal proposals as the input of the mask head, as shown in Fig. 2. This work aims to address the challenge by exploring oriented proposal-driven instance segmentation method, termed Oriented Mask R-CNN. Compared with the baseline Mask R-CNN, our method significantly boost the accuracy with a very big margin.

3 Oriented R-CNN

Our proposed object detection method, called Oriented R-CNN, consists of an Oriented RPN and an oriented detection head (see Fig. 3). It is a two-stage detector, where the first stage generates high-quality oriented proposals in a nearly cost-free manner and the second stage is oriented detection head for proposal classification and regression. Our FPN backbone follows (Lin et al., 2017a), which produces five levels of features $\{P_2, P_3, P_4, P_5, P_6\}$. For simplicity, we do not show the FPN architecture as well as the classification branch in Oriented RPN. Next, we describe the Oriented RPN and oriented detection head in detaily.

3.1 Oriented RPN

Given an input image of any size, Oriented RPN outputs a sparse set of oriented proposals. The entire process is modeled by a light-weight fully-convolutional network, as illustrated in Fig. 4.

Specifically, it takes five levels of features $\{P_2, P_3, P_4, P_5, P_6\}$ of FPN as input and attaches a head of the same design (a 3×3 convolutional layer and two sibling 1×1 convolutional layers) to each level of features. We assign three horizontal anchors with three aspect ratios $\{1{:}2, 1{:}1, 2{:}1\}$ to each spatial location in all levels of features. The anchors have the pixel areas of $\{32^2, 64^2, 128^2, 256^2, 512^2\}$ on $\{P_2, P_3, P_4, P_5, P_6\}$, respectively. Each anchor $\boldsymbol{a}$ is denoted by a 4-dimensional vector $\boldsymbol{a} = (x_\mathrm{a}, y_\mathrm{a}, w_\mathrm{a}, h_\mathrm{a})$, where $(x_\mathrm{a}, y_\mathrm{a})$ is the center coordinate of the anchor, w_a and h_a represent the width and height of the anchor. One of the two sibling 1×1 convolutional layers is regression branch: outputting the offset $\boldsymbol{\delta} = (\delta_x, \delta_y, \delta_w, \delta_h, \delta_\alpha, \delta_\beta)$ of the proposals relative to the anchors. At each location of feature map, we generate 3 proposals, thus the regression branch has 18 outputs. By decoding the regression outputs, we can obtain the oriented proposal. The process of decoding is described as follows:

$$\begin{cases} x = \delta_x \cdot w_\mathrm{a} + x_\mathrm{a}, \quad y = \delta_y \cdot h_\mathrm{a} + y_\mathrm{a} \\ w = w_\mathrm{a} \cdot e^{\delta_w}, \quad h = h_\mathrm{a} \cdot e^{\delta_h} \\ \Delta\alpha = \delta_\alpha \cdot w, \quad \Delta\beta = \delta_\beta \cdot h \end{cases}, \tag{1}$$

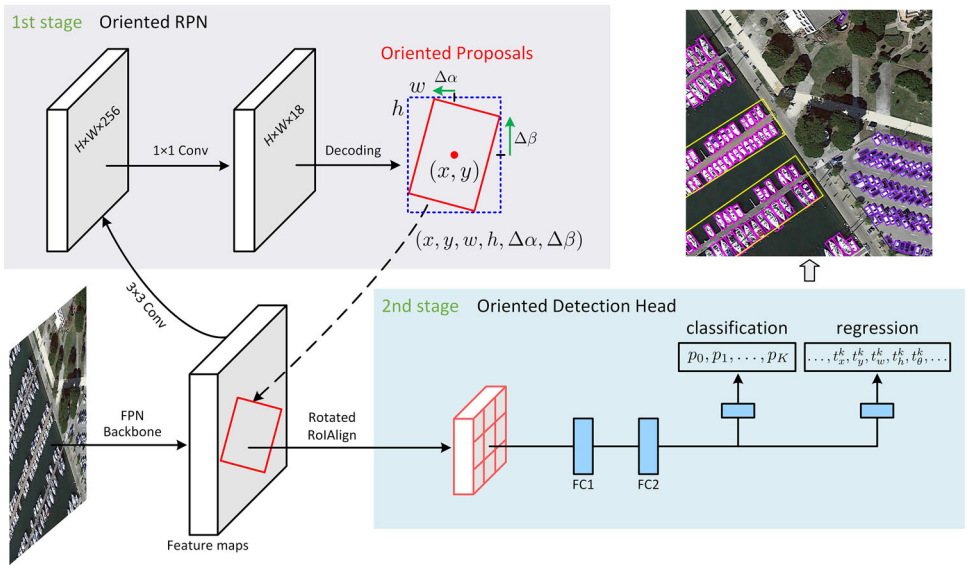

Fig. 3 Overall framework of Oriented R-CNN, which is a two-stage detector built on FPN. The first stage generates oriented proposals by Oriented RPN and the second stage is oriented detection head to classify proposals and refine their spatial locations. For clear illustration, we do not show the FPN as well as the classification branch in Oriented RPN

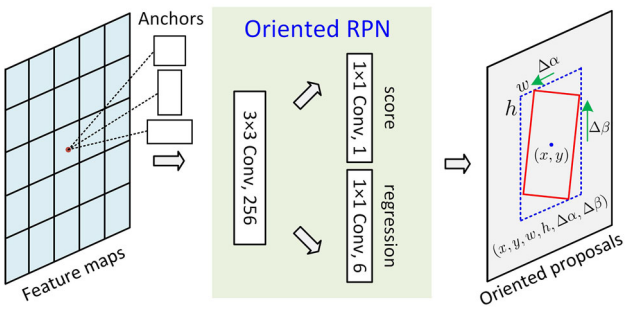

Fig. 4 Oriented RPN is a light-weight fully-convolutional network. It adopts our proposed representation manner of oriented objects, called midpoint offset representation

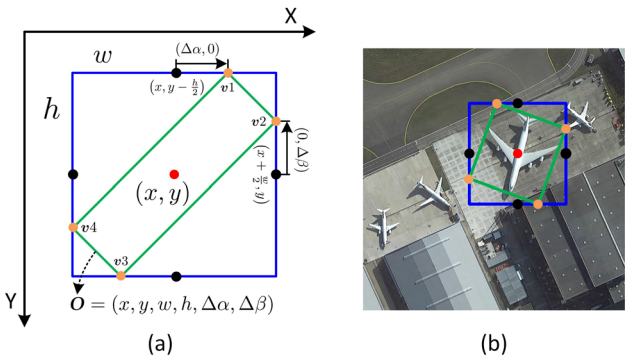

Fig. 5 Illustration of midpoint offset representation. **a** The schematic diagram of midpoint offset representation. **b** An example of midpoint offset representation

where (x, y) is the center coordinate of the predicted proposal, w and h are the width and height of the external rectangle box of the predicted oriented proposal. $\Delta\alpha$ and $\Delta\beta$ are the offsets relative to the midpoints of the top and right sides of the external rectangle. Finally, we produce oriented proposals according to $(x, y, w, h, \Delta\alpha, \Delta\beta)$.

The other sibling convolutional layer estimates the objectness score for each oriented proposal. For clear illustration, we omit the scoring branch in Fig. 3. Oriented RPN is actually a natural and intuitive idea, but its key lies in the representation of oriented objects. Under this circumstance, we design a new and simple representation scheme of oriented objects, called midpoint offset representation.

3.1.1 Midpoint Offset Representation

We propose a novel representation scheme of oriented objects, named midpoint offset representation, as shown in Fig. 5. The black dots are the midpoints of each side of the horizontal box, which is the external rectangle of the oriented bounding box $\boldsymbol{O}$. The orange dots stand for the vertexes of the oriented bounding box $\boldsymbol{O}$.

Specifically, we use an oriented bounding box $\boldsymbol{O}$ with six parameters $\boldsymbol{O} = (x, y, w, h, \Delta\alpha, \Delta\beta)$ to represent an object computed by Eq. (1). Through the six parameters, we can obtain the coordinate set $\mathbf{v} = (\boldsymbol{v}1, \boldsymbol{v}2, \boldsymbol{v}3, \boldsymbol{v}4)$ of four vertexes for each proposal. Here, $\Delta\alpha$ is the offset of $\boldsymbol{v}1$ with respect to the midpoint $(x, y-h/2)$ of the top side of the horizontal box. According to the symmetry, $-\Delta\alpha$ represents the offset of $\boldsymbol{v}3$ with respect to the bottom midpoint $(x, y+h/2)$. $\Delta\beta$ stands for the offset of $\boldsymbol{v}2$ with respect to the right midpoint $(x+w/2, y)$, and $-\Delta\beta$ is the offset of $\boldsymbol{v}4$ with respect to the left midpoint $(x-w/2, y)$. Thus, the coordinates of four vertexes can be expressed as follows:

$$\begin{cases} \boldsymbol{v}1 = (x, y-h/2) + (\Delta\alpha, 0) \\ \boldsymbol{v}2 = (x+w/2, y) + (0, \Delta\beta) \\ \boldsymbol{v}3 = (x, y+h/2) + (-\Delta\alpha, 0) \\ \boldsymbol{v}4 = (x-w/2, y) + (0, -\Delta\beta) \end{cases}. \tag{2}$$

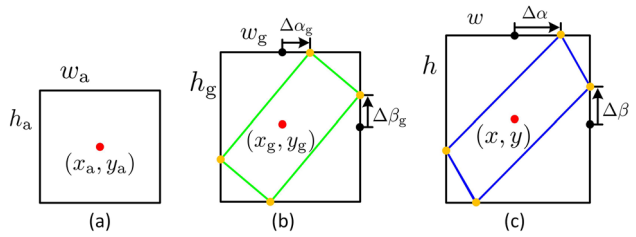

Fig. 6 The illustration of box-regression parameterization. Black dots are the midpoints of the top and right sides, and orange dots are the vertexes of the oriented bounding box. **a** Anchor denoted with (x_a, y_a, w_a, h_a). **b** Oriented ground-truth box denoted with $(x_g, y_g, w_g, h_g, \Delta\alpha_g, \Delta\beta_g)$. **c** Predicted oriented box denoted with $(x, y, w, h, \Delta\alpha, \Delta\beta)$ (Color figure online)

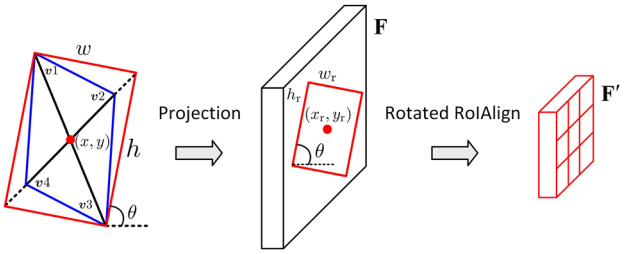

Fig. 7 Illustration of the process of rotated RoIAlign. Blue box is a parallelogram proposal generated by Oriented RPN, and the most-left red box is its corresponding rectangular proposal used for projection and rotated RoIAlign

With the representation manner, we implement the regression for each oriented proposal through predicting the parameters (x, y, w, h) for its external rectangle and inferring the parameters $(\Delta\alpha, \Delta\beta)$ for its midpoint offset.

3.1.2 Loss Function

To train Oriented RPN, the positive and negative samples are defined as follows. First, we assign a binary label $p^* \in \{0, 1\}$ to each anchor. Here, 0 and 1 mean that the anchor belongs to positive or negative sample. To be specific, we consider an anchor as positive sample under one of the two conditions: (i) an anchor having an IoU overlap higher than 0.7 with any ground-truth box, (ii) an anchor having the highest IoU overlap with a ground-truth box and the IoU is higher than 0.3. The anchors are labeled as negative samples when their IoUs are lower than 0.3 with ground-truth box. The anchors that are neither positive nor negative are considered as invalid samples, which are ignored in the training process. It is worth noting that the above-mentioned ground-truth boxes refer to the external rectangles of oriented bounding boxes.

