Theoretically Achieving Continuous Representation of Oriented Bounding Boxes

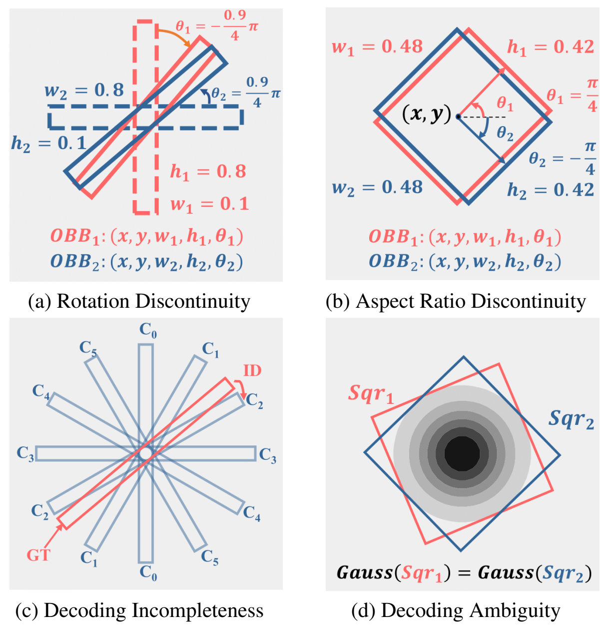
Zi-Kai Xiao[1], **Guo-Ye Yang**[1], **Xue Yang**[2], **Tai-Jiang Mu**[1*], **Junchi Yan**[2], **Shi-Min Hu**[1]
[1]BNRist, Department of Computer Science and Technology, Tsinghua University
[2]Department of CSE & MoE Key Lab of AI, Shanghai Jiao Tong University
{xzk23, yanggy19}@mails.tsinghua.edu.cn {taijiang, shimin}@tsinghua.edu.cn
{yangxue-2019-sjtu, yanjunchi}@sjtu.edu.cn
Code: https://github.com/514flowey/JDet-COBB

Abstract

*Considerable efforts have been devoted to Oriented Object Detection (OOD). However, one lasting issue regarding the discontinuity in Oriented Bounding Box (OBB) representation remains unresolved, which is an inherent bottleneck for extant OOD methods. This paper endeavors to completely solve this issue in a theoretically guaranteed manner and puts an end to the ad-hoc efforts in this direction. Prior studies typically can only address one of the two cases of discontinuity: rotation and aspect ratio, and often inadvertently introduce decoding discontinuity, e.g. Decoding Incompleteness (DI) and Decoding Ambiguity (DA) as discussed in literature. Specifically, we propose a novel representation method called **C**ontinuous **OBB** (**COBB**), which can be readily integrated into existing detectors e.g. Faster-RCNN as a plugin. It can theoretically ensure continuity in bounding box regression which to our best knowledge, has not been achieved in literature for rectangle-based object representation. For fairness and transparency of experiments, we have developed a modularized benchmark based on the open-source deep learning framework Jittor's detection toolbox JDet for OOD evaluation. On the popular DOTA dataset, by integrating Faster-RCNN as the same baseline model, our new method outperforms the peer method Gliding Vertex by 1.13% mAP$_{50}$ (relative improvement 1.54%), and 2.46% mAP$_{75}$ (relative improvement 5.91%), without any tricks.*

1. Introduction

Object detection constitutes a fundamental task within the realm of computer vision. In conventional object detection scenarios [52], the commonplace approach involves the localization of objects using Horizontal Bounding Boxes (HBB). However, in many real-world settings such as remote sensing [3, 32] and scene text [13, 19], where objects exhibit arbitrary orientations, HBBs are inadequate in precisely delineating object boundaries. To overcome this issue, Oriented Bounding Boxes (OBB) [32], conceptualized as rotated rectangles, have been introduced as a more suit-



(a) Rotation Discontinuity (b) Aspect Ratio Discontinuity (c) Decoding Incompleteness (d) Decoding Ambiguity

Figure 1. **Examples of Discontinuity in OBB Representations.** (a) Acute-angle Representation limits the rotation angle of OBBs inside a range of $\frac{\pi}{2}$ ($[-\frac{\pi}{4}, \frac{\pi}{4})$ in this example). The red OBB$_1$ and the blue OBB$_2$ are similar, but their representations are significantly different. (b) Long-edge Representation determines the rotation angle θ by the long side and the x-axis. A slight disturbance in the aspect ratio of square-like OBBs will cause a huge change in their representation, which causes Aspect Ratio Discontinuity. (c) CSL [37] divides the rotation angle into several classifications (6 classifications in this figure). OBB between two classifications cannot be accurately represented, which brings DI. (d) GWD [40] denotes OBBs by Gaussian distribution. As the squares with different rotation angles can correspond to the same Gaussian, the orientation of decoded squares will be ambiguous.

*Corresponding author.

able representation for Oriented Object Detection (OOD).

Various models have been proposed for OOD [2, 7, 17, 20, 35]. However, as illustrated in Figs. 1a-1b, prevalent representation methods for OBBs exhibit discontinuity issues, encoding similar OBBs into distinct vectors. This introduces challenges in training neural networks as regression targets for similar input features may differ significantly, potentially causing confusion and hindering the training process. The relationship between two OBBs can be conceptualized as one being transformed into the other through geometric operations: translation, rotation, scaling, and aspect ratio changes. While translation and scaling are relatively benign, rotation and aspect ratio changes are primary sources of discontinuity.

Rotation discontinuity, often referred to as the "Boundary Problem" [37] or "Rotation Sensitive Error" [23], stems from the periodicity of rotation angles. Although prior efforts [37, 39, 46] have addressed this, most of them still suffer from discontinuity arising from changes in aspect ratio. Some other methods, *e.g*. Gliding Vertex [35], effectively address aspect ratio discontinuity. Nonetheless, these techniques continue to face challenges in overcoming rotation discontinuity. The discontinuity phenomena emerge during the encoding of OBBs into the regression target. Consequently, they can be termed encoding discontinuity.

Additionally, existing methods aim at resolving the encoding discontinuity issue, yet meanwhile would often bring about decoding discontinuity, namely Decoding Incompleteness (DI) and Decoding Ambiguity (DA). DI arises when OBBs cannot be accurately represented, often attributed to angle discretization and classification as exemplified by CSL [37] in Fig. 1c. DA, on the other hand, pertains to instances where distinct OBBs share similar representations, rendering predicted OBBs sensitive to minor disturbances in the model's output, exemplified by GWD [40] in Fig. 1d. Fundamentally, DI and DA result in decoded OBBs lacking continuity concerning their representation. Hence, we categorize DI and DA as decoding discontinuities. The decoding discontinuities bring precision errors and directly degrade the prediction precision.

