On the Arbitrary-Oriented Object Detection: Classification based Approaches Revisited

Xue Yang, Junchi Yan *Senior Member, IEEE*

Abstract—Arbitrary-oriented object detection has been a building block for rotation sensitive tasks. We first show that the boundary problem suffered in existing dominant regression-based rotation detectors, is caused by angular periodicity or corner ordering, according to the parameterization protocol. We also show that the root cause is that the ideal predictions can be out of the defined range. Accordingly, we transform the angular prediction task from a regression problem to a classification one. For the resulting circularly distributed angle classification problem, we first devise a Circular Smooth Label technique to handle the periodicity of angle and increase the error tolerance to adjacent angles. To reduce the excessive model parameters by Circular Smooth Label, we further design a Densely Coded Labels, which greatly reduces the length of the encoding. Finally, we further develop an object heading detection module, which can be useful when the exact heading orientation information is needed e.g. for ship and plane heading detection. We release our OHD-SJTU dataset and OHDet detector for heading detection. Extensive experimental results on three large-scale public datasets for aerial images i.e. DOTA, HRSC2016, OHD-SJTU, and face dataset FDDB, as well as scene text dataset ICDAR2015 and MLT, show the effectiveness of our approach.

Index Terms—Arbitrary-Oriented Object Detection, Boundary Problem, Circular Smooth Label, Densely Coded Labels, Object Heading Detection

1 Introduction

Object detection has been a standing task in computer vision. Recently, rotation detection has played an emerging and vital role in processing and understanding visual information from aerial images [1], [2], [3], [4], [5], scene text [6], [7], [8], [9], [10], [11] and face [12], [13], [14]. The rotation detector can provide accurate orientation and scale information, which will be helpful in applications such as object change detection in aerial images and recognition of sequential characters for multi-oriented scene texts.

Recently, a line of advanced rotation detectors evolved from classic detection algorithms [15], [16], [17], [18], [19] have been developed. Among these methods, detectors based on region regression take the dominance, and the representation of multi-oriented object is achieved by rotated bounding box or quadrangles. Although these rotation detectors have achieved promising results, there are still some fundamental problems. Specifically, we note both the five-parameter regression [2], [3], [9] and the eight-parameter regression [11], [20], [21], [22] methods suffer the issue of discontinuous boundaries, as often caused by angular periodicity or corner ordering, depending on the choice of parameterization protocol. However, the root reasons are not limited to the particular representation of the bounding box. In this paper, we argue that the root cause of boundary problems based on regression methods is that the ideal predictions are beyond the defined range. Thus, the model's loss value suddenly increases at the boundary situation so that the model cannot obtain the prediction result in the simplest and most direct way, and additional complicated treatment is often needed. Therefore, these detectors often have difficulty in boundary conditions. For detection using rotated bounding boxes, the accuracy of angle prediction is critical. A slight angle deviation can lead to notable Intersection-over-Union (IoU) drop, resulting in inaccurate detection, especially for large aspect ratios.

There have been some works addressing the boundary problem. For example, IoU-smooth L1 loss [3] introduces the IoU factor, and modular rotation loss [21] increases the boundary constraint to eliminate the sudden increase in boundary loss and reduce the difficulty of model learning. However, these regression-based detection methods still have not solved the root cause as mentioned above.

In this paper, we aim to devise a more fundamental rotation detection baseline to solve the boundary problem. Specifically, we consider object angle prediction as a classification problem to better limit the prediction results, and then we design a Circular Smooth Label (CSL) to address angle periodicity and to increase the tolerance between adjacent angles. We show that the rotation accuracy error due to the conversion from continuous prediction to discrete bins can be negligible by a fine-granularity. We also introduce four window functions in CSL and explore the effect of different window radius sizes on detection performance. We further design two Densely Coded Labels (DCL), which greatly reduce the length of the encoding while ensuring the angle prediction accuracy does not sacrifice. In order to eliminate the theoretical prediction errors caused by angle dispersion, We also propose an angle fine-tuning mechanism. Finally, we implement object heading detection on the basis of rotation detection, namely OHDet, and release a new dataset OHD-SJTU to the community. Through experiments and visual analysis, we show that CSL-based and

Xue Yang, Junchi Yan (correspondence author) are with Department of Computer Science and Engineering, Shanghai Jiao Tong University, and also with the MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University.
E-mail: {yangxue-2019-sjtu,yanjunchi}@sjtu.edu.cn

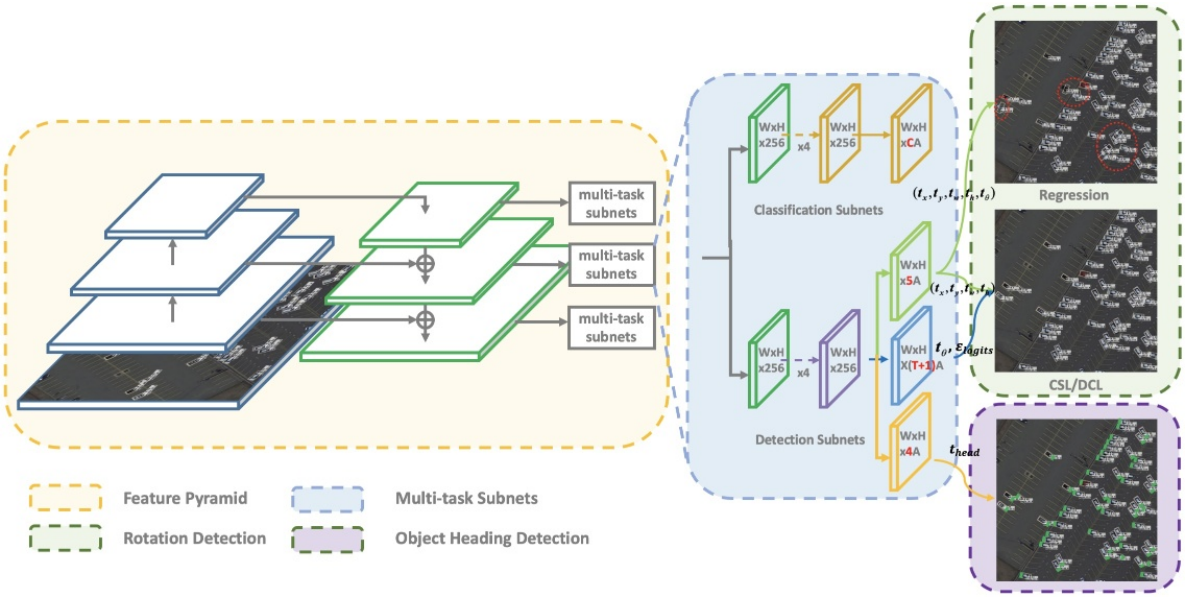


Fig. 1. Architecture of the proposed detector (RetinaNet [15] as an embodiment). 'W' and 'H' refer to the width and height respectively. 'C' and 'T' in red represents the number of object and encoding length of the angle, respectively. Object heading detection refers to further finding the head of the object based on the orientation information obtained by the rotation detection. Note that the CSL/DCL module on the right refers to the classification based prediction function either based on a Circular Smooth Label or the size reduced version Densely Coded Label.

DCL-based rotation detection algorithms are indeed better baseline choices than the angle regression-based protocol on different detectors and datasets. Note the regression-based, CSL-based and DCL-based protocols mentioned in the rest of the paper are named according to the prediction form of the angle. As a byproduct, we show how the object head can be effectively identified after detecting the rotating bounding box.

The preliminary content of this paper has partially appeared in ECCV 2020 [23] and CVPR 2021 [24][1]. The overall contributions of this extended journal version can be summarized as:

- We characterize the boundary problems encountered in different regression-based rotation detection methods [1], [2], [4], [5] and show the root cause is that the ideal predictions are beyond the defined range.
- To effectively dismiss the boundary problem, we design a novel classification based rotation detection paradigm, in contrast to the dominant regression based methods in existing works. The incurred accuracy error is negligible thanks to the devised fine-grained angle discretization (less than 1 degree) which to our best knowledge, has not been developed yet in the literature, and a coarse classification (around 10-degree) model is dated back to the 1990s for face detection [14].
- We develop the Circular Smooth Label (CSL) technique as an independent module. It can be readily reused in existing regression based methods by replacing the regression component with classification, to enhance angular prediction in face of boundary conditions and objects with large aspect ratio. We further design a Densely Coded Label (DCL) to solve the problem of excessive model parameters induced by CSL. DCL can greatly reduce the length of the encoding while maintaining a high angle prediction accuracy. We also propose an angle fine-tuning mechanism to eliminate the theoretical prediction errors caused by angle dispersion which has been a common issue in both CSL and DCL.
- On the basis of rotation detection, we further develop an object heading detector, namely OHDet, to identify the heading of object. In addition, we annotate and release a dataset called OHD-SJTU[2], which can be used for both rotation detection and object heading detection tasks.

1. To obtain a more thorough analysis and comprehensive results, the conference versions [23], [24] have been significantly extended and improved in this journal version, especially in the following aspects: i) We explore the relationship between the angle discrete representation granularity denoted by ω and the detection performance. It shows that discrete granularity ω can be approximated as a CSL technique with a rectangular window function, which has a certain tolerance in the divided angle interval. The difference is that CSL smooths between adjacent angle intervals. See Table 7 in Section 4.2; ii) We use a specific calculation example to explain why the code length has such a large impact on the amount of detection model parameters and calculations, see Section 3.5; iii) As for the angle prediction of the regression branch, we use two forms as the baseline to be compared, include direct regression and indirect regression, see Section 3.8; iv) We verify our approach on additional more challenging datasets, including FDDB, and DOTA-v1.5/v2.0, see Table 10 and Table 11. Among them DOTA-v1.5/v2.0 contain more data and tiny object (less than 10 pixels) than DOTA-v1.0; v) We propose an angle fine-tuning mechanism to eliminate the theoretical prediction errors caused by angle dispersion which has been a common issue in whatever CSL and DCL, see Section 3.6; vi) As a common function for downstream applications, we develop a classification-based object heading detector in Section 3.7. To verify its usefulness, we annotate and release a new dataset for this purpose and perform detection evaluation for both rotation and heading with a considerable amount, and more stringent evaluation indicators are used, as detailed in Section 4.1. To our best knowledge, this is the first public benchmark for multiple-category heading detection, especially at a considerable scale. Finally, we also release the full version of the source code.

2. https://yangxue0827.github.io/OHD-SJTU.html.

- Extensive experimental results on HRSC2016 and DOTA-v1.0 show the state-of-the-art performance of our detector, and the efficacy of the CSL and DCL technique as independent components have been verified across different detectors. The source code [25] is publicly available[3].

The paper is organized as follows. Section 2 introduces the related work. Section 3 presents the main approach in this paper and the experiments are conducted in Section 4. Section 5 concludes this paper.

2 Related Work

In this section, we discuss the related work, including classic region object detection and rotated object detection. Specifically, we discuss some relevant techniques regarding classification based orientation estimation, as well as the recent works on object heading detection.

2.1 Horizontal Region Object Detection

Classic object detection aims to detect general objects with horizontal bounding boxes, and many high performance general-purpose object detectors have been proposed. R-CNN [26] pioneers a method based on CNN detection. Subsequently, region-based models such as Fast R-CNN [16], Faster R-CNN [17], and R-FCN [19] are proposed, which improve the detection speed while reducing computational storage. FPN [18] focuses on the scale variance of objects in images and propose feature pyramid network to handle objects at different scales. SSD [27], YOLO [28] and RetinaNet [15] are representative single-stage methods, and their single-stage structure leads to higher detection speeds. Compared to anchor-based protocols, many anchor-free have become extremely popular in recent years. CornerNet [29], CenterNet [30] and ExtremeNet [31] attempt to predict some keypoints of objects such as corners or extreme points, which are then grouped into bounding boxes. However, horizontal detector does not provide accurate orientation and scale information, which poses problem in real applications such as object change detection in aerial images and recognition of sequential characters for multi-oriented scene texts.

