Point RCNN: An Angle-Free Framework for Rotated Object Detection

Qiang Zhou*, Chaohui Yu*, Zhibin Wang†, Hao Li
Alibaba Group
{jianchong.zq,huakun.ych,zhibin.waz,lihao.lh}@alibaba-inc.com

Figure 1. Comparison of different methods for generating rotated RoI (RRoI). (a) Rotated RPN places multiple rotated anchors with different angles, scales, and aspect ratios. (b) RoI Transformer proposes a RRoI learner to model the RRoI from the horizontal RoI (HRoI) for each feature point based on 3 anchors. (c) Our proposed PointRPN generates accurate RRoI in an anchor-free and angle-free manner.

Abstract

Rotated object detection in aerial images is still challenging due to arbitrary orientations, large scale and aspect ratio variations, and extreme density of objects. Existing state-of-the-art rotated object detection methods mainly rely on angle-based detectors. However, angle regression can easily suffer from the long-standing boundary problem. To tackle this problem, we propose a purely angle-free framework for rotated object detection, called Point RCNN, which mainly consists of PointRPN and PointReg. In particular, PointRPN generates accurate rotated RoIs (RRoIs) by converting the learned representative points with a coarse-to-fine manner, which is motivated by RepPoints. Based on the learned RRoIs, PointReg performs corner points refinement for more accurate detection. In addition, aerial images are often severely unbalanced in categories, and existing methods almost ignore this issue. In this paper, we also experimentally verify that re-sampling the images of the rare categories will stabilize training and further improve the detection performance. Experiments demonstrate that our Point RCNN achieves the new state-of-the-art detection performance on commonly used aerial datasets, including DOTA-v1.0, DOTA-v1.5, and HRSC2016.

1. Introduction

Although object detection has achieved significant progress in natural images, it still remains challenging for rotated object detection in aerial images, due to the arbitrary orientations, large scale and aspect ratio variations, and extreme density of objects [40]. Rotated object detection aims at predicting a set of oriented bounding box (OBB) and the corresponding classes in an aerial image, which has been serving as an essential step in many applications, e.g., urban management, emergency rescue, precise agriculture [9]. Modern rotated object detectors can be divided into two categories in terms of the representation of OBB: angle-based detectors and angle-free detectors.

In angle-based detectors, an OBB of a rotated object is usually represented as a five-parameter \((x, y, w, h, \theta)\). Most existing state-of-the-art methods are angle-based detectors relying on two-stage RCNN frameworks [8, 16, 20, 41, 48]. Generally, these methods use an RPN to generate horizontal or rotated RoIs, then use a designed RoI pooling operator to extract features from these RoIs. Finally, an RCNN head is used to predict the OBB and the corresponding classes. Compared to two-stage detectors, one-stage angle-based detectors [15, 31, 44, 45, 52] directly regress the OBB and classify them based on dense anchors for efficiency.

However, angle-based detectors usually introduce a long-standing boundary discontinuity problem [43, 47] due to the periodicity of angle and exchange of edges. Moreover, the unit between \((x, y, w, h)\) and angle \(\theta\) of the five-parameter representation is not consistent. These obstacles will cause the training unstable and limit the performance.
In contrast with angle-based detectors, angle-free detectors usually represent a rotated object as an eight-parameter OBB \((x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)\), which denotes the four corner points of a rotated object. Modern angle-free detectors \([1, 33, 42, 51]\) directly perform quadrilateral regression, which is more straightforward than the angle-based representation. Unfortunately, although abandoning angle regression and the parameter unit is consistent as well, the performance of existing angle-free detectors is still relatively limited. How to design a more straightforward and effective framework to alleviate the boundary discontinuity problem is key to the success of rotated object detectors.

In this paper, we propose a purely angle-free framework for rotated object detection, called Point RCNN, which can alleviate the boundary discontinuity problem and attain state-of-the-art performance. Concretely, Point RCNN is a two-stage detector and mainly consists of an RPN (PointRPN) and an RCNN head (PointReg), which are both angle-free. Given an input feature map, first, PointRPN learns a set of representative points for each feature point with a coarse-to-fine manner. Then, rotated RoI (RRoI) is generated through the \texttt{MinAreaRect} function of OpenCV \([2]\). Finally, PointReg applies a rotate RoI Align \([8, 16]\) operator to extract RRoI features, and then refines and classifies the eight-parameter OBB of corner points. In addition, the existing methods almost ignore the category imbalance in aerial images, and we propose to resample images of rare categories to stabilize convergence during training.