Due to the absence of a rigorous definition, prior approaches have often addressed discontinuity issues incompletely. To address this, we introduce formal continuity metrics, evaluating previous methods using these benchmarks. As a comprehensive solution to discontinuity problems, we propose **C**ontinuous **OBB** (**COBB**)—a novel, continuous OBB representation satisfying all defined metrics. COBB employs nine parameters derived from continuous functions based on the outer Horizontal Bounding Box (HBB) and OBB area. This ensures continuity as the outer HBB and OBB area undergo continuous changes during shape transformations. Our COBB can be easily integrated into existing OOD methods by simply replacing their original representations of OBB with ours.

We have developed a benchmark using the detection toolbox JDet of Jittor [10] which is an open-source deep learning framework friendly to vision tasks. In particular, a fair comparison across different models is made by aligning the data augmentation schemes and diverse techniques.

Our experiments on this benchmark demonstrate the effectiveness of COBB across diverse datasets and baseline detectors, particularly demonstrating advantages in high-precision object detection. Notably, it achieves a 3.95% improvement in mAP_{75} when applied to Faster R-CNN on the DOTA Dataset. Detailed results are presented in Sec. 5.

Our contributions encompass the following aspects:

- We systematically analyze the inherent discontinuity issues in existing OBB representation methods for OOD, and introduce formal metrics to assess their continuity.
- Building upon our findings, we introduce COBB, a fully continuous representation of OBBs.
- We construct a new benchmark for fair comparisons among OOD methods. Experiments on this benchmark validate the effectiveness of our approach, highlighting its advantages in high-precision OOD.

2. Related Work

2.1. Oriented Object Detection

With the increasing adoption of deep learning in computer vision, object detection models [5, 6, 11, 16, 24, 30, 53] have emerged to enhance computers' capacity for recognizing objects in natural images. Typically tailored for predicting HBBs, these models serve as the foundation for OOD when augmented with modules for OBB prediction. Rotated Faster R-CNN [25] stands as a prominent baseline for OOD, replacing HBB regression targets with OBBs. Several OOD methods, such as RoI Transformer [2], Gliding Vertex [35], ReDet [7], and Oriented R-CNN [33], follow a similar structure and can be implemented on the Rotated Faster R-CNN framework.

While many OOD models share structural similarities, detailed implementation differences exist (*e.g*. Gliding Vertex [35] using ResNet101 as the backbone network, whereas CSL [37] employs ResNet50). To facilitate fair comparisons, we established a uniform pipeline with modular alternatives for implementing these models, minimizing implementation disparities.

2.2. Discontinuity in Oriented Object Detection

Methods aiming to handle the discontinuous representation of OBBs fall into three categories: Loss Improvement, Angle Encoding, and New OBB representation.

Loss Imporvement. Modifying the loss is a direct way to mitigate sudden changes in loss values caused by encoding discontinuity. Approaches like RIL [21] and RSDet [23] propose loss functions that approach zero as the model's output converges to various representations of the ground truth OBB. PIoU [1] and SCRDet [38] incorporate Intersection over Union (IoU) between prediction results and regression targets in their loss. GWD [40], KLD [41], and KFIoU [44] convert OBBs into Gaussian distributions for IoU calculation, introducing potential DA for square-like objects. While showing empirical effectiveness in reducing the impact of discontinuity, these approaches do not provide a theoretical resolution to the problem.

Angle Encoding. Several methods focus on addressing the Periodicity of Angular (PoA), a primary cause of encoding discontinuity [37]. CSL [37] discretizes the rotation angle into a heavy regression target, with subsequent improvements by DCL [39], GF_CSL [29], MGAR [28], and AR-CSL [48]. While these methods enhance rotation continuity, most of them struggle with square-like objects and may introduce DI. PSC [46], FSTC [49], and ACM [34] encode the rotation angle into a continuous vector, yet they still exhibit discontinuity for square-like objects.

New OBB Representation. Other approaches explore alternative representations for OBBs instead of rectangles and rotation angles. Gliding Vertex [35] slides the four vertices of a HBB to construct an OBB. O^{2}D-Net [31] and BBAVectors [45] represent an OBB using its center point and vectors from the center point to midpoints of its sides. PolarDet [51] and CRB [47] leverage polar coordinates, yet the rotation discontinuity still exists. DHRec [22] represents OBBs with double horizontal rectangles but struggles with distinguishing symmetrical tilted thin OBBs.

To the best of our knowledge, no method achieves perfect elimination of discontinuity. Previous approaches either fail in specific boundary situations or introduce DI and DA. The proposed COBB in this paper provides the first completely continuous representation of OBBs.

Table 1. **Comparison of Methods Dedicated to Discontinuity**. Tar (R), Tar (A), Loss (R), and Loss (A) stand for Target Rotation Continuity, Target Aspect Ratio Continuity, Loss Rotation Continuity, and Loss Aspect Ratio Continuity, respectively. Dec (C) and Dec (R) stand for Decoding Completeness and Decoding Robustness. Acute-Angle is the common OBB representation that limits the rotation angle into a range of $\frac{\pi}{2}$ ($[-\frac{\pi}{4}, \frac{\pi}{4})$ for example). Long-Edge is another common OBB representation, which determines the rotation angle θ by the long side and the x-axis, and the θ is within a range of length π. Further explanation of this table is provided in the supplemental material, see Sec. 8.

Method	Tar (R)	Tar (A)	Loss (R)	Loss (A)	Dec (C)	Dec (R)
Acute-Angle	-	✓	-	✓	✓	✓
Long-Edge	-	-	-	-	✓	✓
RIL [21]	-	-	✓	✓	✓	✓
RSDet (l_{mr}^{5p}) [23]	-	-	✓	✓	✓	✓
PIoU [1]	-	-	✓	✓	✓	✓
SCRDet [38]	-	-	✓	✓	✓	✓
GWD [40]	✓	✓	✓	✓	✓	-
KLD [41]	✓	✓	✓	✓	✓	-
KFIoU [44]	✓	✓	✓	✓	✓	-
CSL [37]	✓	-	✓	-	-	-
DCL [39]	✓	-	✓	✓	-	-
GF_CSL [29]	✓	✓	✓	✓	-	-
MGAR [28]	-	-	✓	-	✓	✓
AR-CSL [48]	✓	-	✓	-	-	-
PSC [46]	✓	-	✓	-	✓	✓
FSTC [49]	✓	-	✓	-	✓	✓
ACM [34]	✓	-	✓	-	✓	✓
Gliding Vertex [35]	-	✓	-	✓	✓	-
O^{2}D-Net [31]	-	✓	-	✓	✓	✓
BBAVectors [45]	-	✓	-	✓	✓	-
PolarDet [51]	-	✓	-	✓	✓	✓
DHRec [22]	✓	✓	✓	✓	✓	-
CRB [47]	-	-	✓	-	✓	✓
Ours	✓	✓	✓	✓	✓	✓

Furthermore, some methods assert the resolution of the discontinuity, yet they still struggle with sudden changes in regression targets [34, 46].