Next, we define the loss function L_1 as follows:

$$L_1 = \frac{1}{N}\sum_{i=1}^{N} F_{\text{cls}}\left(p_i, p_i^*\right) + \frac{1}{N} p_i^* \sum_{i=1}^{N} F_{\text{reg}}\left(\boldsymbol{\delta}_i, \boldsymbol{t}_i^*\right). \tag{3}$$

Here, F_{cls} is the cross entropy loss for classification. F_{reg} is the Smooth $L1$ loss for box regression. i is the index of the anchors and N (by default N=256) is the total number of samples in a mini-batch. p_i^* is the ground-truth label of the i-th anchor. p_i is the output of the classification branch of Oriented RPN, which denotes the probability that the proposal belongs to the foreground. $\boldsymbol{\delta}_i$ is the output of regression branch. $\boldsymbol{t}_i^*$ is the supervision offset of the ground-truth box relative to the i-th anchor, which is a six-dimensional vector $\boldsymbol{t}_i^* = \left(t_x^*, t_y^*, t_w^*, t_h^*, t_\alpha^*, t_\beta^*\right)$, formulated as follows:

$$\begin{cases} t_x^* = \left(x_g - x_a\right)/w_a, & t_y^* = \left(y_g - y_a\right)/h_a \\ t_w^* = \log\left(w_g/w_a\right), & t_h^* = \log\left(h_g/h_a\right) \\ t_\alpha^* = \Delta\alpha_g/w_g, & t_\beta^* = \Delta\beta_g/h_g \end{cases}, \tag{4}$$

where $\left(x_g, y_g\right)$, w_g and h_g are the center coordinate, width, and height of external rectangle. $\Delta\alpha_g$ and $\Delta\beta_g$ are the offsets of the top and right vertexes relative to the midpoints of top and left sides. Please see Fig. 6 for more details.

3.2 Oriented Detection Head

Oriented detection head takes the feature maps $\{P_2, P_3, P_4, P_5\}$ and a set of oriented proposals as input. For each oriented proposal, we use rotated RoI alignment (rotated RoIAlign for short) to extract a fixed-size feature vector from its corresponding feature map. Each feature vector is fed into two fully-connected layers (FC1 and FC2, see Fig. 3), followed by two sibling fully-connected layers: one outputs the probability that the proposal belongs to K+1 classes (K object classes plus 1 background class) and the other one produces the offsets of the proposal for each of the K object classes.

3.2.1 Rotated RoIAlign

Rotated RoIAlign is an operation for extracting rotation-invariant features from each oriented proposal. Now, we describe the process of rotated RoIAlign according to Fig. 7. The oriented proposal generated by Oriented RPN is usually a parallelogram (blue box in Fig. 7), which is denoted with the parameters $\mathbf{v} = (\boldsymbol{v}1, \boldsymbol{v}2, \boldsymbol{v}3, \boldsymbol{v}4)$, where $\boldsymbol{v}1$, $\boldsymbol{v}2$, $\boldsymbol{v}3$, and $\boldsymbol{v}4$ are its vertex coordinates. For ease of computing, we need to adjust each parallelogram to a rectangular with direction. To be specific, we achieve this by extending the shorter diagonal (the line from $\boldsymbol{v}2$ to $\boldsymbol{v}4$ in Fig. 7) of the parallelogram to have the same length as the longer diagonal. After this simple operation, we obtain the oriented rectangular (x, y, w, h, θ) (red box in the left part of Fig. 7) from the parallelogram,

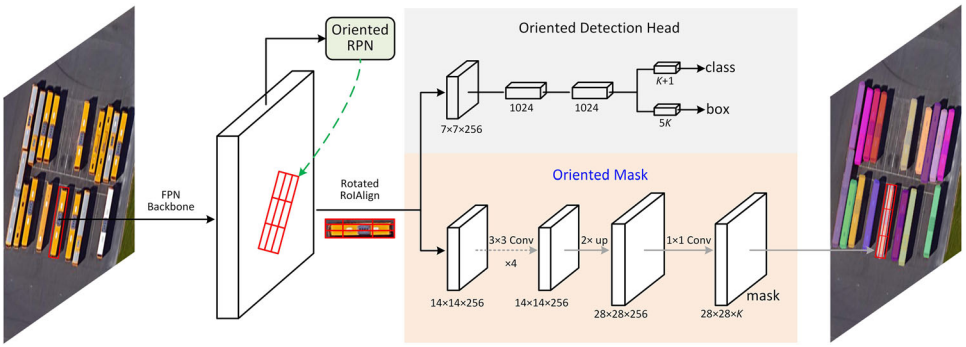

Fig. 8 The Oriented Mask R-CNN architecture, which extends Oriented R-CNN by adding a mask branch. Here, we use FPN backbone for feature extraction. Note that further detail on FPN is not drawn for simplicity

where $\theta \in [-\pi/2, \pi/2]$ is defined by the intersection angle between the horizontal axis and the longer side of the rectangular.

We next project the oriented rectangular (x, y, w, h, θ) to the feature map $\mathbf{F}$ with the stride of s to obtain a rotated RoI, which is defined by $(x_\mathrm{r}, y_\mathrm{r}, w_\mathrm{r}, h_\mathrm{r}, \theta)$ through the following operation:

$$\begin{cases} w_\mathrm{r} = w/s, \quad h_\mathrm{r} = h/s \\ x_\mathrm{r} = \lfloor x/s \rfloor, \quad y_\mathrm{r} = \lfloor y/s \rfloor \end{cases}. \tag{5}$$

Then, each rotated RoI is divided into $m \times m$ grids (m defaults to 7) to get a fixed-size feature map $\mathbf{F}'$ with the dimension of $m \times m \times C$. For each grid with index (i, j) $(0 \le i, j \le m-1)$ in the c-th channel $(1 \le c < C)$, its value is calculated as follows:

$$\mathbf{F}'_c(i, j) = \sum_{(x,y) \in \mathrm{area}(i,j)} \mathbf{F}_c(R(x, y, \theta))/n, \tag{6}$$

where $\mathbf{F}_c$ is the feature of the c-th channel, n is the number of samples localized within each grid, and $\mathrm{area}(i, j)$ is the coordinate set contained in the grid with index (i, j). $R(\cdot)$ is a rotation transformation as the same as (Ding et al., 2019).

3.2.2 Loss Function

As we described above, oriented detection head has two sibling outputs for each rotated RoI. The first one is the classification score $\boldsymbol{p} = (p_0, p_1, \ldots, p_K)$ over $K+1$ categories. The second one outputs the regression offset $\boldsymbol{t}^k = \left(t^k_x, t^k_y, t^k_w, t^k_h, t^k_\theta\right)$ for each of the K object classes, indexed by k $(k = 1, \cdots, K)$.

For a rotated RoI in form of five-dimensional vector $(x_r, y_r, w_r, h_r, \theta)$ labeled with ground-truth class $\boldsymbol{p}^*$ and a regression target $\hat{\boldsymbol{t}} = \left(\hat{t}_x, \hat{t}_y, \hat{t}_w, \hat{t}_h, \hat{t}_\theta\right)$, we use multi-task loss to jointly train the oriented detection head. The loss function L_2 is as follows:

$$L_2 = L_\mathrm{cls}\left(\boldsymbol{p}, \boldsymbol{p}^*\right) + L_\mathrm{reg}\left(\boldsymbol{t}^k, \hat{\boldsymbol{t}}\right), \tag{7}$$

where L_cls is cross entropy loss and L_reg is the Smooth $L1$ loss, which are the same as Eq. (3). Here, $\boldsymbol{t}^k$ and $\hat{\boldsymbol{t}} = \left(\hat{t}_x, \hat{t}_y, \hat{t}_w, \hat{t}_h, \hat{t}_\theta\right)$ are obtained by the following parameterizing equation:

$$\begin{cases} t^k_x = \left(x' - x_\mathrm{r}\right)/w_\mathrm{r}, & \hat{t}_x = \left(\hat{x} - x_\mathrm{r}\right)/w_\mathrm{r} \\ t^k_y = \left(y' - y_\mathrm{r}\right)/h_\mathrm{r}, & \hat{t}_y = \left(\hat{y} - y_\mathrm{r}\right)/h_\mathrm{r} \\ t^k_w = \log\left(w'/w_\mathrm{r}\right), & \hat{t}_w = \log\left(\hat{w}/w_\mathrm{r}\right) \\ t^k_h = \log\left(h'/h_\mathrm{r}\right), & \hat{t}_h = \log\left(\hat{h}/h_\mathrm{r}\right) \\ t^k_\theta = \theta' - \theta, & \hat{t}_\theta = \hat{\theta} - \theta \end{cases}. \tag{8}$$

Here, $\left(x', y'\right)$, w', h' and θ' are the center coordinate, width, height and angle of the predicted oriented bounding box, respectively. $\hat{x}, \hat{y}, \hat{w}, \hat{h}$ and $\hat{\theta}$ are the parameters of the ground-truth of oriented bounding box.

We choose an oriented proposal as positive sample if it has an IoU overlap higher than 0.5 with any oriented ground-truth box. The ground-truth label of the oriented proposal is the same with the ground truth box that has the highest IoU with it.

3.3 Implementation Details

Oriented R-CNN is trained in an end-to-end manner by jointly optimizing Oriented RPN and oriented detection head. During inference, the oriented proposals generated by Oriented RPN generally have high overlaps. In order to reduce the redundancy, we remain 2000 proposals per FPN level in the first stage, followed by Non-Maximum Suppression (NMS). Considering the inference speed, the horizontal NMS with the IoU threshold of 0.8 is adopted. We merge the remaining proposals from all levels, and choose top-1000 ones based on their classification scores as the input of the second stage. In the second stage, poly NMS for each object class is performed on those predicted oriented bounding boxes whose class probability is higher than 0.05. The poly NMS IoU threshold is 0.1.

4 Oriented Mask R-CNN

In this section, we extend our proposed Oriented R-CNN architecture to the instance segmentation task by adding an oriented mask branch, which generates the segmentation results by taking oriented proposals as input. The framework of Oriented Mask R-CNN is illustrated in Fig. 8. In brief, it adopts the same two-stage pipeline as the most popular instance segmentation method Mask R-CNN. The first stage is Oriented RPN. The second stage consists of two parallel branches, namely oriented detection head and oriented mask.

4.1 Network Architecture

We closely follow the Mask R-CNN to build our Oriented Mask R-CNN. To be specific, the Oriented Mask R-CNN branch is a fully convolutional network. It takes the 14×14 RoIs as input, and then followed by four 3×3 convolutional layers. After a stack of four consecutive convolutional layers, we use a 2×2 deconvolutional layer with stride 2 to obtain the high-resolution feature maps with the size of 28×28. Finally the 28×28 feature maps are followed by a 1×1 convolutional with K filters to output the K binary predictions per spatial location. Here K is the number of object classes, which is 15 for the iSAID dataset.

4.2 Implementation Details

For Oriented Mask R-CNN, we set the hyper-parameters following the above-mentioned Oriented R-CNN. Next we introduce the training and inference of Oriented Mask R-CNN in detail.

Training For each oriented proposal, the output of the oriented mask branch is K binary masks with the size of 28×28, one for each of the K classes. We define the mask loss as L_{mask} for each oriented proposal, which is the binary cross-entropy loss defined only on positive proposals. For example, L_{mask} only represents the loss of the k-th mask for the oriented proposal with ground-truth class k, and other masks' outputs do not contribute to L_{mask}. During training, we adopt multi-task loss $L{=}L_1{+}L_2{+}L_{\text{mask}}$ to optimize the Oriented Mask R-CNN. For the definitions of L_1 and L_2, please refer to Sects. 3.1.2 and 3.2.2 for more details.

Inference To speed up the inference time and improve the segmentation accuracy, we use fewer, more accurate detection boxes for oriented mask branch. We choose the top-scoring 1000 detection boxes as the input of oriented mask branch on the iSAID dataset. For each detection box, the oriented mask branch outputs K masks, but we only choose the one with the highest classification score. We then map the 28×28 floating-number mask with the same size as the detection box by resizing and rotation transformation, and binarize them with the threshold of 0.5.