The contributions of this paper are as follows:

- We propose Point RCNN, a purely angle-free framework for rotated object detection. Without introducing angle prediction, Point RCNN is able to address the boundary discontinuity problem.
- We propose PointRPN and PointReg to reformulate angle prediction as the more straightforward points regression. Both of them are angle-free and have consistent parameter units. We further propose to resample images of rare categories to stabilize training and improve overall performance.
- Compared with the state-of-the-art methods, our Point RCNN framework attains higher detection performance on several large-scale datasets.

2. Related Work

2.1. Horizontal Object Detection

In the past decade, object detection has become an important computer vision task and has received considerable attention. One line of research focuses on two-stage detectors \([3, 11, 12, 17, 18, 25, 35]\), which first generates a sparse set of Regions of Interests (RoI) with a Region Proposal Network (RPN), and then performs classification and bounding box regression. While two-stage detectors still attract much attention, another line of research tends to develop efficient one-stage detectors \([10, 22, 26, 28, 34, 37, 50]\), in which SSD \([28]\) and YOLO \([34]\) are the fundamental methods that use a set of pre-defined anchor boxes to predict object category and anchor box offsets. Recently, some anchor-free methods \([10, 22, 50]\) detect object by predicting the center or corner or representative points, which also inspire us to develop the angle-free detector for rotated object.

2.2. Rotated Object Detection

In terms of the representation of oriented bounding box (OBB), modern rotated object detectors can be mainly divided into two categories: angle-based detectors and angle-free detectors.

Angle-based detectors detect rotated object by learning a five-parameter OBB \((x, y, w, h, \theta)\), in which \((x, y, w, h)\) denotes a horizontal bounding box and \(\theta\) denotes the rotated angle between the longer edge and the horizontal axis. RRPN \([31]\) and R\(^2\)PN \([52]\) make use of multiple rotated anchors with different angles, scales, and aspect ratios, which improves the performance while increasing the computational complexity (see Fig. 1(a)). R\(^2\)CNN \([20]\) proposes to detect horizontal and rotated bounding box simultaneously with multi task learning. RoI Transformer \([8]\) proposes a rotated RoI (RROi) learner to transform a horizontal RoI into a RROi, which provides more accurate RROIs with a complex pipeline (see Fig. 1(b)). SCRDet \([48]\) enhances features with attention module and proposes an \(\text{IoU-smooth} \quad L_1\) loss to alleviate the loss discontinuity issue. CSL \([45]\) reformulates angle prediction from regression to classification to alleviate discontinuous boundary problem. GWD \([46]\) and KLD \([49]\) propose more efficient loss function for OBB regression. S\(^2\)A-Net \([15]\) proposes a single-shot alignment network to realize full feature alignment and alleviates the inconsistency between regression and classification. Recently, ReDet \([16]\) proposes to use rotation-equivariant network to encode rotation equivariance explicitly and presents rotation-invariant RoI Align to extract rotation-invariant features. Oriented R-CNN \([41]\) proposes a two-stage detector that consists of an oriented RPN for generating RROi and an oriented RCNN for refining the RROi. Both ReDet and Oriented RCNN provide promising accuracy.

However, the boundary problem in the angle regression learning still causes training unstable and limits the performance. While angle-based detectors still find many applications, angle-free methods are getting more and more attention from the community.

Angle-free detectors reformulate rotated object
regression as learning a eight-parameter OBB \((x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)\), which represents the four corner points of a rotated object. ICN [1] proposes to directly estimate the four vertices of a quadrilateral to regress an oriented object based on image pyramid and feature pyramid. RSDet [33] and Gliding Vertex [42] achieve more accurate rotated object detection via directly quadrilateral regression prediction. Recently, BBA Vectors [51] extends the horizontal keypoint-based object detector to the oriented object detection task. CFA [13] proposes a convex-hull feature adaptation approach for configuring convolutional features. Compared to angle-based methods, angle-free detectors are more straightforward and can alleviate the boundary problem to a large extent. However, the performance is relatively limited yet.