To formally define continuity, we introduce f_{Enc} as the mapping function from an OBB to R^n, and f_{Dec} as the reverse mapping from a subset of R^n to an OBB. Notably, $R(x, \theta)$ denotes the transformation generating an OBB y by rotating the initial OBB x by θ in a clockwise direction. Meanwhile, $A(x, r)$ generates a set of OBBs $\{y, z\}$ by adjusting one side of OBB x to be r times its original length. We refer to S as the set of OBBs, and L symbolizes the loss function. Models utilize f_{Enc} to convert OBBs into regression targets and employ f_{Dec} to translate prediction results into estimated OBBs.

Target Rotation Continuity: Minor rotations should minimally affect the regression target.

$$\forall x \in S, \lim_{\theta \to 0} ||f_{Enc}(x) - f_{Enc}(R(x, \theta))|| = 0. \tag{1}$$

3. Theoretically Continuous Representation

In this section, we first introduce our devised metrics to assess the continuity of existing methods in Sec. 3.1. While Sec. 3.2 unveils our COBB that theoretically ensures continuity under these metrics. The continuity of COBB is rigorously demonstrated in Sec. 3.4, with comprehensive details provided in the supplemental material.

3.1. Metrics for Continuity

As depicted in Fig. 1, prevalent methods for OBB prediction commonly face the challenge of encoding discontinuity. Prior endeavors often address some specific boundary cases, such as nearly horizontal OBBs and square-like OBBs [35, 37, 46]. While these methods may exhibit continuity in certain boundary scenarios, they often overlook others. For instance, CSL [37] maintains rotation continuity for nearly horizontal OBBs but fails for square-like OBBs.

The corresponding discontinuity is often referred to as the "Boundary Problem" [37, 39].

Target Aspect Ratio Continuity: Slight changes in aspect ratio should minimally impact the regression target.

$$\forall x \in S, \lim_{r\to 1} \sum_{y\in A(x,r)} ||f_{Enc}(x) - f_{Enc}(y)|| = 0. \tag{2}$$

Loss Rotation Continuity: Small rotations should minimally affect the loss value.

$$\forall x \in S, \lim_{\theta\to 0} ||L\left(f_{Enc}(x), f_{Enc}\left(R(x,\theta)\right)\right)|| = 0. \tag{3}$$

Loss Aspect Ratio Continuity: Minor aspect ratio changes should minimally alter the loss value.

$$\forall x \in S, \lim_{r\to 1} \sum_{y\in A(x,r)} ||L\left(f_{Enc}(x), f_{Enc}(y)\right)|| = 0. \tag{4}$$

Decoding Completeness: Every OBB can be accurately represented.

$$\forall x \in S, \exists t \in R^n, IoU\left(x, f_{Dec}(t)\right) = 1. \tag{5}$$

Decoding Completeness is equivalent to avoiding Decoding Incompleteness (DI) illustrated in Fig. 1c.

Decoding Robustness: Decoded OBBs should be robust to slight errors in their representation.

$$\begin{aligned}&\forall x \in S, \forall \epsilon > 0, \exists \xi > 0, \forall \Delta d \in R^n \wedge ||\Delta d|| < \xi,\\ &1 - IoU\left(x, f_{Dec}\left(f_{Enc}(x) + \Delta d\right)\right) < \epsilon.\end{aligned} \tag{6}$$

Decoding Robustness is equivalent to avoiding Decoding Ambiguity (DA) illustrated in Fig. 1d.

Previous research has covered the first four metrics, which we formally defined, whereas there's been limited exploration of the last two metrics. Our investigation into existing OBB representation methods helped unveil the neglected discontinuity, which formed the basis for these two metrics. A further detailed explanation is provided in the supplemental material. Tab. 1 summarizes existing methods addressing discontinuity. However, these methods are not universally continuous. To comprehensively resolve the problem of discontinuity, we propose COBB, which ensures both encoding continuity and decoding continuity.

3.2. Our Continuous Representation for OBB

Note that the outer HBB and the area of an OBB undergo continuous changes during shape transformations. Consequently, we sought to represent an OBB with a 5-dimensional vector, (x_c, y_c, w, h, r_a). Here, (x_c, y_c), w, and h refer to the center point, width, and height of the outer HBB, respectively, while r_a is the acreage ratio of the OBB relative to its outer HBB.

It can be proven that only a pair of symmetrical OBBs shares the same (x_c, y_c, w, h, r_a) (detailed proof is provided in the supplemental material). However, directly computing OBBs from (x_c, y_c, w, h, r_a) is a complex process. To address this challenge, we introduce a sliding ratio, r_s, to estimate r_a, defined as follows.

$$r_s = \begin{cases} \frac{x_2 - x_1}{w} & w < h, \\ \frac{y_2 - y_1}{h} & w \geq h, \end{cases} \tag{7}$$

where the x-coordinates of four vertices of the OBB are sorted as x_1, x_2, x_3, x_4 from small to large, and y-coordinates are sorted as y_1, y_2, y_3, y_4. It can be proved that r_s can be computed as $r_s = f(\min(r_a, 1 - r_a))$, where $f : [0, 0.5] \to [0, 0.5]$ is a continuous strictly increasing map (proof provided in the supplemental material). This implies the r_s changes continuously as OBBs transform.

However, as shown in Fig. 2, a total of four different OBBs can be encoded into the same (x_c, y_c, w, h, r_s), leading to potential DA. To mitigate DA, we utilize Intersection over Unions (IoUs) between the target OBB and the four OBBs as scores for classification. Importantly, these IoUs can be directly computed using (x_c, y_c, w, h, r_s) and the classification of the target OBB, eliminating the need for complex computations involving IoU between arbitrary OBBs. The detailed computation process is provided in the supplemental material.

Finally, $(x_c, y_c, w, h, r_s, s_0, s_1, s_2, s_3)$ will be considered as a continuous representation of OBBs, where s_0, s_1, s_2, s_3 are IoUs between the target OBB and the four OBBs with the same (x_c, y_c, w, h, r_s).