2.2 Arbitrary-oriented Object Detection

Aerial images and scene text are the main application scenarios of the rotation detector. Recent advances in multi-oriented object detection are mainly driven by adaption of classical object detection methods using rotated bounding boxes or quadrangles to represent multi-oriented objects. Due to the complexity of the remote sensing image scene and the large number of small, cluttered and rotated objects, multi-stage rotation detectors are still dominant for their robustness. Among them, ICN [4], ROI-Transformer [2], SCRDet [3], R^3Det [1] are state-of-the-art detectors. Gliding Vertex [22] and RSDet [21] achieve more accurate object detection through quadrilateral regression prediction. For scene text detection, RRPN [9] employs rotated RPN to generate rotated proposals and further performs rotated bounding box regression. TextBoxes++ [11] adopts vertex regression on SSD. RRD [10] further improves TextBoxes++ by decoupling classification and bounding box regression on rotation-invariant and rotation sensitive features, respectively. In fact, these mainstream regression-based methods often suffer the boundary problems due to the predictions beyond the defined range.

The idea of segmentation is an effective way of solving boundary problem. For example, segmentation-based protocols [32], [33] are popular in the area of scene text detection. However, these methods are not practical for aerial images, which often contain a large number of densely arranged small objects in multiple categories. In contrast, instance segmentation is more suitable, such as Mask R-CNN [34], SOLO [35], and CondInst [36], but there are also many limitations.

First, Considering that instance segmentation requires a lot of labeling workload, a more straightforward solution is to convert the rotated boxes into binary masks [37]. However, such conversion will introduce many background areas, which will reduce the classification accuracy of pixels and affect the accuracy of the final prediction box. Besides, for the top-down methods (e.g. Mask RCNN), dense scenes will limit the detection of horizontal boxes because of the excessive suppression of dense horizontal overlapping bounding boxes due to non-maximum suppression (NMS), thereby affecting subsequent segmentation. Last but not least, the bottom-up methods, such as SOLO and CondInst, assign different instances to different channels, so they are not suitable for aerial images, which often show large scale scenes with a large number of dense and small objects. Take the parking lot scene in DOTA-v1.5 [38] dataset as an example, a sub-image with a size of 450×600 will contain up to 2,000 vehicles, which are often less than 10 pixels in size and are densely arranged.

The above reasons may help explain why angle-based rotation detection algorithms still dominate in aerial imagery which is an important application area. Therefore, we design a new rotation detection baseline, which basically eliminates the boundary problem by transforming angle prediction from a regression problem to a classification problem.

2.3 Classification for Orientation Information

Early works have been developed for multi-view face detection with arbitrary rotation-in-plane (RIP) angles, by obtaining orientation information via classification. Specifically, the divide-and-conquer technique is adopted in [13], which uses several small neural networks to deal with a small range of face appearance variations individually. In [14], a router network is firstly used to estimate each face candidate's RIP angle. PCN [12] progressively calibrates the RIP orientation of each face candidate and shrinks the RIP range by half in early stages. Finally, PCN makes the accurate final decision for each face candidate to determine whether it is a face and predict the precise RIP angle. In other research areas, [39] adopts ordinal regression for effective future motion classification. [40] obtains the orientation information of the ship by classifying the four sides. The above methods all obtain the approximate orientation range through classification, but cannot be directly applied to

3. https://github.com/yangxue0827/RotationDetection

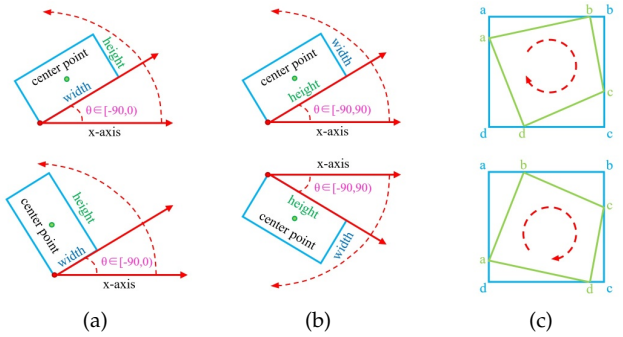


Fig. 2. Popular definitions of bounding boxes in existing literature. (a) Five-parameter method with 90° angular range [1], [3]. (b) Five-parameter method with 180° angular range [2], [9]. (c) Ordered quadrilateral representation [11], [20], [21], [22].

scenarios that require precise orientation information such as aerial images and scene text. They also do not suffer the boundary problem due to the PoA and EoE, because their prediction granularity is very rough and the aspect ratio of object is small.

2.4 Object Heading Detection

DLR 3K [41] is an aerial image dataset that can be used for car head detection. There are 20 images in total, including 3,418 cars and 54 trucks. For car head detection, the authors first use a sliding window strategy with binary classification to detect cars, and then perform a rough estimation of the head through classification, with 16 classes (22.5° rotation difference between adjacent sample groups, respectively). This method needs to train multiple detectors and cannot perform high-precision angle prediction. DRBox [42] and DRBox-v2 [43] define the rotation bounding box according to the head of the object, and the angle predicted by regression can be used to determine the direction of the object head. However, the bounding box of this definition protocol has a relatively large angular range, at $[0^\circ, 360^\circ)$, which is challenging. EAGLE [44] is a large-scale dataset for vehicle detection in aerial imagery, which has still not been released so far. This paper releases a new dataset called OHD-SJTU with labeled heading information, and it covers more categories of objects.

In summary, this work is dedicated to proposing a general detection method that can be used for multi-class high-precision rotation detection and object heading detection.

3 Proposed Approach

Figure 1 gives an overview of our method. The embodiment is a single-stage rotation detector based on the RetinaNet [15]. The figure shows a multi-tasking pipeline, including classification branch, rotation detection branch and object heading detection branch. Among them, rotation detection branch contains regression based prediction and CSL-based prediction, to facilitate the comparison of the performance of the two methods. It can be seen from the figure that CSL-based protocol is more accurate for learning the orientation and scale information of the object. The DCL-based protocol maintains consistent performance with CSL.

Note that the method proposed in this paper is applicable to most regression-based protocols by replacing the regression module with our classification one.

3.1 Regression-based Rotation Detection Method

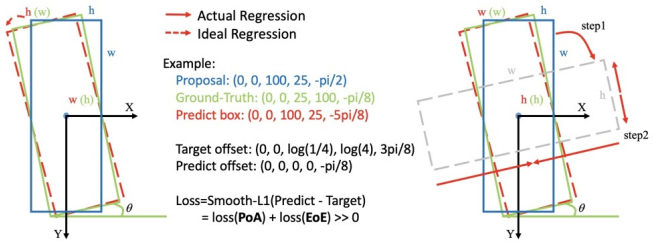
Parametric regression is currently a popular method for rotation object detection, mainly including five-parameter regression-based protocols [1], [2], [3], [5], [8], [9] and eight-parameter regression-based protocols [11], [20], [21], [22]. The commonly used five-parameter regression-based protocols realize arbitrary-oriented bounding box detection by adding an additional angle parameter θ. Figure 2(a) shows one of the rectangular definition (x, y, w, h, θ) with 90° angular range [1], [3], [5], [9], θ denotes the acute angle to the x-axis, and for the other side we refer it as w. It should be distinguished from another definition (x, y, h, w, θ) illustrated in Figure 2(b), with 180° angular range [2], [9], whose θ is determined by the long side (h) of the rectangle and x-axis. The eight-parameter regression-based detectors directly regress the four corners $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$ of the object, so the prediction is a quadrilateral. The key step to the quadrilateral regression is to sort the four corner points in advance, which avoids large loss even if the prediction is correct, as shown in Figure 2(c).

3.2 Boundary Problem of Regression Method

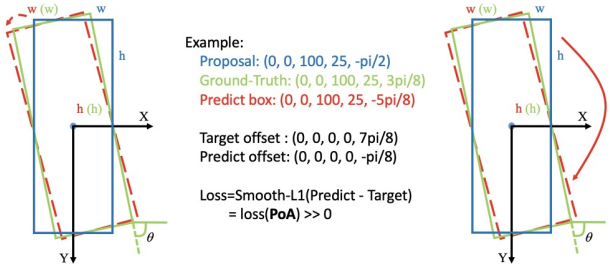
Although the parametric regression-based rotation detection method has achieved competitive performance in visual detection, these methods essentially suffer the discontinuous boundaries problem [3], [21]. On the surface, boundary discontinuity problems are often caused by angular periodicity under the five-parameter protocol, or the corner ordering in the eight-parameter setting. While there exists a more fundamental root cause behind the representation choice of the bounding box.

The boundary discontinuity can cause the loss value suddenly increase at the boundary situation. Thus methods have to resort to particular and often complex tricks to mitigate this issue. Therefore, these detection methods are often inaccurate in boundary conditions. We describe the boundary problem in three typical categories of regression-based protocols according to their different representation forms (the first two refer to the five-parameter methods):

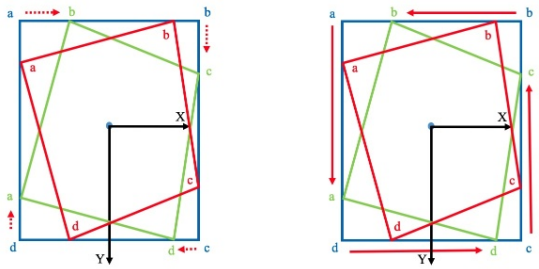
- **The 90°-regression-based protocol, as sketched in Figure 3(a).** It shows that an ideal form of regression (the blue box rotates counterclockwise to the red box), but the loss of this situation is very large due to the periodicity of angular (PoA) and exchangeability of edges (EoE), see the example in Figure 3(a) and Equation 13, 12, 17 for detail. Therefore, the model has to be regressed in other complex forms (such as the blue box rotating clockwise to the gray box while scaling w and h), increasing the difficulty of regression. It should be noted that the prediction box and ground truth in the ideal regression way do have a high IoU value in visual perception, but the prediction box at this time has exceeded our defined range so that we cannot calculate the accurate IoU if no additional judgment processing is performed.



(a) The 90°-regression-based protocol



(b) The 180°-regression-based protocol



(c) Point-regression-based protocol

Fig. 3. Illustration for the boundary problem which persistently exists over three popular categories of regression based protocols. The red solid arrow indicates the actual regression process, and the red dotted shows the ideal regression process.

- **The 180°-regression-based protocol, as illustrated in Figure 3(b).** Similarly, this method also suffers the issue of sharp increase of loss caused by the PoA at the boundary. The model will eventually choose to rotate the proposal a large angle clockwise to get the final predicted bounding box.
- **Point-regression-based protocol, as shown in Figure 3(c).** Through further analysis, the boundary discontinuity problem still exists in the eight-parameter regression method due to the advance ordering of corner points. Consider the situation of an eight-parameter regression in the boundary case, the ideal regression process should be $\{(a \rightarrow b), (b \rightarrow c), (c \rightarrow d), (d \rightarrow a)\}$, but the actual regression process from the blue reference box to the green ground truth box is $\{(a \rightarrow a), (b \rightarrow b), (c \rightarrow c), (d \rightarrow d)\}$. In fact, this situation also belongs to PoA. By contrast, the actual and ideal regression of the blue to red bounding boxes is consistent.

Some approaches have been proposed to solve these problems based on the above analysis. For example, IoU-smooth L1 [3] loss introduces the IoU factor, and modular rotation loss [21] increases the boundary constraint to eliminate the sudden increase in boundary loss and reduce the difficulty of model learning. However, these methods are still regression-based detection methods, and no solution is given from the root cause.

In this paper, we will start from a new perspective and replace regression with classification to achieve better and more robust rotation detectors. We reproduce some classic rotation detectors based on regression and compare them visually under boundary conditions, as shown in Figure 4(a) to Figure 4(d). In contrast, CSL-based and DCL-based protocols have no boundary problem, as shown in Figure 4(i) and 4(e).