In this paper, we propose an effective angle-free framework for rotated object detection, i.e., Point RCNN, which mainly consists of PointRPN and PointReg. Compared with other RRoI generation methods, our PointRPN generates accurate RRoI in an anchor-free and angle-free manner (see Fig. 1(c)).

3. Point RCNN

The overall structure of our Point RCNN is depicted in Fig. 2. We start by revisiting the boundary discontinuity problem of angle-based detectors. Then, we describe the overall pipeline of Point RCNN. Finally, we elaborate the PointRPN and PointReg modules, and propose a balanced dataset strategy to rebalance the long-tailed datasets during training.

3.1. Boundary Discontinuity

Boundary problem [43, 47] is a long-standing problem that existed in angle-based detectors. Take the commonly used five-parameter OBB representation \((x, y, w, h, \theta)\) as an example, where \((x, y)\) represents the center coordinates, \((w, h)\) represents the shorter and longer edges of the box, and \(\theta\) represents the angle between the longer edge and the horizontal axis. As shown in Fig. 3, when the target box is approximately square, a slight variation in edge length may cause \(w\) and \(h\) to swap, leading to a substantial variation of \(\pi/2\) in angle \(\theta\).

This boundary discontinuity issue in angle prediction will confuse the optimization of the network and limit the detection performance.

3.2. Overview

The overall pipeline of Point RCNN is shown in Fig. 2. During training, Backbone-FPN first extracts feature maps given an input image. Then, PointRPN performs representative points regression and generates pseudo OBB for rotated RoI (RRoI). Finally, for each RRoI, PointReg refines the corner points and classifies them for final detection results. Besides, we propose to resample images of rare categories to stabilize training and improve the overall performance.

The overall training objective is described as:

\[
\mathcal{L} = \mathcal{L}_{\text{PointRPN}} + \mathcal{L}_{\text{PointReg}}, \tag{1}
\]
where $\mathcal{L}_{\text{PointRPN}}$ denotes the losses in PointRPN, and $\mathcal{L}_{\text{PointReg}}$ denotes the losses in PointReg. We will describe them in detail in the following sections.

### 3.3. PointRPN

Existing rotated object detection methods generate rotated proposals indirectly by transforming the outputs of RPN [36] and suffer from the boundary discontinuity problem caused by angle prediction. For example, [8, 16] use RoI transformer to convert horizontal proposals to rotated proposals with an additional angle prediction task. Unlike these methods, in this paper, we propose to directly predict the rotated proposals with representative points learning. The learning of points is more flexible, and the distribution of points can reflect the angle and size of the rotated object. The boundary discontinuity problem can thus be alleviated without angle regression.

**Representative Points Prediction.** Inspired by RepPoints [50] and CFA [13], we propose PointRPN to predict the representative points in the RPN stage. The predicted points can effectively represent the rotating box and can be easily converted to rotated proposals in subsequent RCNN stages.

As shown in Fig. 4, PointRPN learns a set of representative points for each feature point. In order to make the features better adapt to the representative points learning, we take a coarse-to-fine prediction manner. In this way, the features will be refined with DCN [7] and the predicted offset in the initial stage. For each feature point, the predicted representative points of the two stages are:

$$
\mathcal{R}^{\text{init}} = \{(x_{i}^{0}, y_{i}^{0})\}_{i=1}^{K},
$$

$$
\mathcal{R}^{\text{refine}} = \{(x_{i}^{1}, y_{i}^{1})\}_{i=1}^{K},
$$

where $K$ denotes the number of predicted representative points and we set $K = 9$ by default. $\{(x_{i}^{0}, y_{i}^{0})\}_{i=1}^{K}$ denotes the initial location, $\{(\Delta x_{i}^{1}, \Delta y_{i}^{1})\}_{i=1}^{K}$ denote the learned offsets in the initial stage, and $\{(\Delta x_{i}^{1}, \Delta y_{i}^{1})\}_{i=1}^{K}$ denote the learned offsets in the refine stage.

**Label Assignment.** PointRPN predicts representative points for each feature point in the initial and refine stages. This section will describe how we determine the positive samples among all feature points for these two stages.