By reversing the above process, a 9-dimensional vector is decoded into a single OBB. Without loss of generality, assuming $w \geq h$, exploiting the properties of similar triangles allows the computation of $x_2 - x_1$ and $y_2 - y_1$:

$$\begin{aligned} y_2 - y_1 &= r_s h, \\ x_2 - x_1 &= \frac{1 - \sqrt{1 - 4 \cdot \frac{h^2}{w^2} \cdot r_s(1 - r_s)}}{2} w. \end{aligned} \tag{8}$$

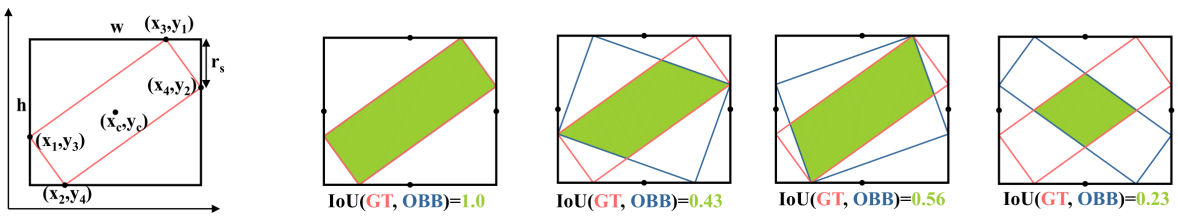
The classification with the highest IoU score determines the style of the generated OBB. Using the HBB, $x_2 - x_1$, $y_2 - y_1$, and the style, the coordinates of the OBB's four vertices can be easily computed. The detailed computation process is provided in supplemental material.

3.3. Implementing COBB in OOD Models

Most models use the bias between the ground truth and the assigned proposal as the regression target to take advantage of the priori information of proposals. In our method, for horizontal proposal region (x_p, y_p, w_p, h_p), the regression target is computed as follows, same as Faster R-CNN [25]:

$$t_x = \frac{\Delta x}{w_p},\ t_y = \frac{\Delta y}{h_p},\ t_w = \ln\left(\frac{w}{w_p}\right),\ t_h = \ln\left(\frac{h}{h_p}\right), \tag{9}$$

where $box_t = (t_x, t_y, t_w, t_h)$ is the target for the outer HBB, and $\Delta x = x_c - x_p$, $\Delta y = y_c - y_p$.



(a) Base Parameters

(b) Four types of OBBs with the same (x_c, y_c, w, h, r_s)

Figure 2. **Example of COBB**. COBB utilizes the outer HBB (x_c, y_c, w, h), sliding ratio r_s, and four IoU scores. (a) Example of the outer HBB and r_s. In this instance, $r_s = \frac{y_2 - y_1}{h}$ when $w > h$, where y_1 and y_2 denote the two smaller y-coordinates among the four vertices of the OBB. (b) Using x_c, y_c, w, h, and r_s, along with the properties of similar triangles, we can derive and solve a system of equations to obtain the parameters for four OBBs (details provided in the supplemental material). Distinguishing between these OBBs is guided by the positional relationship between their vertices and the midpoints on each side.

According to Eq. 7, the value of r_s lies within the range [0, 0.5]. To take advantage of this property, an effective way is to limit the range of prediction results, such as employing the sigmoid function. In this situation, the regression target for r_s is computed as follows:

$$r_{sig} = 2r_s. \tag{10}$$

Another method is extending the domain of r_s as follows:

$$r_{ln} = \begin{cases} 1 + \log_2(r_s) & r_a < 0.5, \\ 1 + \log_2(1 - r_s) & r_a \geq 0.5. \end{cases} \tag{11}$$

Compared with r_{sig}, r_{ln} exhibits increased sensitivity to r_s when r_a is small, aiding detectors in precisely predicting inclined thin objects. Based on the definition of the regression target of r_s, our methods fall into two classifications: COBB-sig and COBB-ln.

The regression target of IoU scores is defined as:

$$s_t = (s_0^\lambda, s_1^\lambda, s_2^\lambda, s_3^\lambda), \tag{12}$$

where λ is a predefined constant to amplify the gap between scores of the ground truth and other classifications.

For models employing oriented proposal regions, we rotate the proposal region and target OBB around the center of the proposal region until its rotation angle becomes zero. Subsequently, we calculate the regression target as that for horizontal proposal regions. By reversing this process, OBBs can be easily recovered from the regression target and the oriented proposal regions. The detailed computation method is provided in the supplemental material.

In our approach, the loss function is defined as:

$$\begin{aligned} L = & w_1 L_{cls} + w_2 L_{box}(box_p, box_t) \\ & + w_3 L_r(r_p, r_t) + w_4 L_s(s_p, s_t), \end{aligned} \tag{13}$$

where box_p, r_p, and s_p denote predicted outer HBBs, predicted r_s, and predicted IoU scores, respectively. r_t is the regression target of r_s, which is either r_{sig} for COBB-sig or r_{ln} for COBB-ln. L_{cls} stands for the classification loss, which aligns with that of the baseline model (*e.g.* cross-entropy loss for Faster R-CNN [25]). L_{box}, L_r, and L_s are Smooth L1 Loss [25]. The hyperparameters w_1, w_2, w_3, and w_4 are predefined constants.

3.4. Theoretical Guarantee of the Continuity

The COBB, as detailed in Sec. 3.2, theoretically ensures continuity under the metrics outlined in Sec. 3.1. Here, we briefly elucidate the reasons behind the continuity, with detailed proofs provided in the supplemental material.

Theoretical Analysis on Encoding Continuity: According to Sec. 3.2, x_c, y_c, w, h, and r_s exhibit continuity concerning the outer HBB and the area of the target OBB. For unambiguous classifications, IoU scores remain continuous concerning x_c, y_c, w, h, and r_s. In cases of ambiguous OBB classifications, the IoU scores remain similar regardless of the classification. Consequently, the regression target produced by our method maintains continuity for both rotation and aspect ratio changes.

Theoretical Analysis on Decoding Continuity: The OBB generation process in Sec. 3.2 ensures precise reversal from the 9 parameters, mitigating inherent DI errors.

To avoid DA, the decoder must resist slight changes in its input. When IoU scores are fixed, the decoded four vertices remain continuous concerning (x_c, y_c, w, h, r_s). Notably, when w is similar to h, the changes in the ordering of values between w and h do not lead to DA, as $x_2 - x_1$ is similar to $y_2 - y_1$. If x_c, y_c, w, h, and r_s are fixed, and a slight perturbation in IoU scores results in a classification error, the IoU between the OBB before and after perturbation decoding is close to 1, adhering to the definition of IoU scores. In summary, our method exhibits resistance to perturbations in predicted results, thereby avoiding DA.