5 Experiments

To evaluate our proposed Oriented R-CNN, we conduct extensive experiments on seven widely-used oriented object detection datasets, which include DOTA-v1.0, DOTA-v1.5, DOTA-v2.0, DIOR-R, HRSC2016, FAIR1M, and ICDAR2015. In addition, we verify our Oriented Mask R-CNN on the iSAID dataset for instance segmentation. Useless otherwise noted, we report the detection accuracy in mean Average Precision (mAP) and the detection speed measured in terms of Frame Per Second (FPS) for the task of object detection. For instance segmentation, we use AP, AP_{50} and AP_{75} as the evaluation criteria. Here, the subscript numbers 50 and 75 denote the IoU thresholds of 0.5 and 0.75.

5.1 Datasets

5.1.1 Oriented Object Detection Datasets

DOTA-v1.0 (Xia et al., 2018) is a large-scale dataset for oriented object detection. It contains 2806 images and 188282 instances with oriented bounding box annotations, covered by the following 15 object classes: Bridge (BR), Harbor (HA), Ship (SH), Plane (PL), Helicopter (HC), Small vehicle (SV), Large vehicle (LV), Baseball diamond (BD), Ground track field (GTF), Tennis court (TC), Basketball court (BC), Soccer-ball field (SBF), Roundabout (RA), Swimming pool (SP), and Storage tank (ST). The image size of the DOTA dataset is large: from 800 $\times$ 800 to 4000×4000 pixels. DOTA-v1.5 has the same images as DOTA-v1.0. It just adds the Container Crane (CC) class and further supplements the annotations of the objects with small sizes (less than 10 pixels).

DOTA-v2.0 (Ding et al., 2021) is the largest aerial object detection dataset released more recently. There are 11268 images and 1793658 instances in DOTA-v2.0. Compared with DOTA-v1.5, it adds two new classes including Airport (Air) and Helipad (Heli). It is noted that the results on the DOTA-v1.0, DOTA-v1.5 and DOTA-v2.0 datasets are produced by submitting the detection results to DOTA's evaluation server.

DIOR-R (Cheng et al., 2022) is another large-scale aerial object detection benchmark. It contains 23463 images with 800×800 pixels and 192518 instances, covered by the following 20 classes: Vehicle (VE), Airplane (APL), Airport (APO), Ship (SH), Bridge (BR), Overpass (OP), Dam (DAM), Harbor (HA), Baseball field (BF), Basketball court (BC), Tennis court (TC), Golf field (GF), Ground track field (GTF), Stadium (STA), Expressway service area (ESA), Expressway toll station (ETS), Train station (TS), Chimney (CH), Storage tank (STO), and Windmill (WM).

HRSC2016 (Liu et al., 2016) is a widely-used dataset for arbitrary-oriented ship detection. It contains 1061 images

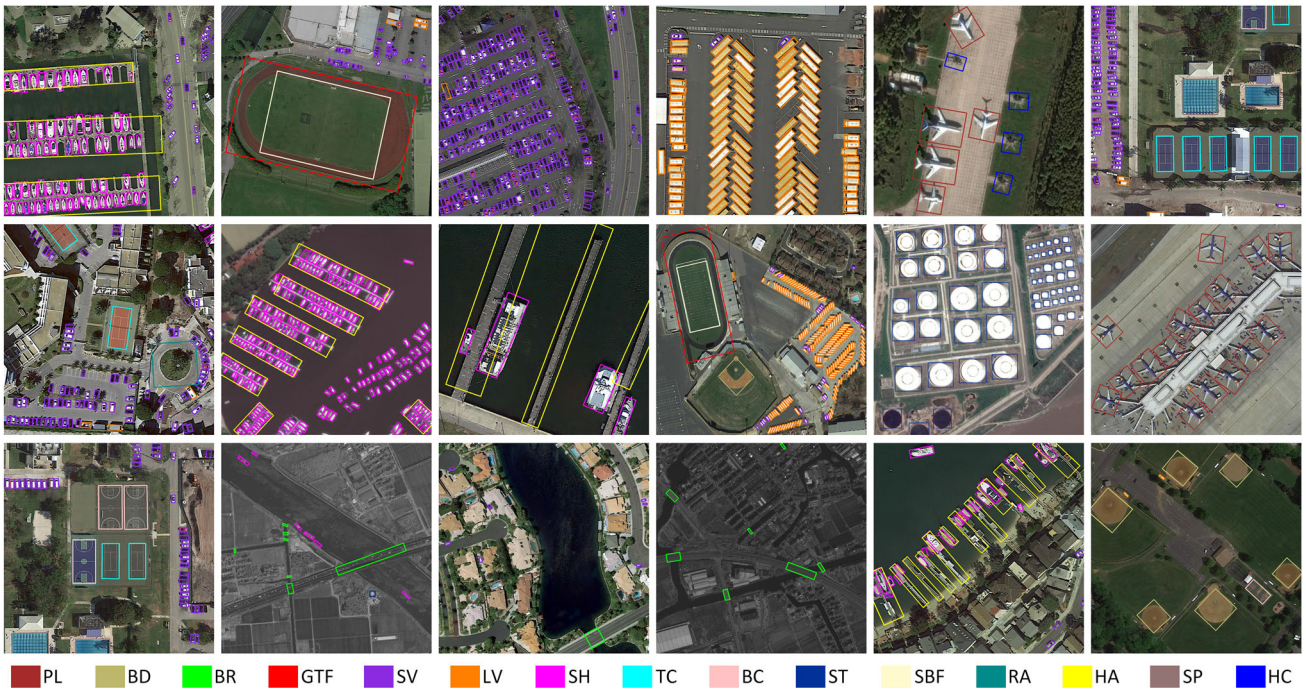

Fig. 9 Examples of detection results on the DOTA-v1.0 dataset using Oriented R-CNN with R50-FPN backbone. The confidence threshold is set to 0.3 when visualizing these results. One color stands for one object class (Color figure online)

with the size ranging from 300 × 300 to 1500 × 900. For a fair comparison, we follow the previous works to report the detection accuracy in terms of mAP with both the PASCAL VOC 2007 (Everingham et al., 2010) and VOC 2012 (Everingham et al., 2015) metrics on the HRSC2016 dataset.

FAIR1M (Sun et al., 2022) is a large-scale remote sensing image dataset for object detection. It has 15266 images and more than one million instances, covered by five categories and 37 sub-categories. The five categories include Airplane, Ship, Vehicle, Court, and Road, while 37 sub-categories are Boeing 737 (B737), Boeing 777 (B777), Boeing 747 (B747), Boeing 787 (B787), Airbus A321 (A321), Airbus A220 (A220), Airbus A330 (A330), Airbus A350 (A350), COMAC C919 (C919), COMAC ARJ21 (ARJ21), other-airplane (OA), passenger ship (PS), motorboat (MB), fishing boat (FB), tugboat (TB), engineering ship (ES), liquid cargo ship (LCS), dry cargo ship (DCS), warship (WS), other-ship (OS), small car (SC), BUS, cargo truck (CT), dump truck (DT), van (VAN), trailer (TRI), tractor(TRC), truck tractor (TT), excavator (EX), other-vehicle (OV), baseball field (BF), basketball court (BC), football field (FF), tennis court (TC), roundabout (RA), intersection (IS), and bridge (BR).

ICDAR2015 (Karatzas et al., 2015) is a popular dataset for scene text detection. It contains 1500 multi-orientated and street-viewed images with the size of 1280×720, 1000 of which are for training and the remaining are for testing. Following the previous works, we use the F-measure as the evaluation metric on ICDAR2015 dataset.

5.1.2 Aerial Instance Segmentation Datasets

iSAID (Waqas Zamir et al., 2019) is a large-scale aerial instance segmentation dataset released in 2019. It shares the same images as DOTA-v1.0, that is, there are 2806 images with the size ranging from 800 × 800 to 4000×4000 pixels. Specifically, iSAID consists of 655451 instances annotated with masks and 15 classes. The training, validation and test sets are consistent with DOTA-v1.0. Similar as the DOTA datasets, there are no public labels for test set, so we need to submit the segmentation results to iSAID's evaluation server.

5.2 Parameter Settings

We use a single RTX 2080Ti GPU with the batch size of 2 for training. The inference time is also tested with a single RTX 2080Ti GPU. The experimental results are produced by using OBBDetection codebase. ResNet50 (He et al., 2016) and ResNet101 (He et al., 2016) are used as our backbones. They are pre-trained on ImageNet (Deng et al., 2009).

Following the common practices, the horizontal and vertical flipping are adopted as data augmentation during training. All networks are trained by Stochastic Gradient Descent (SGD) optimizer with the momentum of 0.9, the weight decay of 0.0001, and the initial learning rate of 0.005. We train the Oriented R-CNN with (i) 12 epochs for the DOTA-v1.0, DOTA-v1.5, DOTA-v2.0, DIOR-R, and FAIR1M datasets, (ii) 36 epochs for the HRSC2016 dataset,

Table 1 Comparison with state-of-the-art methods on the DOTA-v1.0 dataset

Method	Publication	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP	FPS
One-stage																			
DRN (Pan et al., 2020)	CVPR2020	H-104	88.91	80.22	43.52	63.35	73.48	70.69	84.94	90.14	83.85	84.11	50.12	58.41	67.62	68.60	52.50	70.70	–
PIoU (Chen et al., 2020)	ECCV2020	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	–
DAL (Ming et al., 2021)	AAAI2021	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	–
S2ANet (Han et al., 2021a)	TGRS2021	R50-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	**15.3**
R3Det (Yang et al., 2021b)	AAAI2021	R101-FPN	88.76	83.09	50.91	67.27	76.23	80.39	86.72	90.78	84.68	83.24	61.98	61.35	66.91	70.63	53.94	73.79	12.1
RSDet (Qian et al., 2021)	AAAI2021	R101-FPN	89.80	82.90	48.60	65.20	69.50	70.10	70.20	90.50	85.60	83.40	62.50	63.90	65.60	67.20	68.00	72.20	–
BBAVectors (Yi et al., 2021)	WACV2021	R101-FPN	88.35	79.96	50.69	62.18	78.43	78.98	87.94	90.85	83.58	84.35	54.13	60.24	65.22	64.28	55.70	72.32	–
DCL (Yang et al., 2021a)	CVPR2021	R152-FPN	89.10	84.13	50.15	73.57	71.48	58.13	78.00	90.89	86.64	86.78	67.97	67.25	65.63	74.06	67.05	74.06	–
GWD (Yang et al., 2021c)	ICML2021	R101-FPN	89.59	81.18	52.89	70.37	77.73	82.42	86.99	89.31	83.06	85.97	64.07	65.14	68.05	70.95	58.45	74.09	14.8
KLD (Yang et al., 2021d)	NIPS2021	R50-FPN	88.27	76.22	46.22	72.73	72.11	67.84	77.63	90.77	80.67	83.03	52.74	62.23	64.91	65.95	43.22	69.64	14.8
Oriented Reppoint (Li et al., 2022)	CVPR2022	R50-FPN	87.78	77.68	49.54	66.46	78.52	73.12	86.59	90.87	83.75	84.35	53.14	65.63	63.70	68.71	45.91	71.72	15.0
SASM (Hou et al., 2022)	AAAI2022	R50-FPN	87.44	71.31	48.46	68.07	73.93	74.24	83.55	**90.91**	80.36	84.59	57.98	62.84	66.51	63.82	41.17	70.35	13.7
KFIoU (Yang et al., 2023)	ICLR2023	R50-FPN	88.83	77.51	47.79	74.28	71.27	62.72	74.75	90.72	82.34	81.61	58.44	64.23	64.39	67.87	44.07	70.05	14.8
Two-stage																			
Faster R-CNN-O (Ren et al., 2017)	TPAMI2017	R50-FPN	88.44	73.06	44.86	59.09	73.25	71.49	77.11	90.84	78.94	83.90	48.59	62.95	62.18	64.91	56.18	69.05	14.9
Mask R-CNN-O (He et al., 2017)	ICCV2017	R50-FPN	88.70	74.13	50.75	63.66	73.64	73.98	83.68	89.74	78.92	80.26	47.43	65.09	64.79	66.09	59.79	70.71	6.9
Rotated RPN (Ma et al., 2018)	TMM2018	R101	80.94	65.75	35.34	67.44	59.92	50.91	55.81	90.67	66.92	72.39	55.06	52.23	55.14	53.35	48.22	61.01	–
HTC-O (Chen et al., 2019)	CVPR2019	R50-FPN	89.17	75.05	51.95	64.50	74.19	76.30	86.05	90.55	79.51	77.18	50.30	61.23	65.89	68.29	58.01	71.21	6.1
RoI Transformer (Ding et al., 2019)	CVPR2019	R50-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	11.3
SCRDet (Yang et al., 2019)	ICCV2019	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	–
Gliding Vertex† (Xu et al., 2021)	TPAMI2021	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	14.4
ReDet (Han et al., 2021b)	CVPR2021	Re50-FPN	88.79	82.64	53.97	74.00	78.13	84.06	88.04	90.89	87.78	85.75	61.76	60.39	75.96	75.96	63.59	76.25	10.1
Ours																			
Oriented R-CNN	–	R50-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	**15.3**
Oriented R-CNN	–	R101-FPN	88.86	83.48	55.27	76.92	74.27	82.10	87.52	90.90	85.56	85.33	65.51	66.82	74.36	70.15	57.28	76.28	13.3
Oriented R-CNN†	–	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**	**15.3**
Oriented R-CNN†	–	R101-FPN	**90.26**	84.74	**62.01**	**80.42**	79.04	85.07	88.52	90.85	87.24	**87.96**	**72.26**	70.03	**82.93**	**78.46**	68.05	80.52	13.3