For the initial stage, we project each ground-truth box to the corresponding feature level according to its area, and then select the feature point closest to its center as the positive sample. The rule used for projecting the ground-truth box $b_i^*$ to the corresponding feature level is defined as:

$$
l_i = \log_2 \left( \frac{w_i h_i}{s} \right),
$$

where $s$ is a hyper-parameter and is set to 16 by default. $w_i$ and $h_i$ are the width and height of $b_i^*$.

For the refine stage, we use the predicted representative points from the initial stage to help determine the positive samples. Specifically, for each feature point with its corresponding prediction $\mathcal{R}^{\text{init}}$, if the maximum convex-hull IoU (defined in Eq. (6)) between $\mathcal{R}^{\text{init}}$ and ground-truth boxes exceeds the threshold $\tau$, we select this feature point as a positive sample. We set $\tau = 0.1$ in all our experiments.

**Optimization.** The optimization of the proposed PointRPN is driven by classification loss and rotated object localization loss. The learning objective is formulated as:

$$
\mathcal{L}_{\text{PointRPN}} = \lambda_1 + \mathcal{L}_{\text{loc}}^{\text{init}} + \lambda_2 \mathcal{L}_{\text{cls}}^{\text{refine}} + \lambda_3 \mathcal{L}_{\text{loc}}^{\text{refine}},
$$

where $\lambda_1, \lambda_2$, and $\lambda_3$ are the trade-off parameters and are set to 0.5, 1.0, and 1.0 by default, respectively. $\mathcal{L}_{\text{loc}}^{\text{init}}$ denotes the localization loss of the initial stage. $\mathcal{L}_{\text{cls}}^{\text{refine}}$ and $\mathcal{L}_{\text{loc}}^{\text{refine}}$ denote the classification loss and localization loss of the refine stage. Note that the classification loss is only calculated in the refine stage, and the two localization losses are only calculated for the positive samples.

In the initial stage, the localization loss is conducted between the convex-hulls converted from the learned points $\mathcal{R}^{\text{init}}$ (see initial stage in Fig. 4) and the ground-truth OBBS. We use convex-hull GIoU loss [13] to calculate the localization loss:

$$
\mathcal{L}_{\text{loc}}^{\text{init}} = \frac{1}{N_{\text{pos}}} \sum_{i} \left( 1 - \text{CIoU}(\Gamma(\mathcal{R}^{\text{init}}_i), \Gamma(b^*_i)) \right),
$$

where $N_{\text{pos}}$ indicates the number of positive samples of the initial stage. $b_i^*$ is the matched ground-truth OBB. CIoU represents the convex-hull GIoU between the two convex-hulls $\Gamma(\mathcal{R}^{\text{init}}_i)$ and $\Gamma(b^*_i)$, which is differential and can be
The rotated proposals generated by Corner Points Refine.

3.4. PointReg

Corner Points Refine. The rotated proposals generated by PointRPN already provide a reasonable estimate for the target rotated objects. To avoid the problems caused by angle regression and further improve the performance, we turn to refine the four corners of the rotated proposals in the RCNN stage. As shown in Fig. 5, with the rotated proposals as input, we use a RRoI feature extractor [8, 16] to extract RRoI features. Then, given the RRoI features, two consecutive fully-connected and ReLU layers are used to encode the RRoI features. Finally, two fully-connected layers are responsible for predicting the class probability $P$ and refined corners $C$ of the corresponding rotated object. The refined corners are represented as:

$$C = \{(x_i + \Delta x_i, y_i + \Delta y_i)\}^4_{i=1},$$

where $\{(x_i, y_i)\}^4_{i=1}$ denotes the corner coordinates of the input rotated proposals. $\{((\Delta x_i, \Delta y_i))\}^4_{i=1}$ denotes the predicted corner offsets.

Instead of directly performing angle prediction, we refine the four corners of the input rotated proposals. There are three advantages of adopting corner points refinement: 1) it can alleviate the boundary discontinuity problem caused by angle prediction; 2) the parameter units are consistent among the eight parameters; 3) it is possible to improve the localization accuracy using a coarse-to-fine manner.

We can easily extend PointReg to cascade structure for better performance. As shown in Fig. 2, in the cascade structure, the refined rotated proposals of the previous stage are used as the input of the current stage.