3.5. Further Comparison with Peer Methods

Compared with Gliding Vertex: The Gliding Vertex method [35] represents an OBB by sliding the four vertices of its outer HBB. However, rotation continuity is compromised when the OBB is nearly horizontal. Moreover, its decoded results manifest as irregular quadrilaterals, and refining these into accurate OBBs introduces accuracy errors. In contrast, our methods ensure continuous prediction targets and loss values for nearly horizontal OBBs, and the decoded quadrilaterals consistently represent accurate OBBs.

Table 2. **Open source OOD benchmarks.**

Benchmark	ArialDet	OBBDet	AlphaRotate [42]	MMRotate [54]	JDet
DL library	PyTorch	PyTorch	Tensorflow	PyTorch	Jittor
Algorithm	5	10	18	19	20
Dataset	1	6	11	4	6

Compared with CSL-based methods: CSL-based methods [28, 29, 37, 39, 48] discretize rotation angles, converting angle regression into an angle classification problem to address rotation discontinuity. However, angle discretization introduces DI problems and results in a heavy prediction layer. Additionally, most CSL-based methods do not maintain continuity in aspect ratio changes when dealing with square-like OBBs. In contrast, our method ensures encoding continuity in both rotation and aspect ratio changes without introducing DI. Furthermore, our approach encodes an OBB using only 9 parameters.

4. Benchmarking OOD under JDet

4.1. Brief Description of JDet

Our benchmark utilizes the **J**ittor object **DET**ection models library (JDet), an open-source library dedicated to object detection, particularly supporting OOD methods. Built on Jittor [10], a deep learning framework, JDet facilitates the entire training and evaluation processes of object detection models. Preprocessing of diverse datasets precedes training or testing, ensuring a unified format. Various data augmentations, such as rotation and category balancing, are implemented as interchangeable and combinable modules. During testing, JDet supports diverse post-processing techniques for different datasets, with VOC2012 [4] serving as the implementation for evaluation. The library accommodates common object detection frameworks (e.g., Faster R-CNN [25]) and operators for OOD (e.g., RRoI Align [2]).

In total, JDet comprises 20 models and supports 6 datasets. A comparison between JDet and other open-source libraries is presented in Tab. 2.

4.2. Components for Unified Benchmarking

To mitigate variations between models, we categorized several modules and constructed OOD models by assembling these modules. The identified modules include:

- **Backbone:** Extracts features from input images; most models employ ResNet [9] as the backbone network with FPN [14] for feature extraction at different scales.
- **Anchor Generation:** Defines anchors for every pixel in the feature map.
- **Ground Truth Assignment:** Assigns ground truth bounding boxes to proposal regions based on their IoU.
- **Result Generation Network:** Neural networks for classifying anchors or proposal regions and regressing targets from regions.

Table 3. **mAP of models in JDet benchmark on DOTA-v1.0.**

Model	Venue	mAP_{50}	mAP_{75}	$mAP_{50:95}$
H2RBox [43]	ICLR'23	67.62	35.48	36.67
CSL [37]	ECCV'20	67.99	34.51	36.43
RSDet [23]	AAAI'21	68.41	36.93	37.91
RetinaNet [15]	ICCV'17	68.18	36.84	38.15
KLD [41]	NeurIPS'21	68.75	38.68	39.29
KFIoU [44]	ICLR'23	68.99	35.00	37.59
GWD [40]	ICML'21	69.02	38.48	39.62
FCOS [27]	ICCV'19	70.37	39.78	40.25
ATSS [50]	CVPR'20	72.44	39.81	41.08
S^2A-Net [8]	TGRS'21	73.95	37.14	39.89
Faster R-CNN [25]	NeurIPS'15	73.01	40.13	41.33
Gliding Vertex [35]	TPAMI'20	73.31	41.62	41.57
RoI Trans. [2]	CVPR'19	75.59	48.54	46.35
Oriented R-CNN [33]	ICCV'21	75.11	47.48	45.20
ReDet [7]	CVPR'21	76.38	50.83	47.08
Ours (RoI Trans. based)	-	**76.53**	50.41	46.97
Ours (ReDet based)	-	76.52	**51.38**	**47.67**

- **Encoder/Decoder:** Converts proposal regions into regression targets and model outputs into detection results.
- **Region of Interest Feature Extraction:** Extracts features of proposal regions for further detection and refinement, focusing on object-level details compared to image-level backbone features.
- **Loss Function:** Most models employ cross-entropy loss for classification and L1 loss for OBB prediction.

Implementing these modules consistently enhances the uniformity and comparability of the benchmarked models.

4.3. Detection Models in the Benchmark

Experiments are conducted on multiple models within our benchmark framework, all subject to uniform conditions. The baseline models chosen for comparison were Rotated Faster R-CNN [25] and Rotated RetinaNet [15]. To minimize implementation discrepancies, most of the other models are implemented with minimal alterations to their corresponding baseline architectures.

For fairness, we standardized data processing and training settings across different experiments, following the detailed settings outlined in Sec. 5.1. The experimental results on DOTA-v1.0 are presented in Tab. 3, with additional results available in the supplemental material.

5. Experiments

5.1. Datasets and Implementation Details

DOTA [32] is a dataset for remote sensing object detection. We evaluated models on DOTA-v1.0 and DOTA-v1.5. DOTA-v1.0 comprises 2,806 aerial images whose resolu-

(a) rotated Faster R-CNN + KLD (b) rotated Faster R-CNN + Ours

Figure 3. **Visual results of KLD [41] and ours.** Due to DA, KLD struggles to accurately predict the orientation of square-like objects. In contrast, our COBB circumvents DA, enhancing its precision in predicting the orientation of square-like objects.

Table 4. **Comparison: IoU Scores vs. One-hot Coding.** The experimental setup remains consistent with Rotated Faster R-CNN + COBB-sig, except for the OBB classification scores. All experiments were performed on DOTA-v1.0.

Scores	**mAP$_{50}$**	**mAP$_{75}$**	**mAP$_{50:95}$**
One-hot	73.46	43.76	42.90
IoU scores	74.00	44.03	43.29

Table 5. **Comparison of COBB across Proposal Types.** Experiments were performed on DOTA-v1.0 using RoI Transformer, a model that incorporates both Horizontal Proposals (HPs) and Oriented Proposals (OPs), as the baseline.