The bold numbers indicate the best performance and † denotes multi-scale training and testing (the same below)

Table 2 Comparison with state-of-the-art methods on the DOTA-v1.5 dataset

Method	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	CC	mAP	FPS
Faster R-CNN-O (Ren et al., 2017)	R50-FPN	71.89	74.47	44.45	59.87	51.28	68.98	79.37	90.78	77.38	67.50	47.75	69.72	61.22	65.28	60.47	1.54	62.00	14.9
Mask R-CNN-O (He et al., 2017)	R50-FPN	76.84	73.51	49.90	57.80	51.31	71.34	79.75	90.46	74.21	66.07	46.21	70.61	63.07	64.46	57.81	9.42	62.67	6.9
HTC-O (Chen et al., 2019)	R50-FPN	77.80	73.67	51.40	63.99	51.54	73.31	80.31	90.48	75.12	67.34	48.51	70.63	64.84	64.48	55.87	5.15	63.40	6.1
RoI Transformer (Ding et al., 2019)	R50-FPN	71.92	76.07	51.87	69.24	52.05	75.18	80.72	90.53	**78.58**	68.26	49.18	71.74	67.51	65.53	62.16	9.99	65.03	11.3
S2ANet (Han et al., 2021a)	R50-FPN	78.32	76.37	52.39	69.86	**57.58**	74.27	80.83	**90.88**	76.44	**74.65**	49.96	72.37	65.91	64.23	44.17	0.00	64.26	**15.3**
ReDet (Han et al., 2021b)	Re50-FPN	79.20	**82.81**	51.92	**71.41**	52.38	75.73	80.92	90.83	75.81	68.64	49.29	72.03	**73.36**	**70.55**	**63.33**	**11.53**	66.86	10.1
GWD (Yang et al., 2021c)	R50-FPN	71.63	74.59	48.07	68.70	47.05	60.25	76.23	90.83	69.10	68.72	46.13	70.62	61.22	63.44	34.44	0.01	59.44	14.8
KLD (Yang et al., 2021d)	R50-FPN	71.97	77.93	42.42	64.13	47.98	59.79	77.44	90.81	76.52	67.69	46.86	72.36	52.61	61.91	41.97	0.00	59.52	14.8
Oriented R-CNN	R50-FPN	**79.95**	81.00	**53.90**	70.59	52.48	**76.21**	**86.98**	**90.88**	78.33	68.26	**58.94**	**72.60**	72.75	65.32	58.18	3.72	**66.88**	**15.3**

Bold values indicate the best results

Table 3 Comparison with state-of-the-art methods on the DOTA-v2.0 dataset

Method	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	CC	Air	Heli	mAP	FPS
Faster R-CNN-O (Ren et al., 2017)	R50-FPN	71.61	47.20	39.28	58.70	35.55	48.88	51.51	78.97	58.36	58.55	36.11	51.73	43.57	55.33	57.07	3.51	52.94	2.79	47.31	14.9
Mask R-CNN-O (He et al., 2017)	R50-FPN	76.20	49.91	41.61	60.00	41.08	50.77	56.24	78.01	55.85	57.48	36.62	51.67	47.39	55.79	**59.06**	3.64	60.26	8.95	49.47	6.9
HTC-O (Chen et al., 2019)	R50-FPN	77.69	47.25	41.15	60.71	41.77	52.79	58.87	78.74	55.22	58.49	38.57	52.48	49.58	49.58	54.09	4.20	66.38	**11.92**	50.34	6.1
RoI Transformer (Ding et al., 2019)	R50-FPN	71.81	48.39	45.88	64.02	42.09	54.39	59.92	82.70	63.29	58.71	41.04	52.82	53.32	**56.18**	57.94	25.71	63.72	8.70	52.81	11.3
S2ANet (Han et al., 2021a)	R50-FPN	77.83	**54.57**	44.22	62.99	**47.38**	50.87	57.96	79.72	59.25	**65.36**	39.26	53.35	45.72	52.45	34.08	0.67	67.21	5.03	49.88	**15.3**
ReDet (Han et al., 2021b)	Re50-FPN	**79.61**	54.02	**50.36**	62.29	43.64	50.59	**60.94**	79.76	58.78	59.97	**45.23**	54.79	**55.96**	55.21	55.86	21.26	**69.36**	11.31	53.83	10.1
GWD (Yang et al., 2021c)	R50-FPN	72.64	51.84	36.22	59.88	40.03	40.25	46.92	77.49	56.31	58.95	37.83	53.90	39.70	48.28	36.17	0.01	68.15	0.22	45.82	14.8
KLD (Yang et al., 2021d)	R50-FPN	73.50	47.22	38.58	60.96	39.57	43.46	47.24	76.31	56.60	57.72	38.96	52.64	40.74	47.61	41.19	0.41	68.54	10.59	46.77	14.8
Oriented R-CNN	R50-FPN	78.65	51.80	47.15	**65.78**	43.35	**58.29**	60.89	**82.83**	**63.51**	59.50	43.40	**55.79**	52.90	**56.18**	54.13	**27.55**	66.24	5.22	**54.06**	**15.3**

Bold values indicate the best results

Table 4 Comparison with state-of-the-art methods on the DIOR-R dataset

Method	Backbone	APL	APO	BF	BC	BR	CH	DAM	ETS	ESA	GF	GTF	HA	OP	SH	STA	STO	TC	TS	VE	WM	mAP	FPS
Faster R-CNN-O (Ren et al., 2017)	R50-FPN	62.79	26.80	71.72	80.91	34.20	72.57	18.95	66.45	65.75	66.63	79.24	34.95	48.79	81.14	64.34	**71.21**	81.44	47.31	**50.46**	65.21	59.54	19.1
RoI Transformer (Ding et al., 2019)	R50-FPN	63.34	37.88	71,78	**87.53**	40.68	72.60	26.86	**78.71**	68.09	68.96	**82.74**	**47.71**	55.61	81.21	**78.23**	70.26	**81.61**	**54.86**	43.27	65.52	63.87	18.9
Gliding Vertex (Xu et al., 2021)	R50-FPN	65.35	28.87	74.96	81.33	33.88	**74.31**	19.58	70.72	64.70	**72.30**	78.68	37.22	49.64	80.22	69.26	61.13	81.49	44.76	47.71	65.04	60.06	15.4
Oriented R-CNN	R50-FPN	**71.10**	**39.30**	**79.50**	86.20	**43.10**	72.60	**29.50**	66.70	**79.30**	68.60	82.40	43.60	**57.60**	**81.30**	74.20	62.60	81.40	54.80	46.80	**66.00**	**64.30**	**19.3**

Bold values indicate the best results

(iii) 240 epochs for ICDAR2015 dataset, and train Oriented Mask R-CNN with 12 epochs for the iSAID dataset.

For the DOTA-v1.0, DOTA-v1.5, DOTA-v2.0, and FAIR1M datasets, we crop the original images into 1024×1024 patches. The stride of cropping is set to 824, that is, the pixel overlap between two adjacent patches is 200. With regard to multi-scale training and testing, we first resize the original images at three scales (0.5, 1.0 and 1.5) and crop them into 1024×1024 patches with the stride of 524. The poly NMS threshold is set to 0.1 when merging image patches.

For the HRSC2016 and ICDAR2015 datasets, we do not change the aspect ratios of images. The shorter sides of the images are resized to 800 while the longer sides are less than or equal to 1333. The size of input images of the DIOR-R dataset is the original 800×800. For the iSAID dataset, we divide each large image into fixed-sized patches (800×800 pixels) with the stride of 200.

5.3 Oriented Object Detection

We compare the proposed Oriented R-CNN with other state-of-the-arts on seven benchmarks for oriented object detection. For short, we use R50, R101, R152, RXt50, Re50, DAL-34 and H-104 to stand for ResNet50 (He et al., 2016), ResNet101 (He et al., 2016), ResNet152 (He et al., 2016), ResNeXt50 (Xie et al., 2017), Rotation-equivariant ResNet50 (Han et al., 2021b), 34-layer deep aggression network (Zhou et al., 2019), and 104-layer hourglass network, respectively (Yang et al., 2017). Here, Faster R-CNN-O (Ren et al., 2017; Lin et al., 2017a), Mask R-CNN-O (He et al., 2017) and HTC-O (Chen et al., 2019) are realized for oriented object detection by adding an angle parameter on their regression branches.

DOTA-v1.0 We compare our Oriented R-CNN with 21 state-of-the-art oriented object detection methods, measured in terms of mAP and FPS, including 13 one-stage detectors and eight two-stage ones. It is worth noting that the FPS values are computed with the input image size of 1024×1024. The detailed results are reported in Table 1. From the results, we have the following observations. (i) With ResNet50 and ResNet101, Oriented R-CNN obtains 75.87% mAP and 76.28% mAP, respectively, both of which outperform all methods using the same backbones. (ii) Among all detectors, our method has the fastest speed and the highest detection accuracy. Especially, in comparison with ReDet (Han et al., 2021b) which adopts the specific-purpose rotation-equivariant ResNet50 as backbone, our method is faster than it. (iii) Our method significantly surpasses all one-stage detectors in terms of accuracy while having competitive speed. In particular, compared with the strongest competitor S2ANet (Han et al., 2021a), our method has the same speed but with higher mAP (75.87% *vs.* 74.12%). (iv) Using multi-scale training and testing, our method reaches 80.87% mAP

Table 5 Comparison with state-of-the-art methods on the HRSC2016 dataset

Method	Publication	Backbone	mAP_{07}	mAP_{12}	FPS
Rotated RPN (Ma et al., 2018)	TMM2018	R101	79.08	85.64	–
RoI Transformer (Ding et al., 2019)	CVPR2019	R101-FPN	86.20	-	8.9
DRN (Pan et al., 2020)	CVPR2020	H-104	–	92.70	–
PIoU (Chen et al., 2020)	ECCV2020	DLA-34	89.20	–	–
CSL (Yang & Yan, 2020)	ECCV2020	R101-FPN	89.62	96.10	–
DAL (Ming et al., 2021)	AAAI2021	R101-FPN	89.77	–	–
Gliding Vertex (Xu et al., 2021)	TPAMI2021	R101-FPN	88.20	–	–
S2ANet (Han et al., 2021a)	TGRS2021	R50-FPN	90.17	95.01	**14.9**
R3Det (Yang et al., 2021b)	AAAI2021	R101-FPN	89.26	96.01	–
RSDet (Qian et al., 2021)	AAAI2021	R50-FPN	86.50	–	–
BBAVectors (Yi et al., 2021)	WACV2021	R101-FPN	88.60	–	11.7
DCL (Yang et al., 2021a)	CVPR2021	R101-FPN	89.46	96.41	–
GWD (Yang et al., 2021c)	ICML2021	R101-FPN	89.85	97.37	–
KLD (Yang et al., 2021d)	NIPS2021	R50-FPN	89.97	–	–
Oriented R-CNN	–	R50-FPN	90.40	96.50	**14.9**
Oriented R-CNN	–	R101-FPN	**90.50**	**97.60**	12.3