Optimization. The learning of PointReg is driven by classification loss and rotated object localization loss:

$$\mathcal{L}_{\text{PointReg}} = \mu_1 \mathcal{L}_{\text{cls}} + \mu_2 \cdot \mathcal{L}_{\text{loc}},$$

where $\mu_1$ and $\mu_2$ are the trade-off coefficients and are both set to 1.0 by default. $\mathcal{L}_{\text{cls}}$ indicates the classification loss, which is a cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{N} \sum_{i=0}^{C} Y_{i \rightarrow c} \log(P_i),$$

where $N$ denotes the number of training samples in PointReg, $C$ is the number of categories excluding background, $P_i$ is the predicted classification probability of the $i_{th}$ RRoI. $Y_{i \rightarrow c} = 1$ if the ground-truth class of the $i_{th}$ RRoI is $c$; otherwise it is 0. $\mathcal{L}_{\text{loc}}$ represents the localization loss between the refined corners and the corners of ground-truth OBB. We use $L_1$ loss to optimize the corner refinement learning:

$$\mathcal{L}_{\text{loc}} = \frac{1}{N} \sum_{i} |C_i - \vartheta(b^*_i)|,$$

$$\mathcal{L}_{\text{cls}} = \frac{1}{N} \sum_{i} \mathcal{L}_{\text{cls}}.$$

The rotated proposals generated by PointRPN already provide a reasonable estimate for the target rotated objects. To avoid the problems caused by angle regression and further improve the performance, we turn to refine the four corners of the rotated proposals in the RCNN stage. As shown in Fig. 5, with the rotated proposals as input, we use a RRoI feature extractor [8, 16] to extract RRoI features. Then, given the RRoI features, two consecutive fully-connected and ReLU layers are used to encode the RRoI features. Finally, two fully-connected layers are responsible for predicting the class probability $P$ and refined corners $C$ of the corresponding rotated object. The refined corners are represented as:

$$C = \{(x_i + \Delta x_i, y_i + \Delta y_i)\}^4_{i=1},$$

where $\{(x_i, y_i)\}^4_{i=1}$ denotes the corner coordinates of the input rotated proposals. $\{((\Delta x_i, \Delta y_i))\}^4_{i=1}$ denotes the predicted corner offsets.

Instead of directly performing angle prediction, we refine the four corners of the input rotated proposals. There are three advantages of adopting corner points refinement: 1) it can alleviate the boundary discontinuity problem caused by angle prediction; 2) the parameter units are consistent among the eight parameters; 3) it is possible to improve the localization accuracy using a coarse-to-fine manner.

We can easily extend PointReg to cascade structure for better performance. As shown in Fig. 2, in the cascade structure, the refined rotated proposals of the previous stage are used as the input of the current stage.

Optimization. The learning of PointReg is driven by classification loss and rotated object localization loss:

$$\mathcal{L}_{\text{PointReg}} = \mu_1 \mathcal{L}_{\text{cls}} + \mu_2 \cdot \mathcal{L}_{\text{loc}},$$

where $\mu_1$ and $\mu_2$ are the trade-off coefficients and are both set to 1.0 by default. $\mathcal{L}_{\text{cls}}$ indicates the classification loss, which is a cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{N} \sum_{i=0}^{C} Y_{i \rightarrow c} \log(P_i),$$

where $N$ denotes the number of training samples in PointReg, $C$ is the number of categories excluding background, $P_i$ is the predicted classification probability of the $i_{th}$ RRoI. $Y_{i \rightarrow c} = 1$ if the ground-truth class of the $i_{th}$ RRoI is $c$; otherwise it is 0. $\mathcal{L}_{\text{loc}}$ represents the localization loss between the refined corners and the corners of ground-truth OBB. We use $L_1$ loss to optimize the corner refinement learning:

$$\mathcal{L}_{\text{loc}} = \frac{1}{N} \sum_{i} |C_i - \vartheta(b^*_i)|,$$
where \( C_i = \{ (x_j, y_j) \}_{j=1}^4 \) denotes refined corners for the \( i_{th} \) rotated proposal, \( b_i^r = \{ (x_j^r, y_j^r) \}_{j=1}^4 \) denotes the corners of matched ground-truth OBB. \( \hat{\vartheta}(b_i^r) \) denotes the permutation of four corners of \( b_i^r \) with the smallest \( L_1 \) loss \( |C_i - \hat{\vartheta}(b_i^r)| \). Note that \( +L_{loc} \) is only calculated for positive training samples.