3.3 Vanilla Angular Classification

The main cause of boundary problems based on regression methods is that the ideal predictions are beyond the defined range. Therefore, we consider the prediction of object angle as a classification task to restrict the prediction range. One simple solution is to use the object angle as its category label, and the number of categories is related to the angle range. Figure 5(a) shows the label setting for a vanilla classification problem (one-hot label encoding). The conversion from regression to classification can cause certain accuracy error. Taking the five-parameter method with 180° angle range as an example: ω (default $\omega = 1°$) degree per interval refers to a category for labeling. It calculates the maximum accuracy error $Max(error)$ and the expected accuracy error $E(error)$:

$$\begin{aligned} Max(error) &= \frac{\omega}{2} \\ E(error) &= \int_a^b \frac{x}{b-a} dx = \int_0^{\frac{\omega}{2}} \frac{x}{\frac{\omega}{2}-0} dx = \frac{\omega}{4} \end{aligned} \tag{1}$$

where $\omega = AR/C_\theta$ indicates the angle discretization granularity. AR and C_θ represents angle range (the default value is 180) and the number of angle categories, respectively.

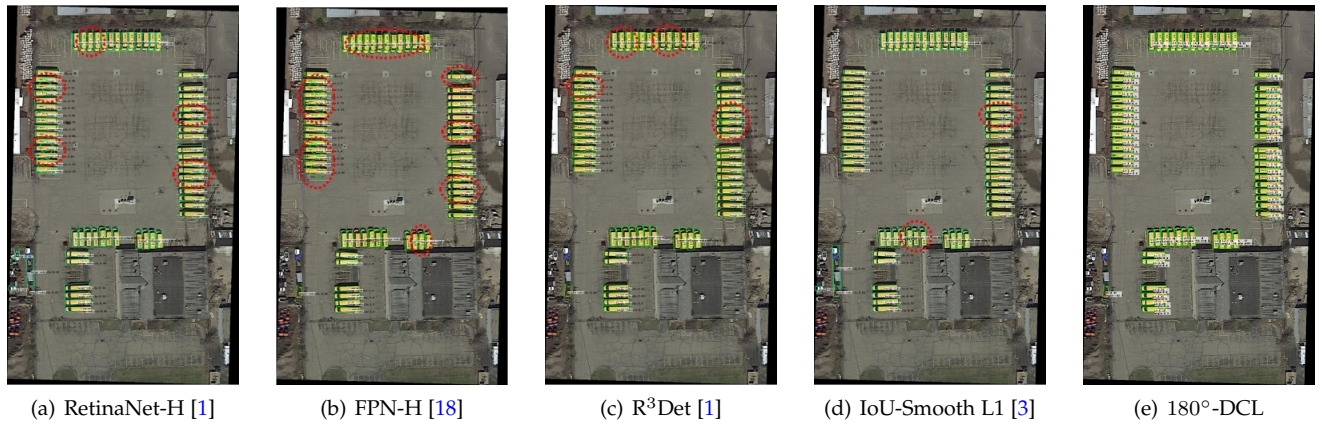
Based on the above equations, one can see the error is slight for a rotation detector with small enough angle discrete granularity ω. For example, when two rectangles with a $1:9$ aspect ratio differ by $0.25°$ and $0.5°$ (default expected and maximum accuracy error), the Intersection over Union (IoU) between them only decreases by 0.02 and 0.05.

The discrete equation and prediction equation of the angle are as follows:

$$\begin{aligned} &\textbf{Encode: } \text{One-Hot}(-\text{Round}((\theta_{gt} - 90)/\omega)) \\ &\textbf{Decode: } 90 - \omega\,(0.5 + \text{Argmax}(\text{Sigmoid}(logits))) \end{aligned} \tag{2}$$

where θ_{gt} presents the angle decimal label.

Applying vanilla classification methods to prediction of angles is appeared earlier in the field of face detection [12], [13], [14]. As these works only need approximate orientation range, e.g.10-degree ($\omega = 10$) in [14]. We find that vanilla angular classification methods are difficult to deal with objects with multiple categories and large aspect ratio, e.g. DOTA dataset. In contrast, the small aspect ratio and single category characteristics in face detection make it unnecessary for high-precision angle prediction, so high-precision angle classification still has not been solved.



(a) RetinaNet-H [1] (b) FPN-H [18] (c) R^3Det [1] (d) IoU-Smooth L1 [3] (e) 180°-DCL

(f) 180°-CSL-Pulse (g) 180°-CSL-Rect. (h) 180°-CSL-Triangle (i) 180°-CSL-Gaussian (j) 90°-CSL-Gaussian

Fig. 4. Comparison of four regression-based rotation detection methods and angle classification-based protocols in the boundary case. 'H' and 'R' represent the horizontal and rotating anchors. Red dotted circles indicate some bad cases. Figures 4(f)-4(i) show that the Gaussian window function performs best, while the pulse function performs worst because it has not learned any orientation and scale information. According to Figure 4(i) and Figure 4(j), the 180°-CSL-based protocol obviously has better boundary prediction due to the EoE problem still exists in the 90°-CSL-based protocol. In general, CSL-based and DCL-based protocols have no boundary problem, as shown in Figure 4(i) and 4(e).

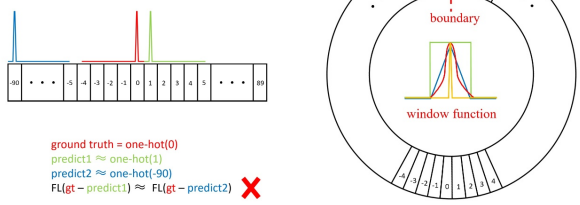
3.4 Circular Smooth Label for Angular Classification

In our analysis, there are two reasons why vanilla classification methods cannot obtain high-precision angle prediction for rotation detection:

Reason i) The EoE problem still exists when the bounding box uses the 90°-based protocol, as shown in Figure 4(j). Moreover, 90°-based protocol has two different border cases (vertical and horizontal), while 180°-based protocol has only vertical border cases.

Reason ii) Note vanilla classification loss is agnostic to the angle distance between the predicted label and ground truth label, thus it is inappropriate for the nature of the angle prediction problem. As shown in Figure 5(a), when the ground truth is 0° and the prediction results of the classifier are 1° and −90° respectively, their prediction losses are the same, but the prediction results close to ground truth should be allowed from a detection perspective.

Therefore, Circular Smooth Label (CSL) technique is designed to obtain more robust angular prediction through classification without suffering boundary conditions, including EoE and PoA. It should be noted that CSL can only solve the PoA, and the EoE problem can be solved by the 180° angular definition method. It can be clearly seen from Figure 5(b) that CSL involves a circular label encoding with periodicity, and the assigned label value is smooth with a



(a) One-hot label. (b) Circle smooth label.

Fig. 5. Two kinds of labels for angular classification. FL means using the focal loss function [15].

certain tolerance. The expression of CSL is as follows:

$$CSL(x) = \begin{cases} g(x), & \theta - r < x < \theta + r \\ 0, & otherwise \end{cases} \quad (3)$$

where $g(x)$ is a window function. r is the radius of the window function. θ represents the angle of the current bounding box. An ideal window function $g(x)$ is required to hold the following properties:

- **Periodicity**: $g(x) = g(x + kT), k \in N$. $T = 180/\omega$ represents the number of bins into which the angle is divided, and the default value is 180.

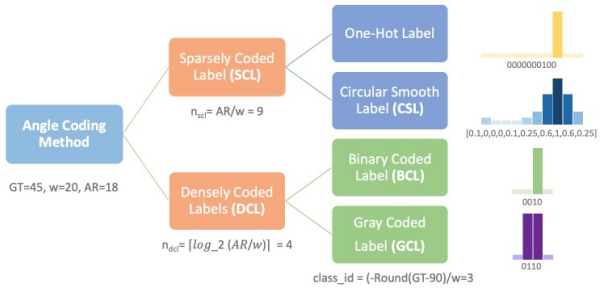


Fig. 6. The relationship between the various angle encoding methods.

- **Symmetry**: $0 \leq g(\theta+\varepsilon) = g(\theta-\varepsilon) \leq 1, |\varepsilon| < r$. θ is the center of symmetry.
- **Maximum**: $g(\theta) = 1$.
- **Monotonic**: $0 \leq g(\theta \pm \varepsilon) \leq g(\theta \pm \varsigma) \leq 1, |\varsigma| < |\varepsilon| < r$. The function presents a monotonous non-increasing trend from the center point to both sides

Figure 5(b) shows four efficient window functions that meet the above four properties: pulse functions, rectangular functions, triangle functions, and Gaussian functions. Note that the label value is continuous at the boundary and there is no arbitrary accuracy error due to the periodicity of CSL. In addition, one-hot label (vanilla classification) is equivalent to CSL when the window function is a pulse function or the radius of the window function is very small. Equation 4 describes the angle prediction process in CSL:

$$\begin{aligned} &\textbf{Encode: } \mathrm{CSL}(-\mathrm{Round}((\theta_{gt}-90)/\omega)) \\ &\textbf{Decode: } 90 - \omega(\mathrm{Argmax}(\mathrm{Sigmoid}(logits)) + 0.5) \end{aligned} \tag{4}$$

3.5 Densely Coded Label for Angular Classification

The CSL-based detectors adopt the so-called Sparsely Coded Label (SCL) encoding technique. Although CSL has many good properties for angle prediction, the The design of CSL will cause the prediction layer to introduce too much parameter and calculation, resulting in inefficiency of the detector. Specifically, the One-Hot and CSL described above are sparse angle encoding methods, which often leads to excessively long angle encoding length:

$$L_{one-hot} = L_{csl} = AR/\omega \tag{5}$$

Binary Coded Label (BCL) [45] and Gray Coded Label (GCL) [46] are two Densely Coded Label (DCL) methods commonly used in the field of electronic communication. Their advantage is that they can represent a larger range of values with less coding length. Thus, they can effectively solve the problem of excessively long coding length in CSL and One-Hot based methods. BCL processes the angle by binarization to obtain a string of codes represented by multiple '0' and '1'. In the encoding of a group of numbers, if any two adjacent codes differ only by one binary number, then this kind of encoding is called Gray Code. In addition, because only one digit is different between the maximum number and the minimum number, it is also called Cyclic Code. The encoding forms between adjacent angles are not much different, which makes GCL also have a certain classification tolerance. The cycle characteristics of GCL are also consistent with circular design idea of CSL. Table 1 compares the coding results of BCL and GCL and Figure 6 shows the relationship between various angle encoding methods.

TABLE 1
The three-digit binary code and gray code corresponding to the decimal number.

Decimal Number	0	1	2	3	4	5	6	7
Binary Coded Label	000	001	010	011	100	101	110	111
Gray Coded Label	000	001	011	010	110	111	101	100

TABLE 2
Comparison of GFlops and Param over rotation detectors, under the same setting and hyperparameters. The baseline is RetinaNet.

Method	ω	GFlops	ΔGFlops	Params	ΔParams	Training	Inference
Reg.	-	139.35	-	36.97 M	-	-	
CSL	1	254.96	+82.96%	45.63 M	+23.42%	∼1/3x	∼1/2x
GCL	1	143.87	**+3.24%**	37.31 M	**+0.92%**	**∼1x**	**∼1x**

The code length of DCL is:

$$L_{dcl} = \lceil \log_2(AR/\omega) \rceil \tag{6}$$

We will use a specific calculation example to explain why the code length has such a large impact on the amount of detection model parameters and calculations. Take RetinaNet as an example, the number of prediction layer channels can be calculated as follows:

$$C_o = A \times L \tag{7}$$

where A represents the number of anchors, and $A = scale\_num \times ratio\_num \times angle\_num$. In this paper $scale\_num = 3, ratio\_num = 7, angle\_num = 1$.

For all prediction layers {P3, P4, P5, P6, P7}, the total Flops and Params are calculated as follows:

$$\begin{aligned} Flops &= \sum_{i=3}^{7} Flops_i = \sum_{i=3}^{7} 2C_i^i K_i^2 H_i W_i C_o^i \\ Params &= C_i K^2 C_o \end{aligned} \tag{8}$$

where K_i, H_i and W_i denote the convolution kernel size of the i-th level prediction layer, and the height and width of the input feature map for the i-th level prediction layer. C_i and C_o represent the number of input and output channels of the i-th level prediction layer. It should be noted that the parameters of different levels of prediction layers are shared.