Bold values indicate the best results

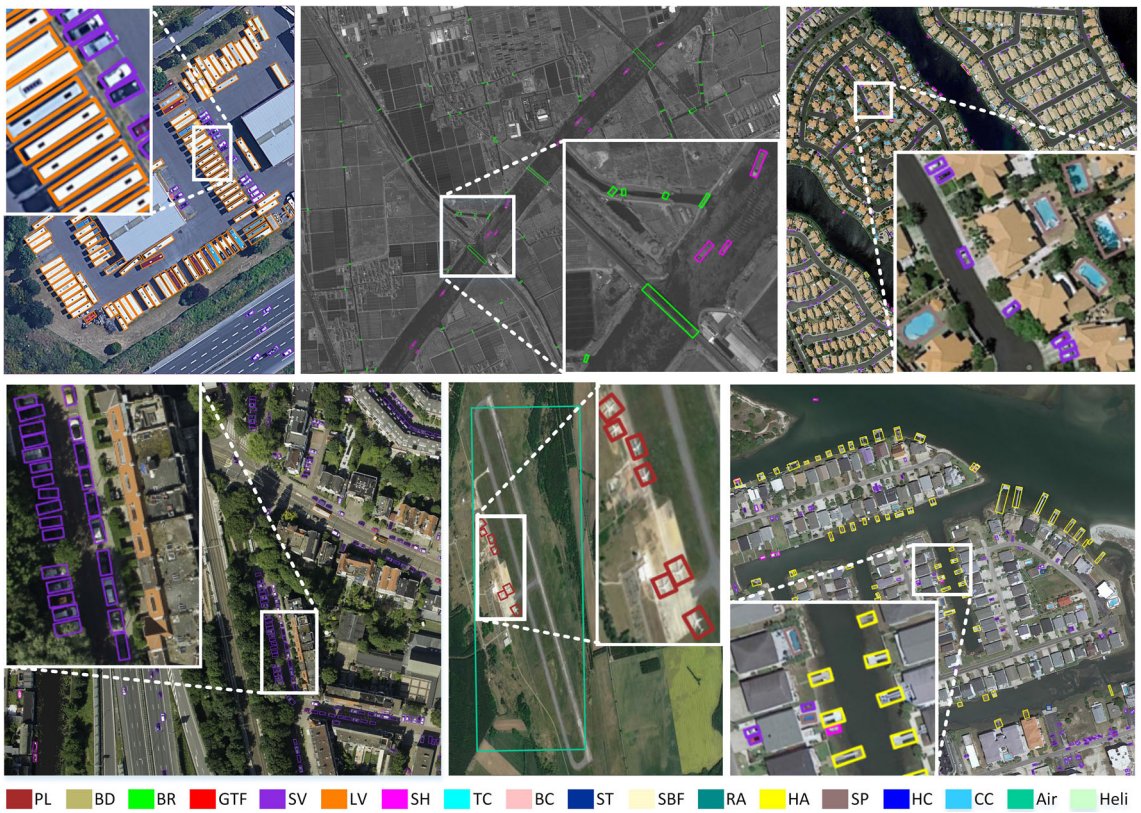

Fig. 10 Examples of detection results on the DOTA-v2.0 dataset using Oriented R-CNN with R50-FPN backbone. Here, white boxes denote the zoomed regions for better visibility

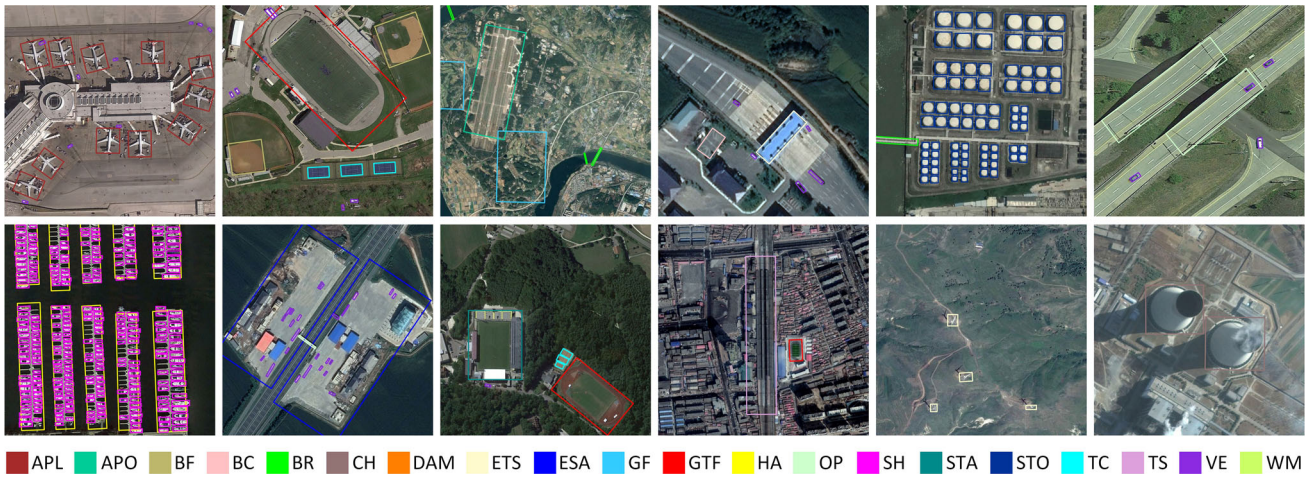

Fig. 11 Examples of detection results on the DIOR-R dataset using Oriented R-CNN with R50-FPN backbone

using ResNet50 as backbone, establishing new state-of-the-arts in aerial object detection. In Fig. 9, we visualize some detection results, which show that Oriented R-CNN works well for detecting the challenging objects with large aspect ratios, small sizes, arbitrary orientations, and dense arrangement.

DOTA-v1.5 & DOTA-v2.0 To evaluate our Oriented R-CNN, we further compare it with eight advanced oriented detectors. The comparison results are listed in Tables 2 and 3, respectively. We have similar observations as that of DOTA-v1.0. With the same ResNet50 backbone, our approach achieves 66.88% mAP and 54.06% mAP on the DOTA-v1.5 and DOTA-v2.0 datasets, respectively, both of which noticeably surpass all eight advanced methods. Compared to the detector ReDet using specially designed rotation equivariant backbone, our method achieves 51.49% speed improvement while maintaining leading accuracy. These results demonstrate the solidness of our proposed method. Figure 10 shows some examples of detection on the DOTA-v2.0 dataset.

DIOR-R & HRSC2016 Table 4 lists the detection results reported by the DIOR-R benchmark and ours. As seen, our Oriented R-CNN obtains the start-of-the-art detection accuracy. We display some visualization results of Oriented R-CNN on the DIOR-R dataset, as shown in Fig. 11. For HRSC2016, we compare Oriented R-CNN with 14 oriented object detection methods including both one-stage and two-stage models. The comparison results are shown in Table 5. Using both the PASCAL VOC2007 and VOC2012 as the metrics, Oriented R-CNN with ResNet101 achieves the best accuracy. Surprisingly, our method with ResNet50 still outperforms all comparison approaches with ResNet101. Some visualization results are shown in Fig. 12.

FAIR1M The comparison results of our method with state-of-the-arts on the FAIR1M dataset are reported in Table 6. Our Oriented R-CNN with ResNet50 obtains 38.85% mAP, outperforming all two-stage and one-stage methods, with the same backbone and experiment setting. Furthermore, Oriented R-CNN reaches 44.81% mAP under multi-scale training and testing, establishing a new state-of-the-art. All this indicates that our approach exhibit excellent performance even for the datasets with more than one million instances.

ICDAR2015 We evaluate our method on the ICDAR2015 dataset, and the evaluation results are presented in Table 7. Using ResNet50 as the backbone, Oriented R-CNN can outperform other advanced methods measured in terms of F-measure and speed (i.e., FPS). This phenomenon demonstrates that our Oriented R-CNN is not only limited to aerial object detection but also shows wide applicability on scene text detection.

Why Oriented R-CNN Works Well? Compared to Faster R-CNN-O, Oriented R-CNN could achieve promising performance on all datasets. In brief, they differ only in the proposal generation network. Faster R-CNN-O utilizes the horizontal RPN, while Oriented R-CNN uses our proposed Oriented RPN. We attribute the reasons why our Oriented R-CNN works to the following factors. (i) In the oriented proposal generation stage, we propose the midpoint offset representation method, which can work well with the horizontal bounding boxes and only needs to predict the midpoint offsets of any two adjacent sides of each horizontal bounding box to determine an oriented candidate box. This alleviates the difficulty of directly regressing oriented candidate boxes from horizontal anchors and achieves high-quality oriented proposal generation, as shown in Fig. 13. (ii) Benefiting from the high-quality oriented proposals generated by Oriented RPN, the features of object regions are more well aligned after RoI alignment, avoiding unnecessary introduction of the features from background or other surrounding objects, and facilitating subsequent classification and location refinement. Likewise, the high-quality proposals improve the quality of

Table 6 Comparison with state-of-the-art methods on the FAIR1M dataset

Coarse Category	Sub Category	RetinaNet	S2ANet	FRCNN	RoITrans	GLVE	**ORCNN**	**ORCNN**†
Airplane	B737	35.24	36.06	33.94	39.15	36.50	35.17	**41.09**
	B747	74.90	84.33	84.25	84.72	81.88	85.17	**85.95**
	B777	10.54	15.62	16.38	14.82	14.06	14.57	**17.38**
	B787	38.49	42.34	47.61	48.88	44.60	47.68	**58.02**
	C919	0.96	1.95	14.44	19.49	10.16	11.68	**20.93**
	A220	41.39	44.09	47.40	**50.31**	46.43	46.55	44.76
	A321	63.51	68.00	68.82	**70.16**	65.47	68.18	68.31
	A330	46.28	63.84	72.71	70.34	67.73	68.60	**74.48**
	A350	63.42	70.00	76.53	72.19	68.31	70.21	**76.74**
	ARJ21	2.29	12.10	26.59	**33.72**	25.97	25.32	29.32
Ship	PS	5.78	8.82	11.03	12.62	10.01	13.77	**17.63**
	MB	23.00	48.03	51.22	55.98	51.63	60.42	**70.13**
	FB	2.52	6.79	6.41	6.12	5.26	9.10	**13.40**
	TB	24.87	34.01	34.19	35.31	34.00	**36.83**	33.75
	ES	6.98	7.49	9.41	9.27	10.34	11.32	**14.11**
	LCS	7.39	18.30	15.17	15.95	14.53	21.86	**26.75**
	DCS	21.75	37.62	32.26	34.15	33.10	38.22	**38.42**
	WS	2.75	22.96	11.27	15.29	12.45	22.67	**31.81**
Vehicle	SC	37.22	61.57	54.56	57.55	54.39	57.62	**74.42**
	Bus	4.25	11.76	22.94	26.43	28.63	24.40	**48.74**
	CT	18.38	34.28	37.74	39.38	36.90	40.84	**49.98**
	DT	12.59	36.03	41.69	44.95	42.16	45.20	**57.38**
	VAN	26.44	54.62	48.23	53.69	48.52	54.01	**75.39**
	TRI	0.01	3.47	12.46	10.95	13.38	15.46	**21.06**
	TRC	0.02	0.96	2.44	2.13	1.39	2.37	**5.38**
	EX	0.13	7.24	11.35	10.99	12.19	13.55	**20.87**
	TT	0.03	0.40	0.32	0.60	0.19	0.24	**3.49**
Court	BC	27.51	38.44	45.18	46.93	45.83	48.18	**54.70**
	TC	79.20	80.44	77.75	79.29	77.99	78.45	**83.08**
	FF	55.44	56.34	52.05	56.72	59.27	60.79	**65.59**
	BF	87.89	87.47	87.19	87.21	86.08	88.43	**89.60**
Road	IS	54.38	50.76	58.71	58.21	58.19	57.90	**60.83**
	RA	**23.91**	16.67	19.38	21.98	19.45	17.57	19.50
	BR	4.38	17.24	20.76	23.31	22.82	28.63	**30.60**
mAP		26.58	34.71	36.83	38.49	36.47	38.85	**44.81**

Bold values indicate the best results
Here, FRCNN, RoITrans, GLVE, and ORCNN represent Faster R-CNN-O, RoI Transformer, gliding vertex, and Oriented R-CNN, respectively.