### 3.5. Balanced Dataset

The extremely nonuniform object densities of aerial images usually make the dataset long-tailed, which may cause the training process to be unstable and limit the detection performance. For instance, DOTA-v1.0 contains 52, 516 ship instances but only 678 ground track field instances [9]. To alleviate this issue, we resample the images of rare categories, which is inspired by [14]. More concretely, first, for each category \( c \in C \), compute the fraction of images \( F_c \) that contains this category. Then, compute the category-level repeat factor for each category:

\[
r_c = \max(1.0, \sqrt{\beta_{thr}/F_c}), \tag{14}
\]

where \( \beta_{thr} \) is a threshold which indicates that there will be not oversampling if \( "F_c > \beta_{thr}" \). Finally, compute the image-level repeat factor \( r_I \) for each image \( I \):

\[
r_I = \max_{c \in C_I}(r_c), \tag{15}
\]

where \( C_I \) denotes the categories contained in \( I \). In other words, those images that contain long-tailed categories will have a greater chance of being resampled during training.

### 4. Experiment

#### 4.1. Datasets

To evaluate the effectiveness of our proposed Point RCNN framework, we perform experiments on two popular large-scale datasets: DOTA [40] and HRSC2016 [30].

DOTA [40] is the largest dataset for oriented object detection with three released versions: DOTA-v1.0, DOTA-v1.5 and DOTA-v2.0. To compare the performance with the state-of-the-art methods, we perform experiments on DOTA-v1.0 and DOTA-v1.5. DOTA-v1.0 contains 2806 images ranging in size from \( 800 \times 800 \) to \( 4000 \times 4000 \), and contains 188, 282 instances in 15 categories: 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). DOTA-v1.5 has the same images as DOTA-v1.0 but contains 402, 089 instances. This is a more challenging dataset, which introduces a new category Container Crane (CC) and more small instances.

HRSC2016 [30] contains 1061 aerial images with size ranges from \( 300 \times 300 \) to \( 1500 \times 900 \). There are 436, 181, and 444 images in the training, validation and test set, respectively.

#### 4.2. Implementation Details

We implement Point RCNN using the MMDetection tool-box [6]. We follow ReDet [16] to use ReResNet with ReFPN as our backbone (ReR50-ReFPN), which has shown the ability to extract rotation-equivariant features. We also verify with the more generalized transformer backbone (Swin-Tiny) to show the generalization and scalability of our Point RCNN.

On the DOTA dataset, following previous methods [8, 15, 16], we crop the image to \( 1024 \times 1024 \) with 824 pixels as a stride and also we resize the image to three scales \( \{0, 5, 1.0, 1.5\} \) for multi-scale data. Random horizontal flipping and random rotation (\( \{-45^\circ, 45^\circ\}\)) are adopted for multi-scale training. On the HRSC2016 dataset, like previous method [16], we resize all the images to \( (800, 512) \), random horizontal flipping is applied during training. Unless otherwise specified, we train all the models with 19 epochs for DOTA and 36 epochs for HRSC2016. To be specific, we train the models using AdamW [21] on 8 Tesla-V100 GPUs with \( \beta_1=0.9 \) and \( \beta_2=0.999 \), with an initial learning rate of 0.0002, a weight decay of 0.05, and a mini-batch size of 16 (2 images per GPU). The learning rate decays by a factor of 10 at each decay step.

#### 4.3. Main Results

We compare our Point RCNN framework with other state-of-the-art methods on three datasets: DOTA-v1.0, DOTA-v1.5, and HRSC2016. As shown in Tab. 1, Tab. 2, and Tab. 3, without bells and whistles, our Point RCNN demonstrates superior performance against state-of-the-art methods.

On DOTA-v1.0, as reported in Tab. 1, Point RCNN achieves state-of-the-art \( 80.71 \) mAP. With the more generalized transformer backbone Swin-Tiny [29] (Swin-T), Point RCNN can further improve the performance by \( 0.61\% \) (from 80.71 to 81.32).