According to the default settings of our paper, the input image size is 800, $C_i = 256$, $C_o = AL = 21L$, K=3. Then, taking $AR = 180$, $w = 1$ as an example, the code length required by CSL and One-Hot are $L_{onehot} = L_{csl} = 180$, while the code length of DCL is only $L_{dcl} = 8$. The total Flops and Params of all prediction layers are $1,291,175,424L$ and $48,384L$. The huge base make the prediction layer occupy the main calculations and parameters of the model. Therefore, the shortening of the code length is necessary and important. Finally, we have counted the total parameters and calculations of three different models, as shown in Table 2. From the perspective of GFlops and Params, detectors based on CSL have increased by about 82.96% and 45.63%, respectively. In contrast, DCL-based protocol only increases by 3.24% and 0.92%. The training and testing time

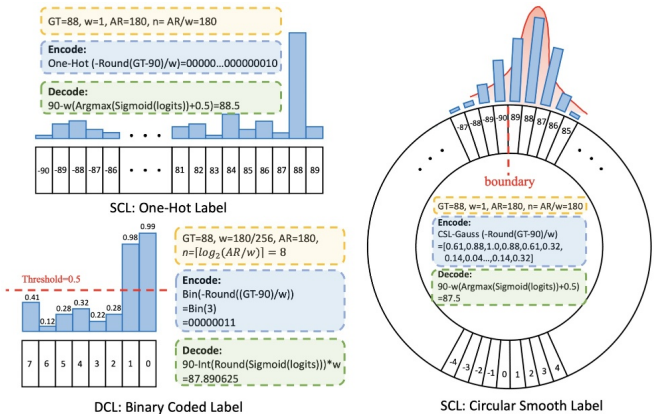


Fig. 7. Examples of encoding and decoding process of One-Hot, CSL-Gaussian and BCL for angle prediction.

of RetinaNet-DCL is about 3 times and 2 times faster than RetinaNet-CSL, respectively.

In the DCL-based method, only the number of categories is a power of 2 to ensure that each coding corresponds to a valid angle. For example, if the 180 degree range is divided into $2^8 = 256$ categories, then the range of each division interval is $\omega = 180/256 = 0.703125°$. According to the $Max(error) = \omega/2$ and $E(error) = \omega/4$, the maximum and expected accuracy error are only $0.3515625°$ and $0.17578125°$, whose influence on final detecton accuracy can be negligible. However, the above condition is not necessary. We find that even with some redundant invalid codes, there is no significant drop in final performance. Equation 9 specifies the encoding and decoding process of DCL (take BCL as an example):

$$\begin{aligned} &\textbf{Encode: } \text{Bin}(-\text{Round}((\theta_{gt} - 90)/\omega)) \\ &\textbf{Decode: } 90 - \omega\text{Int}(\text{Round}(\text{Sigmoid}(logits))) \end{aligned} \tag{9}$$

Figure 7 gives the examples of encoding and decoding process of One-Hot, CSL-Gaussian and BCL for angle prediction.

3.6 Angle Fine-Tuning

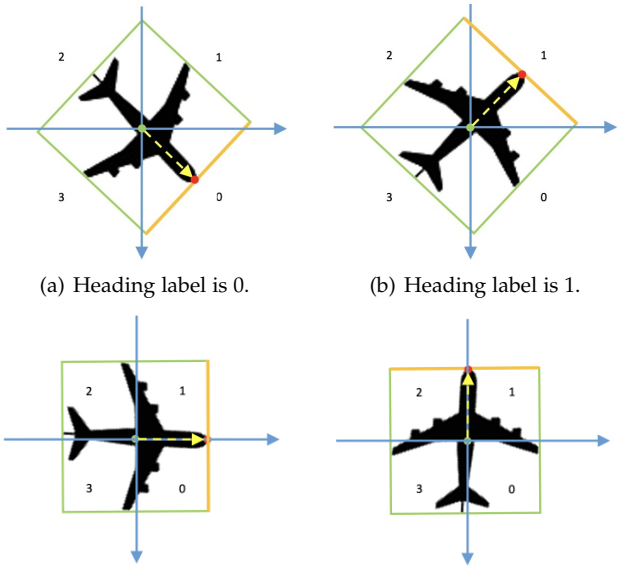
A larger angle discrete granularity ω can alleviate the problem of too many parameters in the prediction layer to a certain extent, and reduce the dependence on the classification ability of the model angle. However, the theoretical angle error in Eq. 1 cannot be ignored at this time. To solve this problem, we predict a smaller angle ε to make up for the error of accuracy caused by angle encoding.

$$\begin{aligned} \theta_{disc} =& \text{Decode}(\text{Encode}(\theta_{gt})) \\ \varepsilon_{gt} =& \theta_{gt} - \theta_{disc}, \quad \varepsilon_{gt} \in \left[-\frac{\omega}{2}, \frac{\omega}{2}\right] \end{aligned} \tag{10}$$

Taking RetinaNet in Figure 1 as an example, we will add an extra one-dimensional output at the prediction layer, denoted as ε_{logits}, and then fine-tune the angle by:

$$\begin{aligned} \varepsilon_{pred} &= (\text{Sigmoid}(\varepsilon_{logits}) - 0.5) * \omega \\ \theta'_{pred} &= \min(\max(\theta_{pred} + \varepsilon_{pred}, -90°), 90°) \end{aligned} \tag{11}$$

where θ_{pred} and θ'_{pred} respectively represent the predicted angle before and after fine-tuning the angle. The $\min$ and $\max$ operations are to avoid PoA.



(a) Heading label is 0. (b) Heading label is 1.

(c) Heading label is 0. (d) Heading label is 1.

Fig. 8. Definition of object heading label. The green bounding box shows the rotation detection. The red dot denotes the head of the object. The green dot indicates the center of the rotating bounding box. The yellow dotted line shows the object's head orientation, and the coordinate quadrant pointed by the yellow dotted line is the heading label.

3.7 Object Heading Detector

Compared with rotation detection, object heading detection is a more fine-grained detection task, which aims to determine the head of the object. Although the rotation detection retains the orientation information of the object, it still cannot determine the accurate head of the object based on the angle of the rotating bounding box alone. By a carefully study, we find that the head of the object must be located in the four sides of the rotating bounding box. Inspired by this discovery, we only need to perform an additional simple four-category to predict the head. Figure 8 shows how the object heading label is defined.

One essential prerequisite for realizing object heading detection is an accurate rotation detector, which is expected to satisfy the following two characteristics:

- In non-boundary situations, the detector can output high-precision rotating bounding boxes.
- In boundary situations, the detector is not sensitive to boundary problem.

Therefore, cascade multi-stage strategy and angle classification technique can be combined to solve the above problems. Based on the above analysis, we have adjusted the entire detector as follows:

- The number of anchors plays a vital role in the performance of the detector, which can be calculated by $num\_scales \times num\_ratios \times num\_angles$. Therefore, we do not use any angle classification technique (e.g. CSL or DCL) in the first stage, so that an appropriate number of anchors can be set in the first stage to provide high-quality initial candidate boxes.
- After using regression prediction to obtain the refined anchor in the first stage, each feature point on

the feature map only retains the refined anchor with the highest confidence. The filtering of the refined anchor makes the parameter A is always equal to 1 at each refined stage, which means that we can use angle classification technique.

- We use 90-degree angle definition method, as shown in Figure 2(a), to reduce AR to a minimum. At the same time, in order to solve the unprocessed boundary problem in the first stage and the EoE problem in the refinement stage, we use the IoU-Smooth L1 loss function [3] in each stage.
- Experimental results show that proper adjacent angle prediction fault tolerance can improve the performance of the detector. When the evaluation standard is not too strict (e.g. DOTA uses AP_{50} as the evaluation metric), we can appropriately increase w and adjust the radius r of the window function to relieve the pressure on the prediction layer.

Combining the head prediction strategy and the above procedures, an efficient object heading detection method is fulfilled, which is called OHDet[4].

3.8 Loss Function Design

Our multi-tasking pipeline contains regression-based prediction branch and angle classification-based prediction branch, to facilitate the performance comparison of the two methods on an equal footing. The model mainly outputs four items for location and size:

$$\begin{aligned} t_x^{'} &= (x^{'} - x_a)/w_a, t_y^{'} = (y^{'} - y_a)/h_a, \\ t_w^{'} &= \log(w^{'}/w_a), t_h^{'} = \log(h^{'}/h_a) \end{aligned} \tag{12}$$

to match the four targets from the ground truth:

$$\begin{aligned} t_x &= (x - x_a)/w_a, t_y = (y - y_a)/h_a, \\ t_w &= \log(w/w_a), t_h = \log(h/h_a) \end{aligned} \tag{13}$$

where x, y, w, h denote the box's center coordinates, width, height and angle, respectively. Variables $x, x_a, x^{'}$ are for the ground truth box, anchor box, and predicted box, respectively (likewise for y, w, h).

As for the angle prediction of the regression branch, we use two forms as the baseline to be compared:

- Direct regression (**Reg.**). The model directly predicts the angle offset $t_\theta^{'}$:

$$\begin{aligned} t_\theta &= (\theta - \theta_a) \cdot \pi/180, \\ t_\theta^{'} &= (\theta^{'} - \theta_a) \cdot \pi/180 \end{aligned} \tag{14}$$

- Indirect regression (**Reg.***). The model predicts two vectors ($t_{\sin\theta}^{'}$ and $t_{\cos\theta}^{'}$) to match the two targets from the ground truth ($t_{\sin\theta}$ and $t_{\cos\theta}$):

$$\begin{aligned} t_{\sin\theta} &= \sin(\theta \cdot \pi/180), t_{\cos\theta} = \cos(\theta \cdot \pi/180), \\ t_{\sin\theta}^{'} &= \sin(\theta^{'} \cdot \pi/180), t_{\cos\theta}^{'} = \cos(\theta^{'} \cdot \pi/180) \end{aligned} \tag{15}$$

To ensure that $t_{\sin\theta}^{'2} + t_{\cos\theta}^{'2} = 1$ is satisfied, we will perform the following normalization processing:

$$t_{\sin\theta}^{'} = \frac{t_{\sin\theta}^{'}}{\sqrt{t_{\sin\theta}^{'2} + t_{\cos\theta}^{'2}}}, \; t_{\cos\theta}^{'} = \frac{t_{\cos\theta}^{'}}{\sqrt{t_{\sin\theta}^{'2} + t_{\cos\theta}^{'2}}} \tag{16}$$

4. https://github.com/SJTU-Thinklab-Det/OHDet_Tensorflow.

Indirect regression is a simpler way to avoid boundary problems. The multi-task loss is defined as follows:

$$\begin{aligned} L =& \frac{\lambda_1}{N_{pos}} \sum_{n=1}^{N_{pos}} L_{reg}(t_n^{'}, t_n) \\ &+ \frac{\lambda_2}{N_{pos}} \sum_{n=1}^{N_{pos}} L_{angle\_cls}(\theta_n^{'}, \theta_n) \\ &+ \frac{\lambda_3}{N_{pos}} \sum_{n=1}^{N_{pos}} L_{reg}(\varepsilon_{gt}, \varepsilon_{pred}) \\ &+ \frac{\lambda_4}{N_{pos}} \sum_{n=1}^{N_{pos}} L_{head}(h_n^{'}, h_n) + \frac{\lambda_5}{N} \sum_{n=1}^{N} L_{cls}(p_n, l_n) \end{aligned} \tag{17}$$

where N and N_{pos} indicate the total number of samples and positive samples, respectively. $t_n^{'}$ denotes the predicted vectors, t_n is the targets vector of ground truth. θ_n and $\theta_n^{'}$ denote the label and prediction of angle, respectively. ε_{gt} and ε_{pred} represent the theoretical angle error and the predicted angle error. h_n and $h_n^{'}$ represent the head of ground truth and prediction bounding box, respectively. l_n represents the label of object, p_n is the probability distribution of various classes calculated by sigmoid function. The hyper-parameter λ_k $(k = 1, 2, ..5)$ control the trade-off and are set to $\{1, 0.5, 20, 0.1, 1\}$ by default. The classification loss L_{cls}, L_{head} and L_{angle_cls} are focal loss [15] or sigmoid cross-entropy loss depends on the detector. The regression loss L_{reg} is smooth L1 loss as used in [16].