Fig. 12 Examples of detection results on the HRSC2016 dataset using Oriented R-CNN with R50-FPN backbone

Table 7 Comparison with state-of-the-art methods on the ICDAR2015 dataset

Method	Precision	Recall	F-measure	FPS
EAST (Zhou et al., 2017)	83.60	73.50	78.20	13.2
RRD (Liao et al., 2018)	85.60	79.00	82.20	6.5
MOSTD (Lyu et al., 2018)	**94.10**	70.70	80.70	3.6
TextSnake (Long et al., 2018)	84.90	80.40	82.60	1.1
SegLink++ (Tang et al., 2019)	83.70	80.30	82.00	–
PSENet (Wang et al., 2019)	86.90	84.50	85.70	1.6
Boundary (Wang et al., 2020a)	82.20	88.10	85.00	–
DRRG (Zhang et al., 2020b)	84.70	**88.50**	86.60	–
SAE (Tian et al., 2019)	85.10	84.50	84.80	3.0
FCENet (Zhu et al., 2021)	84.20	85.10	84.60	–
DBNet++ (Liao et al., 2022)	90.90	83.90	87.30	10.0
Oriented R-CNN	89.10	86.40	**87.50**	**15.9**

Bold values indicate the best results

Table 8 Comparison with state-of-the-art methods on the iSAID dataset

Method	Backbone	AP	AP_{50}	AP_{75}	FPS
Mask R-CNN (He et al., 2017)	R101-FPN	25.65	51.30	22.72	9.6
PANet (Liu et al., 2018)	R101-FPN	34.17	56.57	35.84	6.8
D2Det (Cao et al., 2020)	R101-FPN	37.50	61.00	39.80	—
Oriented Mask R-CNN	R50-FPN	38.91	63.21	41.55	**11.2**
Oriented Mask R-CNN	R101-FPN	39.22	63.43	41.59	8.9
Oriented Mask R-CNN	R152-FPN	**40.15**	**64.46**	**43.19**	7.2

Bold values indicate the best results

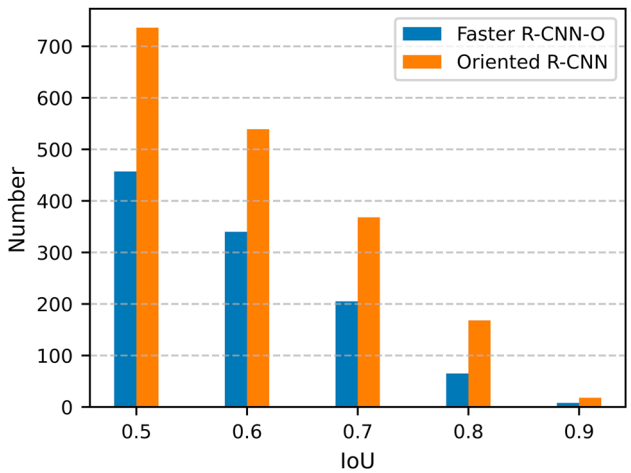

Fig. 13 The number of proposals from Faster R-CNN-O and Oriented R-CNN under various IoU thresholds

the training samples for the second stage of Oriented R-CNN, i.e., the oriented detection head, and help the model to be better optimized.

5.4 Instance Segmentation

We evaluate our Oriented Mask R-CNN on the iSAID dataset and compare it with three advanced instance segmentation methods including Mask R-CNN, PANet and D2Det, whose results are publicly reported on the iSAID benchmark (Waqas Zamir et al., 2019; Cao et al., 2020). The comparison results are summarized in Table 8. As can be seen, Oriented Mask R-CNN outperforms all three methods in terms of AP at different IoU thresholds (AP, AP_{50} and AP_{75}). Our Oriented Mask R-CNN using ResNet50 achieves higher accuracy while maintaining the running speed of 11.2 FPS compared to other methods using ResNet101. Although our model is slower than Mask R-CNN by less than 1 FPS under the same conditions, it exhibits significant advantage in terms of accuracy (39.22% vs. 25.65% AP). Considering both speed and accuracy, our approach performs very well. We attribute the accuracy improvements to our proposed state-of-the-art oriented detector, namely Oriented R-CNN. To be specific, Mask R-CNN and its improved works take the horizontal boxes produced by the Faster R-CNN as input. Their segmentation results are closely related to the predicted boxes. For aerial images, objects often appear with arbitrary orientations (e.g., ships, vehicles, etc.). The horizontal boxes can not well enclose the objects closely. In other words, the IoU values between the boxes and the objects are low. On the contrary, our Oriented Mask R-CNN relies on Oriented R-CNN to provide accurate boxes for the mask branch, thus remarkably boosting the segmentation accu-

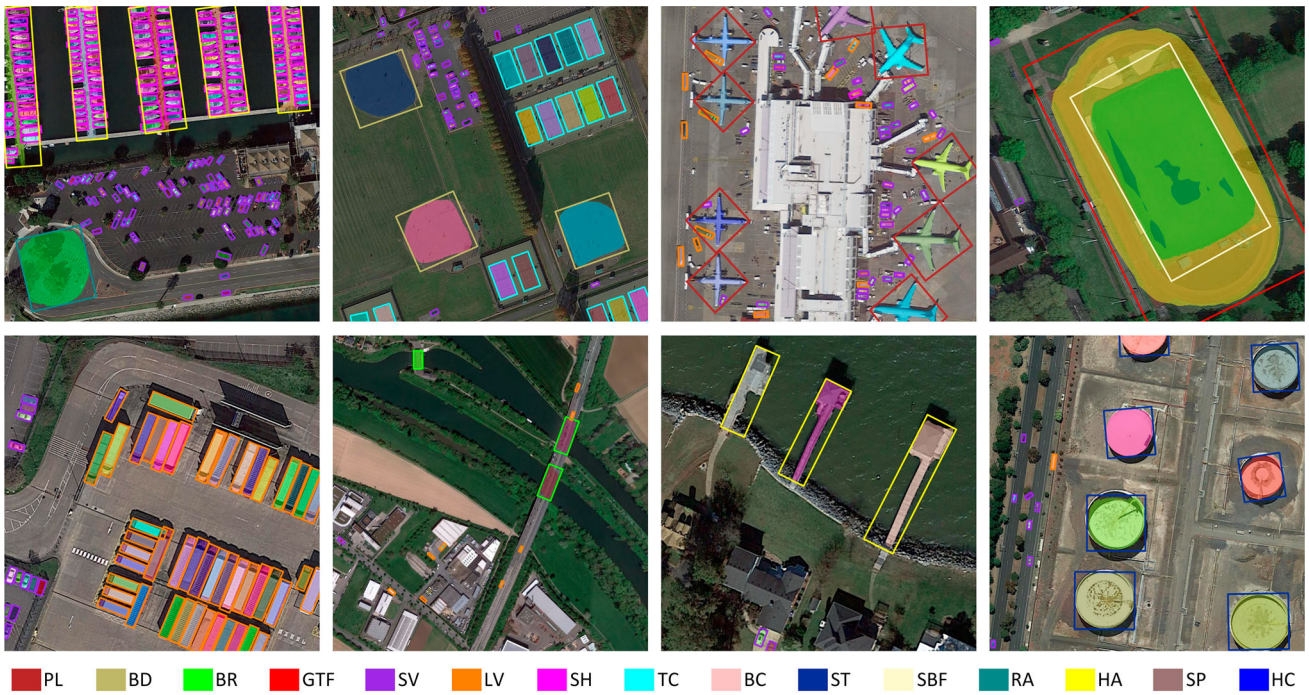

Fig. 14 Examples of instance segmentation results on the iSAID dataset using Oriented Mask R-CNN with ResNet50

Fig. 15 Proposals generated by Oriented RPN on the DOTA dataset. The top-200 proposals per image are displayed

racy. Furthermore, we report the segmentation results of each class in Table 9. As seen, compared with Mask R-CNN, our Oriented Mask R-CNN has non-trivial AP improvement for the categories with evident directions, such as ship, small vehicle, large vehicle and bridge. In Fig. 14, we show the results obtained by our Oriented Mask R-CNN on the iSAID dataset. As illustrated, our method obtains good predictions even under challenging conditions.

Note that Oriented Mask R-CNN is a straightforward structure, which is also the first attempt to use oriented proposals for the task of aerial instance segmentation. We hope our work will encourage future research of instance segmentation on bird's-eye scenes. More strategies, such as multi-scale training and heavy backbones, have the potential to further improve the accuracy but are not the focus of this work.

5.5 Ablation Studies

We perform a series of ablation studies to investigate the quality of proposal, the runtime of proposal generation, the midpoint offset representation, the balance between detection speed and detection accuracy, the performance of Oriented R-CNN in combination with other advanced components, and the contribution of oriented proposals to aerial instance segmentation. If not otherwise specified, the experiments are

Table 9 Comparison with state-of-the-art methods in terms of class-wise instance segmentation on the iSAID dataset

Method	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	AP	FPS
Mask R-CNN (He et al., 2017)	R101-FPN	37.70	42.50	13.00	23.60	6.90	7.40	26.60	54.90	34.60	28.00	20.80	35.90	22.50	25.10	5.30	25.65	9.6
PANet (Liu et al., 2018)	R101-FPN	39.20	45.50	15.10	29.30	15.00	28.80	45.90	74.10	47.40	29.60	33.90	36.90	26.30	36.10	9.50	34.17	6.8
D2Det (Cao et al., 2020)	R101-FPN	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	37.50	–
Oriented Mask R-CNN	R50-FPN	43.79	56.64	20.16	30.25	15.43	33.46	49.22	**77.01**	**54.28**	36.58	35.45	45.45	31.50	38.38	9.17	38.91	**11.2**
Oriented Mask R-CNN	R101-FPN	43.32	56.17	20.18	32.75	15.37	32.63	49.27	76.21	53.86	35.89	40.68	**46.66**	32.40	37.99	8.28	39.22	8.9
Oriented Mask R-CNN	R152-FPN	**44.27**	**57.41**	**20.84**	**33.40**	**15.58**	**33.78**	**50.06**	76.77	53.70	**36.86**	**40.70**	46.34	**33.05**	**38.73**	**14.03**	**40.15**	7.2

Bold values indicate the best results

Table 10 Comparison with different proposal generation methods in Recall on the DOTA-v1.0 validation set

Proposal generation method	R_{2000}	R_{1000}	R_{300}
Rotated RPN (Ma et al., 2018)	70.10	68.50	58.30
RoI transformer (Ding et al., 2019)	87.86	81.85	66.23
Oriented R-CNN	**92.80**	**92.20**	**81.60**

Bold values indicate the best results

conducted on the DOTA-v1.0 validation set, and ResNet50 is used as the backbone.

5.5.1 Proposal Quality versus Proposal Number

As we know, proposal quality and proposal number play important roles in high-performance detectors. To show the superiority of our Oriented RPN, we compare the quality of our proposals to that of two popular oriented proposal generation methods including Rotated RPN (Ma et al., 2018) and RoI Transformer (Ding et al., 2019). Note that the quality of proposals are measured with Recall and the IoU threshold with ground-truth boxes is set to 0.5.

For a fair comparison, we respectively select top-300, top-1000, and top-2000 proposals from each image patch to report the Recall values, denoted as R_{300}, R_{1000}, and R_{2000}. The results are presented in Table 10. As can be seen: (i) With the same proposal number, our Oriented RPN has the highest Recall among all three methods. (ii) Our Oriented RPN achieves the Recall of 92.20% when using 1000 proposals, which is even higher than both that of Rotated RPN and RoI Transformer with 2000 proposals. This suggests that our Oriented RPN can realizes higher Recall with fewer proposals. (iii) The recall of our method drops very slightly (0.6%) when the number of proposals changes from 2000 to 1000, but it goes down sharply when using 300 proposals. We argue that the noticeable decline is caused by the reason that most images on DOTA have more instances than natural images.

In Fig. 15, we show some examples of proposals generated by Oriented RPN on the DOTA dataset. The top-200 proposals per image are displayed. As shown, our proposed Oriented RPN could well localize the objects no matter their sizes, aspect ratios, directions, and denseness.