On DOTA-v1.5, which is more challenging compared to DOTA-v1.0, Point RCNN achieves \( 79.31 \) mAP, which significantly improve the performance by \( 2.51\% \). With the more generalized transformer backbone Swin-T, Point RCNN further improves the performance by \( 0.83\% \) (from 79.31 to 80.14). The results are reported in Tab. 2.

On HRSC2016, as reported in Tab. 3, Point RCNN attains the new state-of-the-art performance under both the VOC2007 and VOC2012 metrics, respectively.
Table 1. Performance comparisons on DOTA-v1.0 test set. * denotes multi-scale training and testing, † denotes the results of using balanced dataset strategy.

Table 2. Performance comparisons on DOTA-v1.5 test set. * denotes multi-scale training and testing, † denotes the results of using balanced dataset strategy.

Table 3. Performance comparisons on HRSC2016 test set. mAP$_{07}$ and mAP$_{12}$ indicate that the results are evaluated under VOC2007 and VOC2012 metrics(%) respectively. We report both results for fair comparisons.

4.4. Ablation Study

In this section, if not specified, all the models are trained only on the training and validation set with scale 1.0 for simplicity, and are tested using multi-scale testing. The metric mAP is evaluated on the DOTA-v1.5 test set and obtained by submitting prediction results to DOTA’s evaluation server.

4.4.1 Effect of PointRPN

To analyze the efficiency of PointRPN, we evaluate the detection recall of PointRPN on the validation set of DOTA-v1.5. For simplicity, we train the models on the training set with scale 1.0 and evaluate the metric recall on the validation set with scale 1.0 as well. The positive IoU threshold is set to 0.5. We select the top-300, top-1000, and top-2000 proposals to calculate their recall values and report the results in Tab. 4. We can find that when the proposals changes from top-2000 to top-1000, the recall value only drops 0.17%. Even if there are only top-300 proposals, the recall still achieves 85.93%.

4.4.2 Effect of Regression Type of PointReg

In this section, we analyze the effectiveness of the OBB regression type of PointReg. The results are shown in Tab. 5.
compared to the five-parameter representation, the eight-parameter regression type brings higher performance.

| Regression type | mAP (%) |
|-----------------|---------|
| $(x, y, w, h, \theta)$ | 77.25 |
| $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$ | 77.60 |

Table 5. Analysis of the effectiveness of OBB regression type of PointReg. The metric mAP is evaluated on the DOTA-v1.5 test set.

### 4.4.3 Effect of Balanced Dataset

| Oversampling threshold ($\beta_{thr}$) | mAP (%) |
|---------------------------------------|---------|
| 0                                     | 73.52   |
| 0.1                                   | 76.49   |
| 0.2                                   | 77.44   |
| 0.3                                   | 77.60   |
| 0.4                                   | 77.48   |

Table 6. Comparison of detection accuracy by varying the oversampling threshold $\beta_{thr}$ of the balanced dataset strategy. The metric mAP is evaluated on the DOTA-v1.5 test set.

In this section, we analyze the impact of the oversampling threshold $\beta_{thr}$ of the balanced dataset strategy. As shown in Tab. 6, we achieve the best detection accuracy of 77.60% at $\beta_{thr} = 0.3$. Therefore, we set $\beta_{thr} = 0.3$ in all other experiments on DOTA.

### 4.4.4 Factor-by-factor Experiment

To explore the effectiveness of each module of our proposed Point RCNN framework, we conduct a factor-by-factor experiment on the proposed PointRPN, PointReg and balanced dataset strategy. The results are depicted in Tab. 7, each component has a positive effect, and all components are combined to obtain the best performance.

| Method                  | PointRPN | Balanced Dataset | PointReg | mAP (%) |
|-------------------------|----------|------------------|----------|---------|
| Baseline                |          |                  |          | 71.36   |
| Point RCNN              | ✓        |                  | ✓        | 74.17   |
|                         | ✓        |                  | ✓        | 74.22   |
|                         | ✓        |                  | ✓        | 77.25   |
|                         | ✓        |                  | ✓        | 77.60   |

Table 7. Factor-by-factor ablation experiments. The detection performance is evaluated on the test set of DOTA-v1.5 dataset.