4 Experiments

We use Tensorflow [47] to implement the proposed methods on a server with GeForce RTX 2080 Ti and 11G memory. The experiments in this article are initialized by ResNet50 [48] by default unless otherwise specified. We perform experiments on both aerial benchmarks and scene text benchmarks to verify the generality of our techniques. Weight decay and momentum are set 0.0001 and 0.9, respectively. We employ MomentumOptimizer over 4 GPUs with a total of 4 images per minibatch (1 image per GPU). At each pyramid level, we use anchors at seven aspect ratios $\{1, 1/2, 2, 1/4, 4, 1/6, 6\}$, and the settings of the remaining anchor numbers are the same as the original RetinaNet and FPN.

4.1 Benchmarks and Protocls

DOTA [38] is a complex aerial image dataset for object detection, which contains objects exhibiting a wide variety of scales, orientations, and shapes. DOTA-v1.0 contains 2,806 aerial images and 15 common object categories from different sensors and platforms. The fully annotated DOTA-v1.0 benchmark contains 188,282 instances, each of which is labeled by an arbitrary quadrilateral. There are two detection tasks for DOTA: horizontal bounding boxes (HBB) and oriented bounding boxes (OBB). The training set, validation set, and test set account for 1/2, 1/6, 1/3 of the entire data set, respectively. In contrast, DOTA-v1.5 uses the same images as DOTA-v1.0, but extremely small instances (less than 10 pixels) are also annotated. Moreover, a new category, containing 402,089 instances in total is added in this version. While DOTA-v2.0 contains 18 common categories, 11,268

TABLE 3
Statistics on the categories and quantities of OHD-SJTU datasets. Abbreviations of the object categories are in brackets.

Tag	Plane (PL)	Ship (SH)	Small vehicle (SV)	Large vehicle (LV)	Harbor (HA)	Helicopter (HC)
L train	8,614	30,386	26,126	16,969	5,983	630
L val	2,754	9,985	5,438	4,387	2,090	73
S train	559	2,318	-	-	-	-
S val	223	1,025	-	-	-	-

TABLE 4
Comparison between different angle prediction methods on DOTA-v1.0 validation set. For CSL-based protocol, the label mode and windows function radius are set to Gaussian and 6, respectively. The baseline is RetinaNet. The angle range is $[-90°, 90°)$.

Angle Pred.	ω	AP_{50}	AP_{75}	$AP_{50:95}$
Reg. ($\Delta\theta$)	-	62.21	26.06	31.49
Reg.* ($\sin\theta$, $\cos\theta$)	-	63.23	30.63	33.19
Cls. CSL	1	64.40	32.58	35.04
Cls. BCL	180/256	65.93	**35.66**	**36.71**
Cls. GCL	180/256	**66.13**	33.65	36.34

images and 1,793,658 instances. Compared to DOTA-v1.5, it includes the new categories. The 11,268 images in DOTA-v2.0 are split into training, validation, test-dev, and test-challenge sets. We divide the images into 600×600 subimages with an overlap of 150 pixels and scale it to 800×800. With all these processes, we obtain about 27,000 patches. The short names for categories are defined as (abbreviation-full name): PL-Plane, BD-Baseball diamond, BR-Bridge, GTF-Ground field track, SV-Small vehicle, LV-Large vehicle, SH-Ship, TC-Tennis court, BC-Basketball court, ST-Storage tank, SBF-Soccer-ball field, RA-Roundabout, HA-Harbor, SP-Swimming pool, HC-Helicopter, CC-container crane, AP-airport and HP-helipad.

ICDAR2015 [49] is the Challenge 4 of ICDAR 2015 Robust Reading Competition, which is commonly used for oriented scene text detection and spotting. This dataset includes 1,000 training images and 500 testing images. In training, we first train our model using 9,000 images from ICDAR 2017 MLT training and validation datasets, then we use 1,000 training images to fine-tune our model.

ICDAR 2017 MLT [50] is a multi-lingual text dataset. It includes 7,200 training images, 1,800 validation images and 9,000 testing images. The dataset is composed of complete scene images in 9 languages, and text regions can be in arbitrary orientations, being more diverse and challenging.

HRSC2016 [51] contains images from two scenarios: ships on sea and ships close inshore. All images are collected from six harbors around the world. The training, validation and test set include 436, 181 and 444 images, respectively.

FDDB [52] is a dataset designed for unconstrained face detection, in which faces have a wide variability of face scales, poses, and appearance. This dataset contains annotations for 5,171 faces in a set of 2,845 images taken from the faces in the Wild dataset [53]. In our paper, we manually use 70% as the training set and the rest as the validation set.

OHD-SJTU is our newly collected and public dataset for rotation detection and object heading detection. OHD-SJTU contains two different scale datasets, called OHD-SJTU-S and OHD-SJTU-L. OHD-SJTU-S is collected publicly from Google Earth with 43 large scene images sized $10,000 \times 10,000$ pixels and $16,000 \times 16,000$. It contains two object categories (ship and plane) and 4,125 instances (3,343 ships and 782 planes). Each object is labeled by an arbitrary quadrilateral, and the first marked point is the head position of the object to facilitate head prediction. We randomly select 30 original images as the training and validation set, and 13 images as the testing set. Figure 9 shows some samples of annotated subimages in OHD-SJTU-S. The scenes cover a decent variety of road scenes and typical: cloud occlusion, seamless dense arrangement, strong changes in illumination/exposure, mixed sea and land scenes and large number of interfering objects. In contrast, OHD-SJTU-L adds more categories and instances, such as small vehicle, large vehicle, harbor, and helicopter. The additional data comes from DOTA-v1.0, but we reprocess the annotations and add the annotations of the object head. According to statistics, OHD-SJTU-L contains six object categories and 113,435 instances. The statistical details are shown in Table 3. Compared with the AP_{50} used by DOTA as the evaluation indicator, OHD-SJTU uses a more stringent $AP_{50:95}$ to measure the performance of the method, which poses a further challenge to the high accuracy of the detector. We divide the training and validation images into 600×600 subimages with an overlap of 150 pixels and scale it to 800×800. In the process of cropping the image with the sliding window, objects whose center point is in the subimage are kept.

All the used datasets are trained by 20 epochs in total, and the learning rate is reduced tenfold at 12 epochs and 16 epochs, respectively. The initial learning rates for Reti-

TABLE 5
Comparison of the four window functions (with radius set to 6) on the DOTA-v1.0 test set. 5-mAP refers to the mean average precision of the five categories with large aspect ratio. mAP means mean average precision of all 15 categories. The base method is RetinaNet-CSL. Note the EoE problem exists for the angle range $[-90°, 0°)$.

Angle Range	Label Mode	BR	SV	LV	SH	HA	5-mAP	mAP
$[-90°, 0°)$	Pulse	9.80	28.04	11.42	18.43	23.35	18.21	39.52
	Rectangular	37.62	54.28	48.97	62.59	50.26	50.74	58.86
	Triangle	37.25	54.45	44.01	60.03	52.20	49.59	60.15
	Gaussian	**41.03**	**59.63**	**52.57**	**64.56**	**54.64**	**54.49**	**63.51**
$[-90°, 90°)$	Pulse	13.95	16.79	6.50	16.80	22.48	15.30	42.06
	Rectangular	36.14	60.80	50.01	65.75	**53.17**	53.17	61.98
	Triangle	32.69	47.25	44.39	54.11	41.90	44.07	57.94
	Gaussian	**40.55**	**66.77**	**51.50**	**73.60**	46.05	**55.69**	**65.69**

TABLE 6
Comparison of detection mAP under different radius on the DOTA-v1.0 test set. The angle range, label mode and ω are set to $[-90°, 90°)$, Gaussian and 1, respectively.

Method	r=0	r=2	r=4	r=6	r=8
RetinaNet-CSL	40.78	59.23	62.12	**65.69**	63.99
FPN-CSL	48.08	70.18	70.09	**70.92**	69.75

TABLE 7
Comparison of detection results under different angle discrete granularity ω on the DOTA-v1.0 test set. The angle range is $[-90°, 90°)$. For CSL-based protocol, the label mode and windows function radius are set to Gaussian and 1, respectively.

Granularity	ω=30	ω=18	ω=10	ω=3	ω=1
RetinaNet-CSL	40.81	66.10	**67.38**	64.81	58.92
Granularity	ω=180/4	ω=180/32	ω=180/64	ω=180/128	ω=180/256
RetinaNet-GCL	62.38	65.59	**67.02**	65.14	64.97

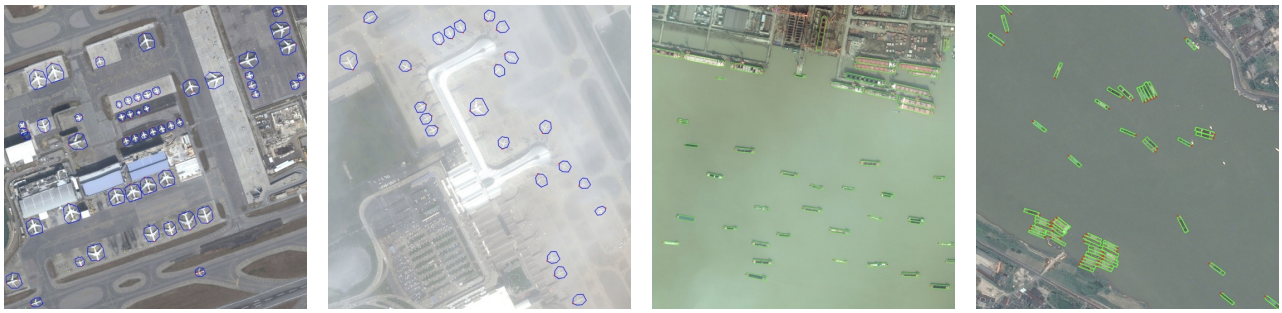
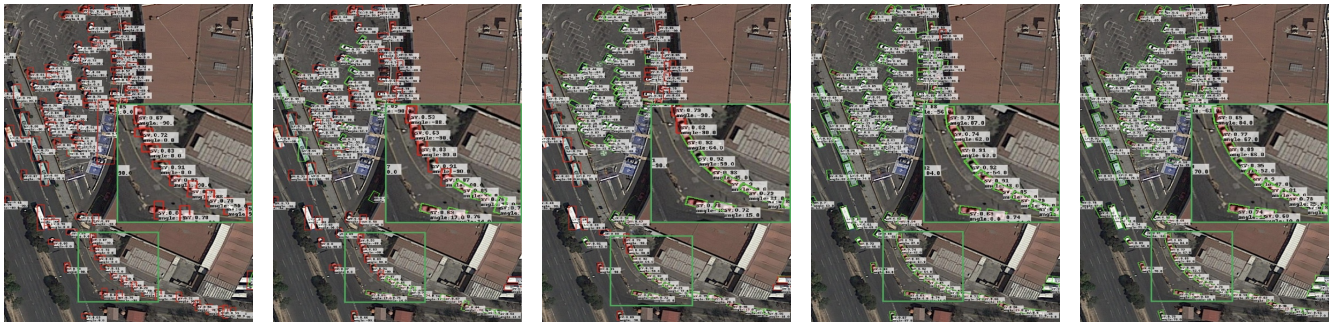


Fig. 9. Samples of annotated subimages in our collected OHD-SJTU-S. Each object is labeled by an arbitrary quadrilateral, and the first marked point is the head position of the object to facilitate head prediction.