5.5.2 Runtime of Proposal Generation

To verify the motivation of our proposed Oriented RPN, we compare the runtime and the number of parameters (Param. for short) of three proposal generation models, including Rotated RPN, RoI Transformer and our Oriented RPN. The detailed comparisons are presented in Table 11. Note that the parameters do not contain that of backbones. As shown above, Oriented RPN has the fewest parameters and the

Table 11 Comparison with different proposal generation methods in terms of runtime and model size on DOTA-v1.0 validation set

Proposal generation method	R_{2000}	Runtime	Param
Rotated RPN (Ma et al., 2018)	70.10	49 ms	0.67 M
RoI transformer (Ding et al., 2019)	87.86	72 ms	14.49 M
Oriented R-CNN	**92.80**	**40** ms	**0.59** M

Bold values indicate the best results

Table 12 Comparison with different representations of oriented objects in terms of Recall on the DOTA-v1.0 validation set

	Representation	Recall
RPN	Angle-based (Ma et al., 2018)	87.40
	Polygon-based (Zhou et al., 2017)	88.56
	Midpoint Offset	**92.80**

Bold values indicate the best results

Table 13 Comparisons of gliding vertex and midpoint offset representations for two-stage detection

First Stage	Second Stage	mAP	FPS
(x, y, w, h)	Gliding Vertex	73.23	**14.4**
Gliding Vertex	(x, y, w, h, θ)	74.88	9.4
Gliding Vertex	Gliding Vertex	**75.11**	9.0
(x, y, w, h)	**Midpoint Offset**	74.01	**15.5**
Midpoint offset	(x, y, w, h, θ)	75.87	15.3
Midpoint offset	**Midpoint offset**	**75.90**	15.0

Bold values indicate the best results
The representations of gliding vertex and midpoint offset are $(x, y, w, h, \alpha_1, \alpha_2, \alpha_3, \alpha_4, r)$ and $(x, y, w, h, \Delta\alpha, \Delta\beta)$.

fastest speed among all three methods. Especially, it is nearly twice as fast as the current state-of-the-art proposal generation model, namely RoI Transformer (40 ms vs. 72 ms). This explains why our Oriented RPN has a significant boost to detection speed. We attribute the benefits to the light designs of Oriented RPN, that is, it neither needs to set dense anchors as Rotated RPN nor utilizes complex transformation like RoI Transformer.

5.5.3 Oriented Object Representations

To study the effectiveness of our proposed midpoint offset representation, we compare it with two popular representation manners of oriented objects. They are angle-based and polygon-based representations. Here, the angle-based manner represents an oriented object with five parameters including the center coordinate, width, height, and angle of the oriented box. The polygon-based way uses the coordinates of four corner vertices to denote an oriented object. For more information, we refer the readers to (Zhou et al., 2017).

Table 12 reports the Recall values of oriented proposals obtained by using three representation manners and the IoU threshold with ground-truth boxes is set to 0.5. As we can see, our midpoint offset representation achieves the highest Recall compared with angle-based and polygon-based manners.

For the improvements, we attribute the reasons as two aspects: (i) the midpoint offset representation works in cooperation with horizontal bounding boxes and thus could avoid the ambiguity in defining the order of four vertices; (ii) the midpoint offset representation limits the offset ranges of corner vertices on the side of horizontal boxes and provides bounded offset ranges for oriented proposal generation, facilitating the learning of proposal generation network.

Additionally, we compare the midpoint offset representation with gliding vertex representation for two-stage detection. The comparison results are presented in Table 13. When we utilize both midpoint offset and gliding vertex representations (Xu et al., 2021) to proposal generation (see the second and fifth rows of Table 13), the former is 62.76% faster than the latter, yet with higher accuracy (75.87% vs. 74.88% mAP). By employing midpoint offset representation to output oriented boxes from horizontal ones at the second stage, we obtain higher accuracy (74.01% vs. 73.23% mAP) and faster inference speed (15.5 vs. 14.4 FPS) than using gliding vertex representation. Meanwhile, gliding vertex and midpoint offset representations are scaled to both the two stages of detectors, where the former lags behind the latter in terms of speed and accuracy, as shown in the third and sixth rows of Table 13. These results suggest that (i) Gliding vertex representation does not generalize well to proposal generation at the first stage, (ii) Our representation can work well with proposal generation and, like gliding vertex representation, be easily applied to oriented bounding box regression of objects at the second stage, and (iii) Midpoint offset representation outperforms gliding vertex scheme under the same conditions. We argue that our representation outperforms gliding vertex representation in generality, accuracy, and efficiency for two reasons. The first is that the boxes from our representation are parallelograms, which are easier to post-process than arbitrary quadrilaterals from gliding vertex representation. Thus, our representation can be conveniently applied to different stages of detection. The other reason is that our midpoint offset representation is simpler than the gliding vertex representation, enjoying fewer parameters and being easier to optimize. As shown in Fig. 16, the regression error of our

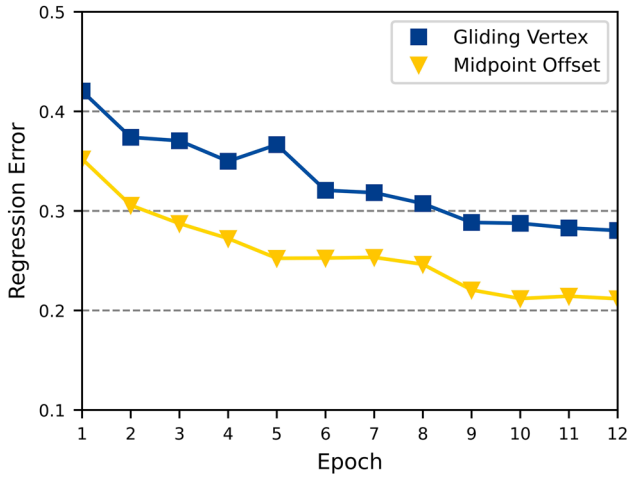

Fig. 16 Comparison of regression error between gliding vertex and midpoint offset representations during training

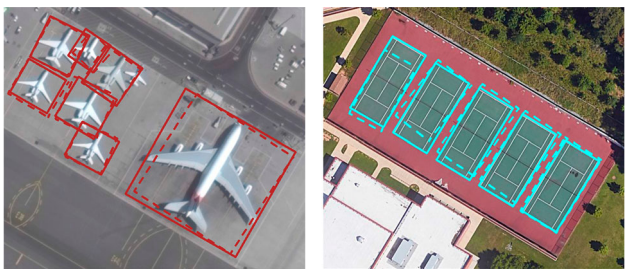

Fig. 17 Illustration of the regression process from the initial oriented proposals (dashed boxes) to the final oriented bounding boxes with solid lines

representation is smaller during training compared to gliding vertex representation and converges to a lower value.

In fact, our midpoint offset method can be applied to both Oriented RPN and Oriented R-CNN head (see Table 13). However, we observe that the difference in accuracy between different representations in the second stage is not obvious (75.87% vs. 75.90% mAP). We attribute the reason to the fact that the quality of the oriented proposals generated by Oriented RPN using midpoint offset representation is already quite promising in orientation prediction. In this case, Oriented R-CNN head requires only minor adjustments to the orientations of proposals (see Fig. 17), and the representation method has a very little impact on Oriented R-CNN head. Therefore, we adopt a simpler and more common five-parameter representation manner (x, y, w, h, θ) for Oriented R-CNN head to form a more generalized baseline.

5.5.4 Speed Versus Accuracy

To investigate the balance between detection speed and detection accuracy, we compare Oriented R-CNN to 5 representative two-stage detectors including Faster R-CNN-O, Mask R-CNN-O, HTC-O, Rotated RPN and RoI Transformer. Table 11 reports the comparison results on the DOTA-v1.0 test set, measured in terms of mAP and FPS. Here, 2000 and 1000 stand for the numbers of proposals.

As reported in Table 14: (i) Oriented R-CNN outperforms all comparison methods in both terms of accuracy and speed. In particular, compared with the strongest competitor RoI Transformer, our method surpasses it by 1.26% (2.48%) mAP and speeds up by 30.08% (22.40%) under the proposal numbers of 2000 (1000). (ii) Our method has the same mAP (75.87%) for both the proposal numbers of 2000 and 1000, but with a faster speed of 15.3 FPS when taking 1000 proposals as input. Therefore, in order to trade-off the inference speed and detection accuracy, we choose 1000 proposals from Oriented R-CNN as the input of oriented detection head. For other methods, since there exist obvious decline in accuracy, all of them adopt 2000 proposals as input, which is consistent with the settings of the publications. All these suggest the superiority of our Oriented RPN once again.

Table 14 Speed versus accuracy on the DOTA-v1.0 dataset

	2000 proposals		1000 proposals	
	mAP	FPS	mAP	FPS
Faster R-CNN-O (Ren et al., 2017)	69.05	13.9	68.79	14.3
Mask R-CNN-O (He et al., 2017)	70.71	6.9	70.39	7.8
Rotated RPN (Ma et al., 2018)	60.45	12.8	59.79	13.4
HTC-O (Chen et al., 2019)	71.21	6.1	70.57	7.2
RoI transformer (Ding et al., 2019)	74.61	11.3	73.39	12.5
Oriented R-CNN	75.87	14.7	75.87	15.3

Table 15 Results of Oriented R-CNN with advanced components

Method	CAS	P-NMS	#A=1	mAP
	✗	✗	✗	75.87
	✓	✗	✗	76.68
	✗	✓	✗	76.28
	✗	✗	✓	76.07
	✓	✓	✗	76.98
	✓	✗	✓	76.81
	✗	✓	✓	76.39
Oriented R-CNN	✓	✓	✓	77.16

Here, CAS, P-NMS and #A=1 denote cascade structure, poly NMS of Oriented RPN, and tiling one anchor per location, respectively.

Table 16 Horizontal proposals versus oriented proposals for aerial instance segmentation on the iSAID dataset

Method	AP	AP_{50}
Mask R-CNN (horizontal proposals)	25.65	51.30
Oriented mask R-CNN (oriented proposals)	39.22	63.43

5.5.5 Combination with Other Advanced Components

To further improve our detection model, we explore some advanced components including cascade structure (CAS for short), poly NMS of Oriented RPN (P-NMS for short), and tiling one anchor per location (#A=1 for short). Here, CAS means adding one additional stage after the Oriented R-CNN's detection head to refine bounding box locations. P-NMS refers to replacing the horizontal NMS originally used in Oriented RPN with poly NMS, and #A=1 represents that we set a square anchor per location and use adaptive training sample selection (ATSS) (Zhang et al., 2020a). Table 15 presents the results of our Oriented R-CNN using three advanced components above. Applying CAS to Oriented R-CNN, the accuracy is improved from 75.87 to 76.68% mAP with an absolute gain of 0.81%. By utilizing poly NMS, the mAP is increased to 76.28%. When we set one anchor for each position and use ATSS in Oriented RPN, an accuracy gain occurs, and the process of oriented proposal generation is further simplified without placing dense anchors. In addition, these advanced components can be combined to further improve detection accuracy (see the fifth to eighth rows of Table 15). With all three improvement strategies, Oriented R-CNN achieves 77.16% mAP, which is very competitive in the single-scale training/testing manner and serves as higher baselines for future research.

5.5.6 Oriented Proposals for Instance Segmentation

To explore the contribution of oriented proposals, we report the results when using horizontal proposals and oriented proposals for aerial instance segmentation, respectively. Here, the aerial instance segmentation based on horizontal proposals is implemented by Mask R-CNN. For oriented proposal-based aerial instance segmentation, we realize it by replacing the conventional RPN with Oriented RPN. Note that we conduct the experiments on the iSAID dataset and ResNet101 is adopted as the backbone.

As shown in Table 16, our oriented proposal-based instance segmentation outperforms the horizontal proposal-based Mask R-CNN by the large margins of 13.57% and 12.13% in terms of AP and AP_{50}, respectively. This is a good proof that oriented proposals play a vital role in aerial instance segmentation. We argue that the obvious gains are mainly from the alleviation of feature misalignment between proposals and oriented objects. To be specific, oriented proposals can cover objects well and have high IoU values with them. On the one hand, the high-quality proposals can improve the instance segmentation quality because mask prediction is performed only within proposals. On the other hand, the RoI features corresponding to oriented proposals can align objects accurately and greatly boost the accuracy of classification. Due to mask score identical to that of classification, the improvement of classification also leads to a gain of instance segmentation.