### 4.4.5 Visualization Analysis

We visualize some detection results on DOTA-v1.0 test set. Fig. 6 shows some examples of the learned points of PointRPN, which indicates that PointRPN is capable of learning representative points of rotated object. Fig. 7 shows some final detection results of Point RCNN, the red points denote the corner points learned by PointReg and the colored OBBs converted by the MinAreaRect function of OpenCV are the final results.

### 4.5. Limitations

Although experiments substantiate the superiority of Point RCNN over state-of-the-art methods, our method does not perform well enough in some categories, e.g., PL (Plane), which needs to be further explored. Point RCNN also needs to use rotate NMS to remove duplicate results, which may mistakenly delete the TP. Transformer-based methods [4] may be the potential solutions, which will be the future work.
5. Conclusion

In this work, we revisit rotated object detection and propose a purely angle-free framework for rotated object detection, named Point RCNN, which mainly consists of a PointRPN for generating accurate RROIs, and a PointReg for refining corner points based on the generated RROIs. In addition, we propose a balanced dataset strategy to overcome the long-tailed distribution of different object classes in aerial images. Extensive experiments on several large-scale benchmarks demonstrate the significant superiority of our proposed framework against the state-of-the-arts.

References

[1] Seyed Majid Azimi, Eleonora Vig, Reza Bahmanyar, Marco Körner, and Peter Reintartz. Towards multi-class object detection in unconstrained remote sensing imagery. In Asian Conference on Computer Vision, pages 150–165. Springer, 2018.

[2] G. Bradski. The OpenCV Library. Dr. Dobb’s Journal of Software Tools, 2000.

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 6154–6162, 2018.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, ECCV, 2020.

[5] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4974–4983, 2019.

[6] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiuru Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MDetection: Open mmlab detection toolbox and benchmark. 2019.

[7] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In Proceedings of the IEEE international conference on computer vision, pages 764–773, 2017.

[8] Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu. Learning roi transformer for oriented object detection in aerial images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2849–2858, 2019.

[9] Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Micheal Ying Yang, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pellillo, et al. Object detection in aerial images: A large-scale benchmark and challenges. arXiv preprint arXiv:2102.12219, 2021.

[10] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qiming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In International Conference on Computer Vision (ICCV), pages 6569–6578, 2019.

[11] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 1440–1448, 2015.

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 580–587, 2014.

[13] Zonghao Guo, Chang Liu, Xiaosong Zhang, Jianbin Jiao, Xiaoyang Ji, and Qixiang Ye. Beyond bounding-box: Convex-hull feature adaptation for oriented and densely packed object detection. In CVPR, pages 8792–8801, June 2021.

[14] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In CVPR, 2019.

[15] Jiaming Han, Jian Ding, Jie Li, and Gui-Song Xia. Align deep features for oriented object detection. IEEE Transactions on Geoscience and Remote Sensing, 2021.

[16] Jiaming Han, Jian Ding, Nan Xue, and Gui-Song Xia. Redet: A rotation-equivariant detector for aerial object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2786–2795, 2021.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.

[18] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3588–3597, 2018.

[19] Ray A Jarvis. On the identification of the convex hull of a finite set of points in the plane. Information processing letters, 2(1):18–21, 1973.

[20] Yingying Jiang, Xiangyu Zhu, Xiaobing Wang, Shuli Yang, Wei Li, Hua Wang, Pei Fu, and Zhenbo Luo. R 2 cnn: Rotational region cnn for arbitrarily-oriented scene text detection. In 2018 24th International Conference on Pattern Recognition (ICPR), pages 3610–3615. IEEE, 2018.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, ICLR, 2015.

[22] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), pages 734–750, 2018.

[23] Chengzheng Li, Chunyan Xu, Zhen Cui, Dan Wang, Zeyun Jie, Tong Zhang, and Jian Yang. Learning object-wise semantic representation for detection in remote sensing imagery. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 20–27, 2019.

[24] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene...
[52] Zenghui Zhang, Weiwei Guo, Shengnan Zhu, and Wenxian Yu. Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks. *IEEE Geoscience and Remote Sensing Letters*, 15(11):1745–1749, 2018. 1, 2, 7