(a) radius=0 (b) radius=2 (c) radius=4 (d) radius=6 (e) radius=8

Fig. 10. Visualization of detection results (RetinaNet-H CSL-Based) under different window function radius. The red bounding box indicates that no orientation and scale information has been learned, and the green bounding box is the correct detection result.

TABLE 8
Comparison of detection results under different angle discretization granularities denoted by ω on the DOTA-v1.0 validation set.

Method	ω	BR	SV	LV	SH	HA	5-mAP$_{50}$	mAP$_{50}$	mAP$_{75}$	mAP$_{50:95}$
Reg	-	34.52	51.42	50.32	73.37	55.93	53.12	62.21	26.07	31.49
CSL	180/180	35.94	53.42	61.06	81.81	62.14	58.87	64.40	32.58	35.04
BCL	180/4	30.74	40.54	50.98	72.07	59.54	50.77	62.38	24.88	31.01
	180/8	36.65	52.58	60.46	82.24	61.60	58.71	**66.17**	33.14	35.77
	180/32	**39.83**	54.41	60.62	80.81	60.32	**59.20**	65.93	**35.66**	**36.71**
	180/64	38.22	**54.70**	60.16	80.75	60.11	58.79	65.00	34.31	36.00
	180/128	36.76	53.73	**61.35**	**82.52**	58.42	58.56	65.14	34.28	35.69
	180/180	37.42	53.72	58.70	80.73	**63.31**	58.78	65.83	33.94	36.35
	180/256	37.66	53.83	60.66	80.43	60.74	58.66	64.97	33.52	35.21
	180/512	37.93	53.85	58.52	80.04	60.87	58.24	64.88	33.09	34.99
GCL	180/4	30.90	41.20	48.30	72.93	60.16	50.70	62.98	23.83	30.81
	180/8	36.88	51.10	59.81	82.40	61.57	58.35	65.23	33.92	35.29
	180/32	38.04	**54.77**	60.88	**82.75**	61.24	**59.54**	65.11	**34.67**	36.15
	180/64	**38.05**	54.36	60.59	81.84	60.39	59.05	64.78	33.23	35.67
	180/128	37.74	54.36	59.43	81.15	60.51	58.64	**66.13**	33.65	**36.34**
	180/256	35.81	53.78	58.35	81.45	59.84	57.85	64.87	33.77	35.97
	180/512	37.99	54.23	**61.61**	80.84	**62.13**	59.36	64.34	34.08	35.92

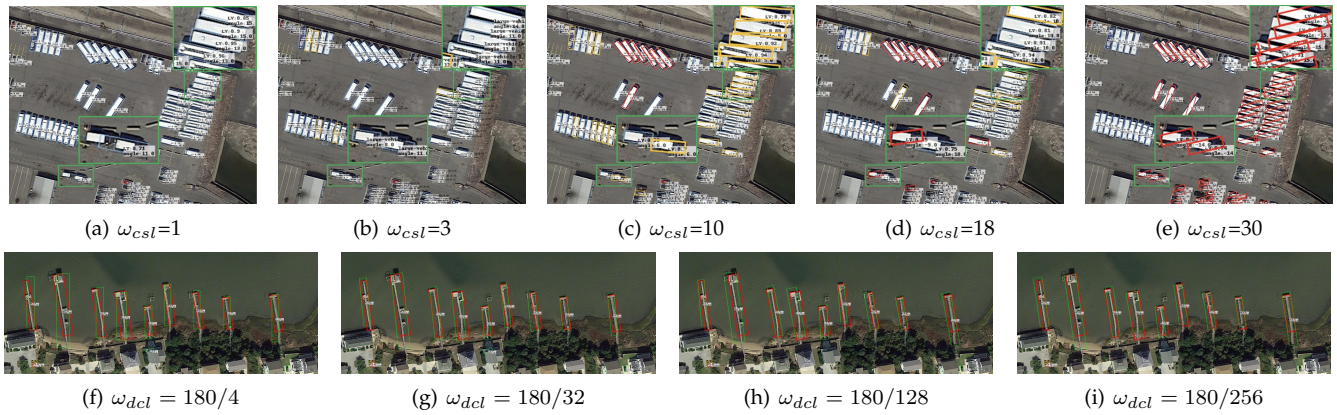
naNet and FPN are 5e-4 and 1e-3 respectively. The number of image iterations per epoch for DOTA-v1.0, DOTA-v1.5, DOTA-v2.0, ICDAR2015, MLT, HRSC2016, FDDB, OHD-SJTU-S and OHD-SJTU-L are 54k, 64k, 80k, 10k, 10k, 5k, 4k, 5k and 10k respectively, and doubled if data augmentation and multi-scale training are used.

4.2 Ablation Study

Comparison of classification and regression. Table 4 compares four different angle prediction methods on DOTA-v1.0 validation set, two of which are based on regression and the other two are via classification. Among them, the indirect regression (**Reg.***) is a relatively simple way to eliminate boundary problem, so it has a better performance than the direct regression (**Reg.**). In contrast, the classification-based prediction methods CSL, BCL and GCL outperform, achieving 35.04%, 36.71% and 36.34% on the DOTA-v1.0 validation set, respectively.

Comparison of four window functions in CSL method. Table 5 shows the performance comparison of the four window functions on the DOTA-v1.0 dataset. It also details the accuracy of the five categories with larger aspect ratio and more border cases in the dataset. We believe that these categories can better reflect the advantages of our method. In general, the Gaussian window function performs best, while the pulse function performs worst because it has not learned any orientation and scale information. Figures 4(f)-4(i) show the visualization of the four window functions. According to Figure 4(i) and Figure 4(j), the 180°-CSL-based protocol obviously has better boundary prediction due to the EoE problem still exists in the 90°-CSL-based protocol. Figure 4 shows the consistent results with the those in Table 5.

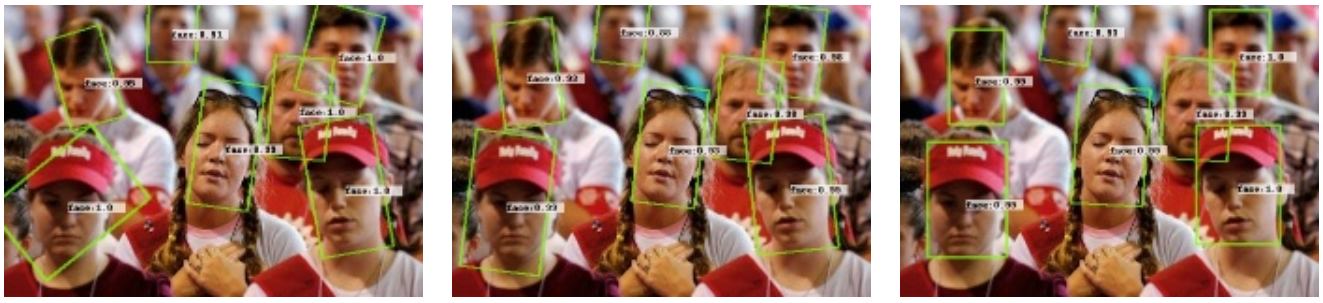
Suitable window radius in CSL method. The Gaussian window form has shown best performance, while here we study the effect of radius of the window function. When the radius is too small, the window function tends to a pulse function. Conversely, the discrimination of all predictable results becomes smaller when the radius is too large. Therefore, we choose a suitable radius range from 0 to 8, Table 6 shows the performance of the two detectors in this range. Although both detectors achieve the best performance with a radius of 6, the single-stage detection method is more sensitive to radius. We speculate that the instance-level feature extraction capability (like RoI Pooling [16] and RoI Align [34]) in the two-stage detector is stronger than the image-level in the single-stage detector. Therefore, the two-stage detection method can distinguish the difference



(a) ω_{csl}=1 (b) ω_{csl}=3 (c) ω_{csl}=10 (d) ω_{csl}=18 (e) ω_{csl}=30

(f) $\omega_{dcl} = 180/4$ (g) $\omega_{dcl} = 180/32$ (h) $\omega_{dcl} = 180/128$ (i) $\omega_{dcl} = 180/256$

Fig. 11. Visualization of detection results (RetinaNet-Based) under different angle discrete granularity ω. For CSL, The red bounding box indicates that there is a large angle prediction error, and the orange bounding box indicates an acceptable angle prediction error. For DCL, the red and green box indicate ground truth and prediction.



(a) RetinaNet (b) CSL (c) DCL

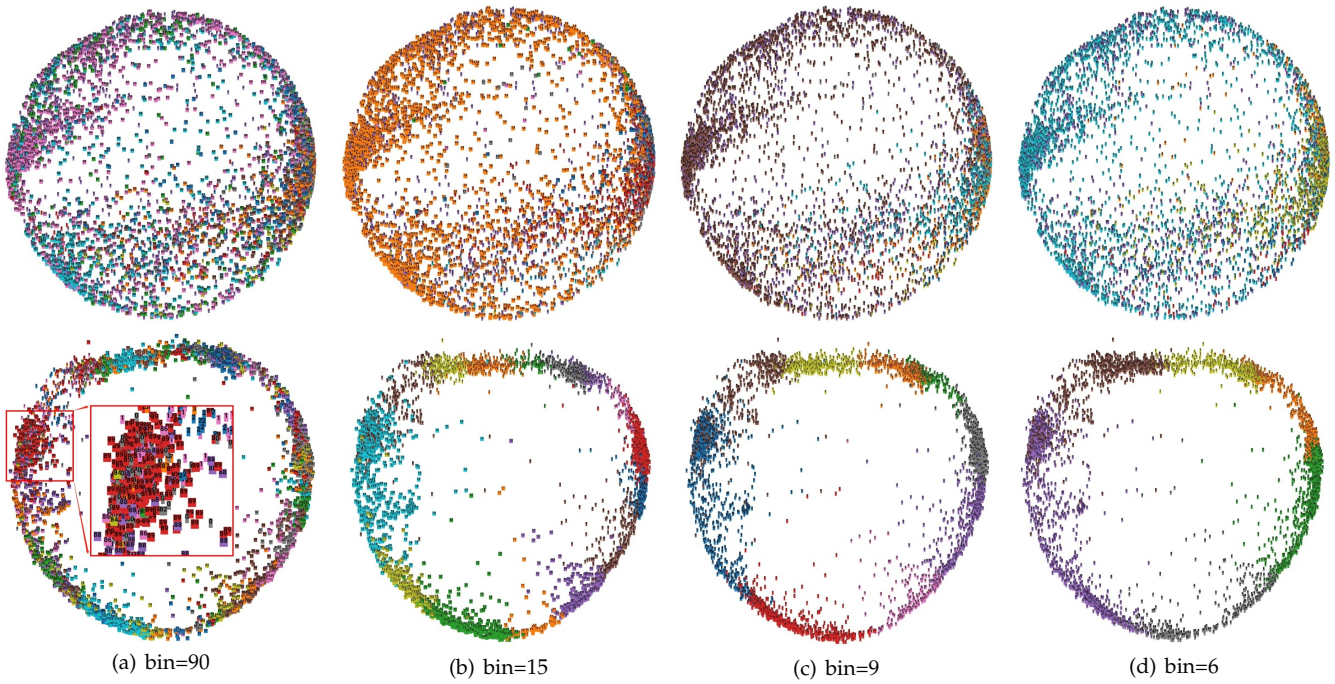
Fig. 12. Visual comparison between classification-based and regression-based protocols on the FDDB dataset.

TABLE 9
Comparison between classification-based and regression-based protocols on the DOTA-v1.0 test set. 'H' and 'R' represent the horizontal and rotating anchors, respectively.