6 Conclusion

In this paper, we identify the generation of oriented proposals as the primary obstacle that prevents two-stage detectors from achieving fast inference speed. To this end, we first designed a novel representation of oriented objects, called midpoint offset representation. With the representation, we proposed the Oriented RPN to produce high-quality proposals in a cost-free manner. Based on Oriented RPN, we presented a simple but effective two-stage oriented detectors, termed Oriented R-CNN. In addition, we extended our method to achieve Oriented Mask R-CNN for aerial instance segmentation, which could effectively alleviate the problem of feature misalignment. Our Oriented R-CNN achieves state-of-the-art accuracy on all seven benchmarks for oriented object detection, especially, with the fastest speed among all detectors. Moreover, we conducted extensive experiments to evaluate Oriented Mask R-CNN on the iSAID dataset for aerial instance segmentation. We hope our methods could serve as two solid baselines for oriented object detection and aerial instance segmentation.

Acknowledgements This work was supported in part by the National Science Foundation of China under Grants 62376223 and 62136007, in part by the Natural Science Basic Research Program of Shaanxi under Grants 2021JC-16 and 2023-JCZD-36, and in part by the Doctorate Foundation of Northwestern Polytechnical University under Grant CX2021082. We also thank Chunbo Lang for his valuable and constructive suggestions during the revision of the manuscript.

Data Availability The datasets generated during the current study are available in the DOTA repository (https://captain-whu.github.io/DOTA/), the DIOR-R repository (https://gcheng-nwpu.github.io/#Datasets), the HRSC2016 repository (https://sites.google.com/site/hrsc2016), and the iSAID repository (https://captain-whu.github.io/iSAID/). The source code is available in Github at https://github.com/jbwang1997/OBBDetection.

References

Bolya, D., Zhou, C., Xiao, F., et al. (2019). Yolact: Real-time instance segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp 9157–9166

Cai, Z., & Vasconcelos, N. (2021). Cascade r-CNN: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(5), 1483–1498.

Cao, J., Cholakkal, H., Anwer, R.M., et al. (2020). D2det: Towards high quality object detection and instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 11,485–11,494

Chen, K., Pang, J., Wang, J., et al. (2019). Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 4974–4983

Chen, Z., Chen, K., Lin, W., et al. (2020). PIoU Loss: Towards accurate oriented object detection in complex environments. In *Proceedings of the European Conference on Computer Vision*, pp 195–211

Cheng, G., Wang, J., Li, K., et al. (2022). Anchor-free oriented proposal generator for object detection. *IEEE Transactions on Geoscience and Remote Sensing, 60*, 1–11.

Cheng, G., Lai, P., Gao, D., et al. (2023a). Class attention network for image recognition. *Science China Information Sciences, 66*(3), 1–13.

Cheng, G., Lang, C., & Han, J. (2023b). Holistic prototype activation for few-shot segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(4), 4650–4666.

Cheng, G., Li, Q., Wang, G., et al. (2023c). SFRNet: Fine-grained oriented object recognition via separate feature refinement. *IEEE Transactions on Geoscience and Remote Sensing, 61*, 1–10.

Cheng, G., Yuan, X., Yao, X., et al. (2023d). Towards large-scale small object detection: Survey and benchmarks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(11), 13467–13488.

Deng, J., Dong, W., Socher, R., et al. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 248–255

Ding, J., Xue, N., Long, Y., et al. (2019). Learning RoI transformer for oriented object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 2849–2858

Ding, J., Xue, N., Xia, G. S., et al. (2021). Object detection in aerial images: A large-scale benchmark and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2021.3117983

Everingham, M., Van Gool, L., Williams, C. K., et al. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision, 88*(2), 303–338.

Everingham, M., Eslami, S. A., Van Gool, L., et al. (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision, 111*(1), 98–136.

Follmann, P., & König, R. (2019). Oriented boxes for accurate instance segmentation. arXiv preprint arXiv:1911.07732

Gao, S. H., Cheng, M. M., Zhao, K., et al. (2021). Res2net: A new multi-scale backbone architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(2), 652–662.

Guo, Z., Liu, C., Zhang, X., et al. (2021). Beyond bounding-box: Convex-hull feature adaptation for oriented and densely packed object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 8792–8801

Han, J., Ding, J., Li, J., et al. (2021). Align deep features for oriented object detection. *IEEE Transactions on Geoscience and Remote Sensing, 60*, 1–11.

Han, J., Ding, J., Xue, N., et al. (2021b). Redet: A rotation-equivariant detector for aerial object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 2786–2795

He, K., Zhang, X., Ren, S., et al. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 770–778

He, K., Gkioxari, G., Dollár, P., et al. (2017). Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pp 2961–2969

Hou, L., Lu, K., Xue, J., et al. (2022). Shape-adaptive selection and measurement for oriented object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp 923–932

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 7132–7141

Huang, G., Liu, Z., van der Maaten, L., et al. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 4700–4708

Huang, Z., Huang, L., Gong, Y., et al. (2019). Mask scoring r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 6409–6418

Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., et al. (2015). Icdar 2015 competition on robust reading. In *Proceedings of the International Conference on Document Analysis and Recognition*, pp 1156–1160

Lang, C., Cheng, G., Tu, B., et al. (2023). Few-shot segmentation via divide-and-conquer proxies. *International Journal of Computer Vision*. https://doi.org/10.1007/s11263-023-01886-8

Lang, C., Cheng, G., Tu, B., et al. (2023). Base and meta: A new perspective on few-shot segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(9), 10669–10686.

Li, J., Lin, Y., Liu, R., et al. (2021). RSCA: Real-time segmentation-based context-aware scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 2349–2358

Li, W., Chen, Y., Hu, K., et al. (2022). Oriented reppoints for aerial object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp 1829–1838

Li, Y., Hou, Q., Zheng, Z., et al. (2023). Large selective kernel network for remote sensing object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp 16,794–16,805

Liao, M., Zhu, Z., Shi, B., et al. (2018). Rotation-sensitive regression for oriented scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 5909–5918

Liao, M., Zou, Z., Wan, Z., et al. (2022). Real-time scene text detection with differentiable binarization and adaptive scale fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2022.3155612

Lin, T.Y., Dollár, P., Girshick, R., et al. (2017a). Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 2117–2125

Lin, T.Y., Goyal, P., Girshick, R., et al. (2017b). Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp 2980–2988

Liu, L., Ouyang, W., Wang, X., et al. (2020). Deep learning for generic object detection: A survey. *International Journal of Computer Vision, 128*(2), 261–318.

Liu, S., Qi, L., Qin, H., et al. (2018). Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 8759–8768

Liu, Z., Wang, H., Weng, L., et al. (2016). Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds. *IEEE Geoscience and Remote Sensing Letters, 13*(8), 1074–1078.

Long, S., Ruan, J., Zhang, W., et al. (2018). Textsnake: A flexible representation for detecting text of arbitrary shapes. In *Proceedings of the European Conference on Computer Vision*, pp 20–36

Lyu, P., Yao, C., Wu, W., et al. (2018). Multi-oriented scene text detection via corner localization and region segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 7553–7563

Ma, J., Shao, W., Ye, H., et al. (2018). Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia, 20*(11), 3111–3122.

Ming, Q., Zhou, Z., Miao, L., et al. (2021). Dynamic anchor learning for arbitrary-oriented object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp 2355–2363

Pan, X., Ren, Y., Sheng, K., et al. (2020). Dynamic refinement network for oriented and densely packed object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 11,207–11,216

Qian, W., Yang, X., Peng, S., et al. (2021). Learning modulated loss for rotated object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp 2458–2466

Ren, S., He, K., Girshick, R., et al. (2017). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1137–1149.

Sun, X., Wang, P., Yan, Z., et al. (2022). FAIR1M: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing, 184*, 116–130.

Tang, J., Yang, Z., Wang, Y., et al. (2019). Seglink++: Detecting dense and arbitrary-shaped scene text by instance-aware component grouping. *Pattern Recognition, 96*, 6954–6966.

Tian, Z., Shu, M., Lyu, P., et al. (2019). Learning shape-aware embedding for scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 4234–4243

Tian, Z., Shen, C., & Chen, H. (2020). Conditional convolutions for instance segmentation. In *Proceedings of the European Conference on Computer Vision*, pp 282–298

Wang, H., Lu, P., Zhang, H., et al. (2020a). All you need is boundary: Toward arbitrary-shaped text spotting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp 12,160–12,167

Wang, W., Xie, E., Li, X., et al. (2019). Shape robust text detection with progressive scale expansion network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 9336–9345

Wang, X., Kong, T., Shen, C., et al. (2020b). Solo: Segmenting objects by locations. In *Proceedings of the European Conference on Computer Vision*, pp 649–665

Waqas Zamir, S., Arora, A., Gupta, A., et al. (2019). isaid: A large-scale dataset for instance segmentation in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp 28–37

Xia, G.S., Bai, X., Ding, J., et al. (2018). Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 3974–3983

Xie, E., Sun, P., Song, X., et al. (2020). Polarmask: Single shot instance segmentation with polar representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 12,193–12,202

Xie, S., Girshick, R., Dollar, P., et al. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 1492–1500

Xie, X., Cheng, G., Wang, J., et al. (2021). Oriented r-cnn for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp 3520–3529

Xie, X., Cheng, G., Li, Q., et al. (2023). Fewer is more: Efficient object detection in large aerial images. *Science China Information Sciences*. https://doi.org/10.1007/s11432-022-3718-5

Xie, X., Lang, C., Miao, S., et al. (2023). Mutual-assistance learning for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2023.3319634

Xu, Y., Fu, M., Wang, Q., et al. (2021). Gliding vertex on the horizontal bounding box for multi-oriented object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(4), 1452–1459.

Yang, J., Liu, Q., & Zhang, K. (2017). Stacked hourglass network for robust facial landmark localisation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp 79–87

Yang, X., & Yan, J. (2020). Arbitrary-oriented object detection with circular smooth label. In *Proceedings of the European Conference on Computer Vision*, pp 677–694

Yang, X., Yang, J., Yan, J., et al. (2019). Scrdet: Towards more robust detection for small, cluttered and rotated objects. In *Proceedings of the IEEE International Conference on Computer Vision*, pp 8232–8241

Yang, X., Hou, L., Zhou, Y., et al. (2021a). Dense label encoding for boundary discontinuity free rotation detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 15,819–15,829

Yang, X., Liu, Q., Yan, J., et al. (2021b). R3Det: Refined single-stage detector with feature refinement for rotating object. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp 3163–3171

Yang, X., Yan, J., Ming, Q., et al. (2021c). Rethinking rotated object detection with Gaussian Wasserstein distance loss. In *Proceedings of the International Conference on Machine Learning*, pp 11,830–11,841

Yang, X., Yang, X., Yang, J., et al. (2021d). Learning high-precision bounding box for rotated object detection via Kullback-Leibler divergence. In *Proceedings of the Advances in Neural Information Processing Systems*

Yang, X., Zhou, Y., Zhang, G., et al. (2023). The KFIoU loss for rotated object detection. In *Proceedings of the International Conference on Learning Representation*

Yi, J., Wu, P., Liu, B., et al. (2021). Oriented object detection in aerial images with box boundary-aware vectors. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp 2150–2159

Zhang, S., Chi, C., Yao, Y., et al. (2020a). Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 9759–9768

Zhang, S.X., Zhu, X., Hou, J.B., et al. (2020b). Deep relational reasoning graph network for arbitrary shape text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 9699–9708

Zhou, X., Yao, C., Wen, H., et al. (2017). EAST: An efficient and accurate scene text detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 5551–5560

Zhou, X., Wang, D., Krähenbühl, P. (2019). Objects as points. arXiv preprint arXiv:1904.07850

Zhu, Y., Chen, J., Liang, L., et al. (2021). Fourier contour embedding for arbitrary-shaped text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 3123–3131

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com