HPs	**OPs**	**mAP$_{50}$**	**mAP$_{75}$**	**mAP$_{50:95}$**
-	-	75.59	48.54	46.35
COBB-ln	-	76.27	50.23	47.06
-	COBB-ln	76.10	48.11	46.32
COBB-ln	COBB-ln	76.53	50.41	46.97

Table 6. **Comparing Different Regression Targets.** The experimental setup mirrors that of Rotated Faster R-CNN + COBB-ln. One approach utilizes r_a, representing the OBB's acreage ratio concerning its outer HBB, for Regression Target (RT) calculation, while the alternative method uses r_s, the sliding ratio. All experiments were conducted on DOTA-v1.0.

RT	**mAP$_{50}$**	**mAP$_{75}$**	**mAP$_{50:95}$**
Using r_a	74.13	43.31	42.94
Using r_s	74.44	44.08	43.53

tion is between 800× 800 and 4,000× 4,000, and a total of 188,282 target instances are annotated, covering 15 common categories. DOTA-v1.5 maintains the same image and dataset segmentation as DOTA-v1.0 but introduces labeling for extremely small objects (less than 10 pixels) and incorporates the container crane (CC) category.

DIOR [3] serves as a large-scale resource for remote sensing object detection, encompassing a total of 23,463 images, and spanning 20 distinct target categories. As stipulated in [3], DIOR is partitioned into a training set of 11,725 images and a testing set of 11,738 images.

HRSC2016 [18] is a ship detection dataset. Our training incorporates both the training and validation sets, while the test set is reserved for assessing model accuracy.

FAIR1M [26], designed for fine object detection in aerial images, consists of 5 categories with 37 subcategories. The dataset is available in two versions, -1.0 and -2.0. For FAIR1M-1.0, model training utilized the training set, and model evaluation was performed on the test set. For FAIR1M-2.0, models were trained on both the training and validation sets, with evaluation conducted on the test set.

All experiments were performed using a single NVIDIA RTX 3090. The models utilized ResNet-50 [9] and FPN [14] to extract multi-level feature maps. SGD optimization was employed during the training stage. Data augmentation included random flipping, with each image having a 50% chance of horizontal flipping followed by a 50% chance of vertical flipping.

5.2. Ablation Study

Comparison between IoU Scores and One-hot Coding: In Sec. 3.2, we implemented IoU scores to differentiate OBBs sharing the same (x_c, y_c, w, h, r_s). Alternatively, one-hot coding seems simpler for classification. We compared the two methods on rotated Faster R-CNN + COBB-sig, as recorded in Tab. 4. The model's accuracy using one-hot coding is lower than that using IoU scores due to the discontinuity introduced by one-hot coding.

Comparison of COBB across Proposal Types: The implementation of COBB on horizontal and oriented proposals is discussed in Sec. 3.3. To validate its effectiveness on both types, we conducted experiments on RoI Transformer [2], which employs both horizontal and oriented proposals. Results in Tab. 5 demonstrate integrating COBB enhances mAP$_{50}$ for both proposal types, with a more significant improvement observed in horizontal proposals.

Comparison between r_a and r_s: In Sec. 3.2, we approximated r_a with r_s. Further emphasizing the superiority of r_s over r_a, experiments were conducted on Faster R-CNN, as shown in Tab. 6. The results illustrate that r_s is more effective than r_a. This effectiveness stems from the complexity of recovering an OBB from the outer HBB and r_a, potentially leading to precision loss. Moreover, slight prediction errors on r_a may cause significantly larger errors in the predicted OBB than errors caused by slight r_s errors. Detailed insights are available in the supplemental material.

Table 7. **mAP across datasets.** COBB-sig takes r_{sig} for horizontal proposals, and COBB-ln takes r_{ln} for horizontal ones. COBB-ln-sig takes r_{ln} for horizontal ones, and r_{sig} for rotated ones. The definition of COBB-ln-ln and COBB-sig-sig is similar.

Models	DOTA-v1.0			DOTA-v1.5			DIOR		HRSC2016		FAIR1M-1.0	FAIR1M-2.0
	mAP_{50}	mAP_{75}	$mAP_{50:95}$	mAP_{50}	mAP_{75}	$mAP_{50:95}$	mAP_{50}	mAP_{75}	mAP_{50}	mAP_{75}	mAP_{50}	mAP_{50}
Rotated Faster R-CNN [25]	73.01	40.13	41.33	63.52	35.36	35.96	60.64	35.26	83.34	31.64	35.16	40.16
Gliding Vertex [35]	73.31	41.62	41.57	63.12	36.98	36.32	61.49	36.24	92.23	58.52	36.37	40.82
+COBB-sig	74.00	44.03	43.29	64.03	36.88	37.17	62.28	**37.70**	92.69	68.87	**36.81**	41.11
+COBB-ln	**74.44**	**44.08**	**43.53**	**64.35**	**37.62**	**37.30**	**62.58**	37.55	**92.71**	**72.29**	36.53	**41.23**
RoI Trans. [2]	75.59	48.54	46.35	65.69	41.76	40.36	66.09	44.26	96.73	88.76	39.31	43.93
+COBB-sig-sig	76.49	50.26	46.63	65.88	42.76	40.85	66.72	45.01	96.72	90.60	39.61	44.42
+COBB-ln-sig	**76.55**	49.91	46.68	**67.18**	41.75	40.80	67.47	**45.51**	96.71	90.89	**39.82**	**44.78**
+COBB-ln-ln	76.53	**50.41**	**46.97**	66.66	**43.29**	**40.96**	**67.53**	45.27	**97.19**	**91.35**	39.66	44.54
Oriented R-CNN [33]	75.11	47.48	45.20	65.47	40.35	39.31	64.38	41.19	96.61	86.49	38.30	42.90
+COBB-sig	75.52	48.35	45.61	**66.25**	41.34	**40.04**	**65.65**	**42.78**	**96.77**	87.43	38.81	43.31
+COBB-ln	**76.25**	**48.48**	**45.92**	66.18	**41.42**	40.01	65.42	42.19	96.74	**88.23**	**38.83**	**43.43**

5.3. Results and Analysis

Detailed results on different datasets and detectors are presented in Tab. 7, with comprehensive ablation study details available in the supplemental material.

Results on DOTA: The results on DOTA-v1.0 show that Gliding Vertex outperforms Faster R-CNN by 0.30% in mAP_{50} and 1.49% in mAP_{75}. Despite its accuracy advantage, Gliding Vertex suffers from discontinuity and DI, limiting its overall accuracy.

Our methods demonstrate superiority over existing approaches, especially in high-precision detection. Specifically, COBB outperforms Faster R-CNN by 1.21%, 3.92%, and 2.08% in mAP_{50}, mAP_{75}, and $mAP_{50:95}$ on average. On RoI Transformer and Oriented R-CNN, our method also significantly enhances the accuracy. For RoI Transformer, it improves mAP_{50}, mAP_{75}, and $mAP_{50:95}$ by 0.93%, 1.69%, and 0.41%, respectively. On Oriented R-CNN, it outperforms by 0.77%, 0.94%, and 0.56%, respectively. Notably, our method exhibits clear advantages in mAP_{75}, signifying its capability in high-precision object detection, a result of continuous representation and avoidance of DI and DA.