Baseline	Angle Range	Angle Pred.	PoA	EoE	Label Mode	BR	SV	LV	SH	HA	5-mAP	mAP
RetinaNet-H	$[-90°, 0°)$	Reg. ($\Delta\theta$)	✓	✓	-	41.15	53.75	48.30	55.92	55.77	50.98	63.18
		Cls.: CSL		✓	Gaussian	41.03	59.63	52.57	64.56	54.64	54.49 **(+3.51)**	63.51 **(+0.33)**
		Cls.: GCL		✓	-	42.49	62.38	49.13	69.06	56.40	55.89 **(+4.09)**	64.94 **(+1.76)**
	$[-90°, 90°)$	Reg. ($\Delta\theta$)	✓			38.31	60.48	49.77	68.29	51.28	53.63	64.17
		Reg.* ($\sin\theta, \cos\theta$)			-	41.52	63.94	44.95	71.18	53.22	54.96 **(+1.33)**	65.78 **(+1.61)**
		Cls.: CSL			Gaussian	42.25	68.28	54.51	72.85	53.10	58.20 **(+4.57)**	67.38 **(+3.21)**
		Cls.: GCL			-	39.78	67.20	56.02	74.10	53.82	58.18 **(+4.55)**	67.02 **(+2.85)**
		Cls.: BCL			-	41.40	65.82	56.27	73.80	54.30	58.32 **(+4.69)**	67.39 **(+3.22)**
RetinaNet-R	$[-90°, 0°)$	Reg. ($\Delta\theta$)	✓	✓	-	32.27	64.64	71.01	68.62	53.52	58.01	62.76
		Cls.: CSL		✓	Gaussian	35.14	63.21	73.92	69.49	55.53	59.46 **(+1.45)**	65.45 **(+2.69)**
R^3Det	$[-90°, 0°)$	Reg. ($\Delta\theta$)	✓	✓	-	44.15	75.09	72.88	86.04	61.01	67.83	70.66
		Cls.: BCL		✓	Gaussian	46.84	74.87	74.96	85.70	57.72	68.02 **(+0.19)**	71.21 **(+0.55)**
FPN-H	$[-90°, 0°)$	Reg. ($\Delta\theta$)	✓	✓	-	44.78	70.25	71.13	68.80	54.27	61.85	68.25
		Cls.: CSL		✓	Gaussian	45.46	70.22	71.96	76.06	54.84	63.71 **(+1.86)**	69.02 **(+0.77)**
	$[-90°, 90°)$	Reg. ($\Delta\theta$)	✓		-	45.88	69.37	72.06	72.96	62.31	64.52	69.45
		Cls.: CSL			Gaussian	47.90	69.66	74.30	77.06	64.59	66.70 **(+2.18)**	70.92 **(+1.47)**
		Cls.: GCL			-	47.56	69.81	74.03	76.56	64.29	66.45 **(+1.93)**	70.83 **(+1.38)**

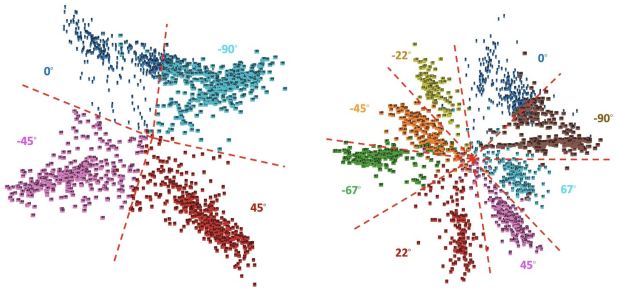
between the two approaching angles. Figure 10 compares visualizations using different window radius. When the radius is 0, the detector cannot learn any orientation and scale information, which is consistent with the performance of the pulse function above. As the radius becomes larger and more optimal, the detector can learn the angle in any direction. Please refer to the enlarged part of the figure.

Angle discretization granularity ω. In general, the smaller the angle discrete granularity ω, the more accurate the angle predicted by the model, as shown in Figure 11. However, considering that the criterion for judging the object to be detected is that IoU is greater than 0.5 (such as the DOTA-v1.0 dataset). Therefore, a proper ω can make the model have a certain degree of fault tolerance, and can achieve better detection performance. It can be seen from Table 7 that when the discrete granularity is 10, the CSL-based protocol can achieve the highest performance on the DOTA-v1.0 dataset, and when ω is 30, the excessive angular prediction error makes the performance of the model drop sharply. Discrete granularity ω can be approximated as a CSL technique with



(a) bin=90 (b) bin=15 (c) bin=9 (d) bin=6

Fig. 13. Angular feature visualization of the 90-CSL-FPN detector on the DOTA-v1.0 dataset. First, we divide the entire angular range into several bins, and bins are different between columns. The two rows show two-dimensional feature visualizations of pulse and Gaussian function, respectively. Each point represents an RoI of the test set with an index of the bin it belongs to.



(a) $\omega = 180/4$ (b) $\omega = 180/8$

Fig. 14. Angular feature visualization of the RetinaNet-DCL. The red dotted lines divide the categories.

a rectangular window function, which has a certain tolerance in the divided angle interval. The difference between them is that CSL smooths between adjacent angle intervals.

Similar conclusions can still be obtained in the DCL method, as shown in Table 7 and Table 8. In general, the smaller ω, the higher theoretical upper bound of the model's performance. However, the decrease of ω will lead to an increase in the number of angle categories, which poses a challenge to the angle classification performance of the model. Therefore, we need to explore the impact of ω on the detection performance under different IoU thresholds, and find a suitable range of ω. In order to get the performance indicators under different IoU threshold, we conduct experiments on the DOTA-v1.0 validation set, and the number of image iterations per epoch is 40k. According to Table 8, when the number of angle categories is between 32 and 128, the performance of the model reaches its peak. If the number of categories is too small, the theoretical accuracy loss is too large, resulting in a sharp drop in performance; if the number of categories is too large, the angle classification network of the model cannot be effectively processed and the performance will decrease slightly. Figure 11 shows the comparison of angle estimates under different ω. Compared with CSL, DCL can set a smaller w (e.g. $w < 1$) without too much parameter overhead, and no need to adjust the window function radius at the same time.

Redundant invalid coding. To make each code have a corresponding different angle value, the number of categories must be a power of 2 in the DCL-based method. However, this is not required. When we only set 180 categories, about 76 codings are invalid, but BCL-based method can still achieve good performance, at 36.35% as shown in Table 8. We also artificially increase the length based on the theoretical shortest code length to increase the proportion of invalid codes, and the performance is only slightly reduced.

Performance of CSL and DCL on other detectors. Four detectors in Table 9, including RetinaNet-H, RetinaNet-R, R^3Det and FPN-H, are used to compare the performance differences among CSL-based, DCL-based and regression-based protocols. The former two are single-stage detectors, whose anchor format is different. One of the remaining two is a cascade multi-stage strategy based method and the other is a classic two-stage detection method. It can be clearly seen that CSL and DCL have better detection ability for objects with large aspect ratios and more boundary conditions. It also should be noted that CSL and DCL are designed to solve the boundary problem, whose proportion in the entire dataset is relatively small, so the overall performance (mAP) is not as obvious as the five categories listed (5-mAP). Overall, the CSL-based and DCL-based rotation detection algorithms are indeed better baseline choices than the angle

TABLE 10
Comparison between classification-based and regression-based protocols on the text dataset ICDAR2015, MLT, aerial dataset HRSC2016, and face dataset FDDB. Note 2007 and 2012 in bracket means using the 2007 and 2012 evaluation metric, respectively. Except for FDDB's baseline is RetinaNet [15], the others are FPN [18].

Method	ICDAR2015			MLT			HRSC2016		FDDB	
	Recall	Precision	Hmean	Recall	Precision	Hmean	mAP (2007)	mAP (2012)	AP_{50} (2012)	AP_{75} (2012)
baseline	81.81	83.07	82.44	56.15	80.26	66.08	88.33	94.70	95.92	55.81
CSL	83.00	84.30	83.65 **(+1.21)**	56.72	80.77	66.64 **(+0.56)**	89.62 **(+1.29)**	96.10 **(+1.40)**	96.64 **(+0.72)**	73.22 **(+17.41)**
GCL	82.56	84.72	83.63 **(+1.19)**	57.54	80.65	67.16 **(+1.08)**	89.56 **(+1.23)**	96.02 **(+1.32)**	96.16 **(+0.24)**	74.06 **(+18.25)**

TABLE 11
Ablative study by accuracy (%) of CSL and DCL on the OBB task of DOTA-v1.0/v1.5/v2.0.

Method	Angle Pred.	DOTA-v1.0	DOTA-v1.5	DOTA-v2.0
RetinaNet-H	Reg. ($\Delta\theta$)	64.17	56.10	43.06
	Cls: CSL	67.38	58.55	43.34
	Cls: BCL	**67.39**	**59.38**	**45.46**

TABLE 12
Ablation experiment of our proposed angle fine-tuning technique on the HRSC2016 dataset. The baseline is RetinaNet.

Method	Fine-Tune	ω_{csl}=18, ω_{gcl}=180/8		ω_{csl}=10, ω_{gcl}=180/128		ω_{csl}=1, ω_{gcl}=180/256	
		AP_{50}	AP_{75}	AP_{50}	AP_{75}	AP_{50}	AP_{75}
CSL		46.72	3.68	74.07	16.89	81.17	44.75
	✓	76.17	20.47	81.41	32.57	81.37	46.51
GCL		57.34	5.36	77.77	30.35	76.60	34.20
	✓	66.70	28.27	78.07	32.62	76.56	34.60

TABLE 13
Accuracy and speed on HRSC2016. Here (07) and (12) means using the 2007 and 2012 evaluation metric, respectively.

Method	Backbone	mAP (07)	mAP (12)
R^2CNN [8]	ResNet101	73.07	79.73
RC1 & RC2 [51]	VGG16	75.70	–
RRPN [9]	ResNet101	79.08	85.64
R^2PN [54]	VGG16	79.60	–
RetinaNet-H [1]	ResNet101	82.89	89.27
RRD [10]	VGG16	84.30	–
RoI-Transformer [2]	ResNet101	86.20	–
Gliding Vertex [22]	ResNet101	88.20	–
BBAVectors [55]	ResNet101	88.60	–
DRN [56]	Hourglass104	–	92.70
CenterMap OBB [57]	ResNet50	–	92.80
SBD [20]	ResNet50	–	93.70
RetinaNet-R [1]	ResNet101	89.18	95.21
R^3Det [1]	ResNet101	89.26	96.01
CSL	ResNet101	**89.62**	96.10
GCL	ResNet101	89.56	96.02
BCL	ResNet101	89.46	**96.41**

TABLE 14
Detection accuracy on ICDAR2015. † and ‡ denote that the method uses external data and stronger pre-trained weight, respectively.

Method	Venue	Backbone	Precison	Recall	F-measure
CTPN [58]	ECCV'16	VGG16	74.2	51.5	60.8
EAST† [6]	CVPR'17	VGG16	83.5	73.4	78.2
DeepReg [59]	ICCV'17	VGG16	82.0	80.0	81.0
RRPN† [9]	TMM'18	VGG16	82.0	73.0	77.0
PixelLink [60]	AAAI'18	VGG16	82.9	81.7	82.3
PAN‡ [61]	ICCV'19	ResNet18	82.9	77.8	80.3
TextField†‡ [62]	TIP'19	VGG16	84.3	80.5	82.4
TextDragon†‡ [63]	ICCV'19	VGG16	84.8	81.8	83.1
DBNet(736)‡ [33]	AAAI'20	ResNet18	86.8	78.4	82.3
DBNet(1,152)‡ [33]	AAAI'20	ResNet50	91.8	83.2	87.3
PAN++‡ [64]	TPAMI'21	ResNet18	86.7	78.4	82.3
PAN++†‡ [64]	TPAMI'21	ResNet50	91.4	83.9	**87.5**
PolarMask++‡ [65]	TPAMI'21	ResNet50	86.2	80.0	83.4
CSL(800) (FPN based)	-	ResNet50	84.3	83.0	83.7
GCL(800) (FPN based)	-	ResNet50	84.7	82.6	83.6

regression-based protocol.