On DOTA-v1.5, our method effectively boosts the baseline detector accuracy. On average, the performance gain is 1.89%, 0.84%, and 1.03% in mAP_{75} for Faster R-CNN, RoI Transformer, and Oriented R-CNN, respectively. This highlights its effectiveness in small object detection.

Results on HRSC2016: HRSC2016 involves ship objects that are in large aspect ratios. As shown in Tab. 7, a substantial gap is observed between COBB-sig and COBB-ln. As discussed in Sec. 3.3, COBB-ln's advantage lies in its better capture of slight changes in r_s for inclined large aspect ratio objects. Accordingly, COBB-ln outperforms COBB-sig by 3.42% in mAP_{75} and Gliding Vertex by 13.77%. On RoI Transformer, COBB-ln-ln outperforms RoI Transformer by 2.59%.

Results on DIOR and FAIR1M: Experiments on less common datasets namely DIOR and FAIR1M, are also conducted. On DIOR, COBB-ln outperforms Faster R-CNN and RoI Transformer by 1.94% and 1.44%, respectively, in mAP_{50}, and by 2.29% and 1.01% in mAP_{75}. On FAIR1M-1.0 and FAIR1M-2.0, our method significantly improves baseline detectors.

Visualization Results: Fig. 3 visually compares the results of KLD [41] and COBB. KLD's precision for the orientation of square-like objects is compromised by DA, whereas COBB accurately represents these objects, achieving strong performance by eliminating DA.

6. Conclusion

We have extensively shown the presence of boundary discontinuity in existing OOD models. To solve this problem, we have introduced COBB, an innovative continuous OBB representation method. Our experimental results showcase the effectiveness of our proposed method, achieving a notable improvement of 3.95% in mAP_{75} on Rotated Faster R-CNN applied to the DOTA Dataset, without employing any additional techniques. COBB also has limitations. The outer HBB, sliding ratio r_s, and IoU scores exhibit irregular variations during OBB rotation, restricting its impact on rotation-equivariant detectors (*e.g.* ReDet [7]). Despite this, COBB proves effective in enhancing most OOD models by eliminating discontinuity.

7. Ackownledgement

The work was supported by the National Science and Technology Major Project under Grant 2021ZD0112902, the National Natural Science Foundation of China under Grant (62220106003, 62222607, 72342023), and the Research Grant of Beijing Higher Institution Engineering Research Center and Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology.

References

[1] Zhiming Chen, Kean Chen, Weiyao Lin, John See, Hui Yu, Yan Ke, and Cong Yang. Piou loss: Towards accurate oriented object detection in complex environments. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, pages 195–211. Springer, 2020. 3

[2] Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu. Learning roi transformer for oriented object detection in aerial images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2849–2858. Computer Vision Foundation / IEEE, 2019. 2, 6, 7, 8

[3] Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Michael Ying Yang, Serge J. Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(11): 7778–7796, 2022. 1, 7

[4] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010. 6

[5] Keren Fu, Yao Jiang, Ge-Peng Ji, Tao Zhou, Qijun Zhao, and Deng-Ping Fan. Light field salient object detection: A review and benchmark. *Computational Visual Media*, 8(4): 509–534, 2022. 2

[6] Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R. Martin, Ming-Ming Cheng, and Shi-Min Hu. Attention mechanisms in computer vision: A survey. *Computational Visual Media*, 8(3):331–368, 2022. 2

[7] Jiaming Han, Jian Ding, Nan Xue, and Gui-Song Xia. Redet: A rotation-equivariant detector for aerial object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 2786–2795. Computer Vision Foundation / IEEE, 2021. 2, 6, 8

[8] Jiaming Han, Jian Ding, Jie Li, and Gui-Song Xia. Align deep features for oriented object detection. *IEEE Trans. Geosci. Remote. Sens.*, 60:1–11, 2022. 6

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. 6, 7, 5

[10] Shi-Min Hu, Dun Liang, Guo-Ye Yang, Guo-Wei Yang, and Wen-Yang Zhou. Jittor: a novel deep learning framework with meta-operators and unified graph execution. *Sci. China Inf. Sci.*, 63(12), 2020. 2, 6, 5

[11] Yuqing Lan, Yao Duan, Chenyi Liu, Chenyang Zhu, Yueshan Xiong, Hui Huang, and Kai Xu. ARM3D: attention-based relation module for indoor 3d object detection. *Computational Visual Media*, 8(3):395–414, 2022. 2

[12] Yuxuan Li, Qibin Hou, Zhaohui Zheng, Ming-Ming Cheng, Jian Yang, and Xiang Li. Large selective kernel network for remote sensing object detection. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 16748–16759. IEEE, 2023. 6

[13] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-Song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 5909–5918. Computer Vision Foundation / IEEE Computer Society, 2018. 1

[14] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 936–944. IEEE Computer Society, 2017. 6, 7, 5

[15] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(2):318–327, 2020. 6

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, pages 21–37. Springer, 2016. 2

[17] Zikun Liu, Jingao Hu, Lubin Weng, and Yiping Yang. Rotated region based CNN for ship detection. In *2017 IEEE International Conference on Image Processing, ICIP 2017, Beijing, China, September 17-20, 2017*, pages 900–904. IEEE, 2017. 2

[18] Zikun Liu, Liu Yuan, Lubin Weng, and Yiping Yang. A high resolution optical satellite image dataset for ship recognition and some new baselines. In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2017, Porto, Portugal, February 24-26, 2017*, pages 324–331. SciTePress, 2017. 7

[19] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Trans. Multim.*, 20(11):3111–3122, 2018. 1

[20] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Trans. Multim.*, 20(11):3111–3122, 2018. 2

[21] Qi Ming, Lingjuan Miao, Zhiqiang Zhou, Xue Yang, and Yunpeng Dong. Optimization for arbitrary-oriented object detection via representation invariance loss. *IEEE Geosci. Remote. Sens. Lett.*, 19:1–5, 2022. 2, 3

[22] Guangtao Nie and Hua Huang. Multi-oriented object detection in aerial images with double horizontal rectangles. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(4):4932–4944, 2023. 3, 1

[23] Wen Qian, Xue Yang, Silong Peng, Junchi Yan, and Yue Guo. Learning modulated loss for rotated object detection. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 2458–2466. AAAI Press, 2021. 2, 3, 6

[24] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788. IEEE Computer Society, 2016. 2

[25] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015. 2, 4, 5, 6, 8

[26] Xian Sun, Peijin Wang, Zhiyuan Yan, Feng Xu, Ruiping Wang, Wenhui Diao, Jin Chen, Jihao Li, Yingchao Feng, Tao Xu, et al. 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, 2022. 7

[27] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: fully convolutional one-stage object detection. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 9626–9635. IEEE, 2019. 6

[28] Hao Wang, Zhanchao Huang, Zhengchao Chen, Ying Song, and Wei Li. Multigrained angle representation for remote-sensing object detection. *IEEE Trans. Geosci. Remote. Sens.*, 60:1–13, 2022. 3, 6

[29] Jian Wang, Fan Li, and Haixia Bi. Gaussian focal loss: Learning distribution polarized angle prediction for rotated object detection in aerial images. *IEEE Trans. Geosci. Remote. Sens.*, 60:1–13, 2022. 3, 6, 1

[30] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. PVT v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 2

[31] Haoran Wei, Yue Zhang, Zhonghan Chang, Hao Li, Hongqi Wang, and Xian Sun. Oriented objects as pairs of middle lines. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169:268–279, 2020. 3

[32] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge J. Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. DOTA: A large-scale dataset for object detection in aerial images. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3974–3983. Computer Vision Foundation / IEEE Computer Society, 2018. 1, 6

[33] Xingxing Xie, Gong Cheng, Jiabao Wang, Xiwen Yao, and Junwei Han. Oriented R-CNN for object detection. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 3500–3509. IEEE, 2021. 2, 6, 8, 5

[34] Hang Xu, Xinyuan Liu, Haonan Xu, Yike Ma, Zunjie Zhu, Chenggang Yan, and Feng Dai. Rethinking boundary discontinuity problem for oriented object detection. *CoRR*, abs/2305.10061, 2023. 3

[35] Yongchao Xu, Mingtao Fu, Qimeng Wang, Yukang Wang, Kai Chen, Gui-Song Xia, and Xiang Bai. Gliding vertex on the horizontal bounding box for multi-oriented object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(4):1452–1459, 2021. 2, 3, 5, 6, 8, 1

[36] Guo-Ye Yang, Xiang-Li Li, Zi-Kai Xiao, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. Sampling equivariant self-attention networks for object detection in aerial images. *IEEE Trans. Image Process.*, 32:6413–6425, 2023. 6

[37] Xue Yang and Junchi Yan. Arbitrary-oriented object detection with circular smooth label. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VIII*, pages 677–694. Springer, 2020. 1, 2, 3, 4, 6

[38] Xue Yang, Jirui Yang, Junchi Yan, Yue Zhang, Tengfei Zhang, Zhi Guo, Xian Sun, and Kun Fu. Scrdet: Towards more robust detection for small, cluttered and rotated objects. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 8231–8240. IEEE, 2019. 3

[39] Xue Yang, Liping Hou, Yue Zhou, Wentao Wang, and Junchi Yan. Dense label encoding for boundary discontinuity free rotation detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 15819–15829. Computer Vision Foundation / IEEE, 2021. 2, 3, 4, 6, 1

[40] Xue Yang, Junchi Yan, Qi Ming, Wentao Wang, Xiaopeng Zhang, and Qi Tian. Rethinking rotated object detection with gaussian wasserstein distance loss. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, pages 11830–11841. PMLR, 2021. 1, 2, 3, 6

[41] Xue Yang, Xiaojiang Yang, Jirui Yang, Qi Ming, Wentao Wang, Qi Tian, and Junchi Yan. Learning high-precision bounding box for rotated object detection via kullback-leibler divergence. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 18381–18394, 2021. 3, 6, 7, 8, 1

[42] Xue Yang, Yue Zhou, and Junchi Yan. Alpharotate: A rotation detection benchmark using tensorflow. *CoRR*, abs/2111.06677, 2021. 6

[43] Xue Yang, Gefan Zhang, Wentong Li, Yue Zhou, Xuehui Wang, and Junchi Yan. H2rbox: Horizontal box annotation is all you need for oriented object detection. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. 6

[44] Xue Yang, Yue Zhou, Gefan Zhang, Jirui Yang, Wentao Wang, Junchi Yan, Xiaopeng Zhang, and Qi Tian. The kfiou loss for rotated object detection. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. 3, 6, 1

[45] Jingru Yi, Pengxiang Wu, Bo Liu, Qiaoying Huang, Hui Qu, and Dimitris N. Metaxas. Oriented object detection in aerial images with box boundary-aware vectors. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*, pages 2149–2158. IEEE, 2021. 3

[46] Yi Yu and Feipeng Da. Phase-shifting coder: Predicting accurate orientation in oriented object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 13354–13363. IEEE, 2023. 2, 3, 1

[47] Yuan Yuan, Zhiguo Li, and Dandan Ma. Feature-aligned single-stage rotation object detection with continuous boundary. *IEEE Trans. Geosci. Remote. Sens.*, 60:1–11, 2022. 3

[48] Ying Zeng, Xue Yang, Qingyun Li, Yushi Chen, and Junchi Yan. ARS-DETR: aspect ratio sensitive oriented object detection with transformer. *CoRR*, abs/2303.04989, 2023. 3, 6

[49] Rufei Zhang, Yuqing Wang, Sheng Shen, Wei Zhao, Zhiliang Zeng, Nannan Li, and Dongjin Li. Trigonometric-coded refined detector for high precision oriented object detection. *IEEE Geosci. Remote. Sens. Lett.*, 20:1–5, 2023. 3

[50] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9756–9765. Computer Vision Foundation / IEEE, 2020. 6

[51] Pengbo Zhao, Zhenshen Qu, Yingjia Bu, Wenming Tan, and Qiuyu Guan. Polardet: A fast, more precise detector for rotated target in aerial images. *International Journal of Remote Sensing*, 42(15):5831–5861, 2021. 3

[52] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE Trans. Neural Networks Learn. Syst.*, 30(11):3212–3232, 2019. 1

[53] Tao Zhou, Deng-Ping Fan, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. RGB-D salient object detection: A survey. *Computational Visual Media*, 7(1):37–69, 2021. 2

[54] Yue Zhou, Xue Yang, Gefan Zhang, Jiabao Wang, Yanyi Liu, Liping Hou, Xue Jiang, Xingzhao Liu, Junchi Yan, Chengqi Lyu, Wenwei Zhang, and Kai Chen. Mmrotate: A rotated object detection benchmark using pytorch. In *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*, pages 7331–7334. ACM, 2022. 6