Performance of CSL and DCL on other datasets. To further verify that CSL-based and DCL-based protocols are better than baseline, we have also verified it in other datasets, including the text dataset ICDAR2015, MLT, and another remote sensing dataset HRSC2016. These three datasets are single-class object detection datasets, whose objects have a large aspect ratio. Although boundary conditions still account for a small proportion of these datasets, CSL and DCL still shows stronger performance advantage. As shown in Table 10, compared with the regression-based protocol, the CSL-based protocol is improved by 1.21%, 0.56%, and 1.29% (1.4%) respectively under the same experimental configuration. The same improvement is also reflected in the DCL-based protocol. For face dataset FDDB (Figure 12), AP_{50} cannot better reflect the advantages of the proposed technique due to the aspect ratio of face is small. In contrast, CSL/GCL shows a great performance improvement on AP_{75}, about 17.41%/18.25%. For more challenging datasets (e.g. DOTA-v1.5, DOTA-v2.0 in Table 11), CSL/DCL still has a steady improvement. These experimental results provide strong support for demonstrating the versatility of the CSL-based and DCL-based protocols.

Angle Fine-Tuning. Table 12 compares the performance before and after using angle fine-tuning under different sizes of ω on the HRSC2016. The low-precision indicator (mAP_{50}) has a certain tolerance for angle errors. Take two objects with the same scale and the same center as an example, when their aspect ratio is 1:9, their IoU is still close to 0.7 when the angle deviation is $5°$, which shows that mAP_{50} cannot better reflect the advantages of the angle fine-tuning mechanism when ω is small. When $\omega_{csl} = 1$, $\omega_{gcl} = 180/256$, the maximum angle that can be fine-tuned is only $0.5°$ and $0.35°$ according to Eq. 10, so the effectiveness of fine-tuning technique on mAP_{50} is not significant in Table 12. In contrast, the high-precision indicator (mAP_{75}) well reflects the advantages of the angle fine-tuning technique, and the larger the ω, the more significant its advantages. When ω becomes larger that the theoretical accuracy exceeds the tolerance of mAP_{50}, angle fine-tuning technique can effectively adjust the prediction angle to reduce the negative impact of theoretical angle errors and improve both mAP_{50} and mAP_{75}.

Visual analysis of angular features. By zooming in on part of Figure 4(i), we show that the prediction of the boundary conditions become continuous (for example, two large vehicle in the same direction predicted $90°$ and $-88°$, respectively). This phenomenon reflects the purpose of de-

TABLE 15
Detection accuracy (AP) on each category of object and overall performance (mAP) on the DOTA-v1.0 test set, using different backbones.

Method	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
FR-O [38]	ResNet101	79.09	69.12	17.17	63.49	34.20	37.16	36.20	89.19	69.60	58.96	49.4	52.52	46.69	44.80	46.30	52.93
IENet [66]	ResNet101	80.20	64.54	39.82	32.07	49.71	65.01	52.58	81.45	44.66	78.51	46.54	56.73	64.40	64.24	36.75	57.14
R-DFPN [5]	ResNet101	80.92	65.82	33.77	58.94	55.77	50.94	54.78	90.33	66.34	68.66	48.73	51.76	55.10	51.32	35.88	57.94
TOSO [67]	ResNet101	80.17	65.59	39.82	39.95	49.71	65.01	53.58	81.45	44.66	78.51	48.85	56.73	64.40	64.24	36.75	57.92
PIoU [68]	DLA-34 [69]	80.9	69.7	24.1	60.2	38.3	64.4	64.8	90.9	77.2	70.4	46.5	37.1	57.1	61.9	64.0	60.5
R^2CNN [8]	ResNet101	80.94	65.67	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	60.67
RRPN [9]	ResNet101	88.52	71.20	31.66	59.30	51.85	56.19	57.25	90.81	72.84	67.38	56.69	52.84	53.08	51.94	53.58	61.01
Axis Learning [70]	ResNet101	79.53	77.15	38.59	61.15	67.53	70.49	76.30	89.66	79.07	83.53	47.27	61.01	56.28	66.06	36.05	65.98
ICN [4]	ResNet101	81.40	74.30	47.70	70.30	64.90	67.80	70.00	90.80	79.10	78.20	53.60	62.90	67.00	64.20	50.20	68.20
RADet [71]	ResNeXt101 [72]	79.45	76.99	48.05	65.83	65.46	74.40	68.86	89.70	78.14	74.97	49.92	64.63	66.14	71.58	62.16	69.09
RoI-Transformer [2]	ResNet101	88.64	78.52	43.44	75.92	68.81	73.68	83.59	90.74	77.27	81.46	58.39	53.54	62.83	58.93	47.67	69.56
P-RSDet [73]	ResNet101	89.02	73.65	47.33	72.03	70.58	73.71	72.76	90.82	80.12	81.32	59.45	57.87	60.79	65.21	52.59	69.82
CAD-Net [74]	ResNet101	87.8	82.4	49.4	73.5	71.1	63.5	76.7	90.9	79.2	73.3	48.4	60.9	62.0	67.0	62.2	69.9
O^2-DNet [75]	Hourglass104 [76]	89.31	82.14	47.33	61.21	71.32	74.03	78.62	90.76	82.23	81.36	60.93	60.17	58.21	66.98	61.03	71.04
AOOD [77]	DPN [78]	89.99	81.25	44.50	73.20	68.90	60.33	66.86	90.89	80.99	86.23	64.98	63.88	65.24	68.36	62.13	71.18
Cascade-FF [79]	ResNet152	89.9	80.4	51.7	**77.4**	68.2	75.2	75.6	90.8	78.8	84.4	62.3	64.6	57.7	69.4	50.1	71.8
BBAVectors [55]	ResNet101	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
SCRDet [3]	ResNet101	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
SARD [80]	ResNet101	89.93	84.11	54.19	72.04	68.41	61.18	66.00	90.82	87.79	86.59	65.65	64.04	66.68	68.84	68.03	72.95
GLS-Net [81]	ResNet101	88.65	77.40	51.20	71.03	73.30	72.16	84.68	90.87	80.43	85.38	58.33	62.27	67.58	70.69	60.42	72.96
DRN [56]	Hourglass104	89.71	82.34	47.22	64.10	76.22	74.43	85.84	90.57	86.18	84.89	57.65	61.93	69.30	69.63	58.48	73.23
FADet [82]	ResNet101	**90.21**	79.58	45.49	76.41	73.18	68.27	79.56	90.83	83.40	84.68	53.40	65.42	74.17	69.69	64.86	73.28
MFIAR-Net [83]	ResNet152	89.62	84.03	52.41	70.30	70.13	67.64	77.81	90.85	85.40	86.22	63.21	64.14	68.31	70.21	62.11	73.49
R^3Det [1]	ResNet152	89.49	81.17	50.53	66.10	70.92	78.66	78.21	90.81	85.26	84.23	61.81	63.77	68.16	69.83	67.17	73.74
RSDet [21]	ResNet152	90.1	82.0	53.8	68.5	70.2	78.7	73.6	**91.2**	87.1	84.7	64.3	68.2	66.1	69.3	63.7	74.1
Gliding Vertex [22]	ResNet101	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
Mask OBB [37]	ResNeXt-101	89.56	**85.95**	54.21	72.90	76.52	74.16	85.63	89.85	83.81	86.48	54.89	69.64	73.94	69.06	63.32	75.33
FFA [84]	ResNet101	90.1	82.7	54.2	75.2	71.0	79.9	83.5	90.7	83.9	84.6	61.2	68.0	70.7	76.0	63.7	75.7
APE [85]	ResNeXt-101	89.96	83.62	53.42	76.03	74.01	77.16	79.45	90.83	87.15	84.51	**67.72**	60.33	**74.61**	71.84	65.55	75.75
CenterMap OBB [57]	ResNet101	89.83	84.41	**54.60**	70.25	77.66	78.32	87.19	90.66	84.89	85.27	56.46	69.23	74.13	71.56	66.06	76.03
CSL	ResNet152	89.33	84.88	53.70	75.68	77.57	80.21	84.18	89.80	86.57	86.22	71.86	64.48	73.48	**74.84**	66.05	77.26
GCL	ResNet152	89.26	83.59	53.05	72.76	78.13	81.97	86.94	90.36	85.98	86.94	66.19	65.56	73.29	70.56	**69.99**	76.97
BCL	ResNet152	89.32	83.54	53.60	72.70	**78.94**	**82.66**	87.27	90.69	86.61	**87.98**	66.49	66.97	73.20	70.65	69.90	**77.37**

signing the CSL: the labels are periodic (circular) and the prediction of adjacent angles has a certain tolerance. In order to confirm that the angle classifier has indeed learned this property, we visually analyze the angular features of each region of interest (RoI) in the FPN detector by principal component analysis (PCA) [86], as shown in Figure 13. The detector does not learn the orientation information well when the pulse window function is used. It can be seen from the first row of Figure 13 that the feature distribution of RoI is relatively random, and the prediction results of some angles occupy the vast majority. For the Gaussian function, the feature distribution is obvious a ring structures, and the features of adjacent angles are close to each other and have a certain overlap. It is this property that helps CSL-based detectors to eliminate boundary problems and accurately obtain the orientation and scale information. We also show the visualization results when the number of angle categories are 4 and 8, as shown in Figure 14.

4.3 Comparison with the State-of-the-Art Methods

Results on HRSC2016. The HRSC2016 contains lots of large aspect ratio ship instances with arbitrary orientation, which poses a huge challenge to the positioning accuracy of the detector. Table 13 shows that our models achieve superior performances, about 89.62% (96.10%), 89.56% (96.02%) and 89.46% (96.41%), for CSL, GCL and BCL respectively.

Results on ICDAR2015. Scene text detection has been a well-studied field and many advanced techniques are specifically designated to texts while our proposed model is general for rotation detection. Moreover, to achieve competitive performance in text detection, it often further involves non-trivial processing and like using external data, such as RRPN [9], PAN [61], TextField [62], FOTS [7] and TextDragon [63], powerful pre-trained weights on SynthText [87] or COCO [88] (e.g. SBD [20], SegLink [89], TextField [62], DBNet [33], PAN++ [64], PolarMask++ [65]), model ensemble (e.g. Inceptext [90]). so we are refrained to over compare with state-of-the-art text detection models tailored to texts, and become more focused on the proposed components themselves which we do not want to couple with other factors. Table 14 shows the comparative experiments on the ICDAR2015, and our methods obtain the competitive F-measure without using external data or powerful pre-trained weight.

TABLE 16
Detection accuracy on each object (AP) and overall performance (mAP) on OHD-SJTU-S. 'H' and 'R' represent the horizontal and rotating anchors, respectively. Here the numbers in the subscript of AP i.e. 50, 75, 95 represent the threshold of IoU. Note our model's performance gain is even pronounced on the challenging AP_{75} and $AP_{50:95}$ metrics.

Method	PL (AP_{50})	SH (AP_{50})	AP_{50}	AP_{75}	$AP_{50:95}$
R^2CNN [8]	**90.91**	77.66	84.28	55.00	52.80
RRPN [9]	90.14	76.13	83.13	27.87	40.74
RetinaNet-H [1]	90.86	66.32	78.59	58.45	53.07
RetinaNet-R [1]	90.82	**88.14**	**89.48**	74.62	61.86
R^3Det [1]	90.82	85.59	88.21	67.13	56.19
OHDet (ours)	90.74	87.59	89.06	**78.55**	**63.94**

Results on DOTA-v1.0. We choose DOTA-v1.0 as the main validation dataset due to the complexity of the remote sensing and the large number of small, cluttered and rotated objects in the dataset. The used data augmentation include random horizontal, vertical flipping, random graying, and random rotation. Training and testing scale is set to $[400, 600, 720, 800, 1000, 1100]$. As shown in Table 15, CSL-based, GCL-based, BCL-based protocols show competitive performance, at the accuracy of 77.26%, 76.97% and 77.37%, respectively.

Results on OHD-SJTU. We assess the performance of state-of-the-art rotation object detection methods on OHD-SJTU,

(a) Ship (b) Harbor

(c) Vehicle (d) Plane

Fig. 15. Detection examples of our method in large-scale scenarios on OHD-SJTU. Our method can both effectively handle the dense and rotating cases. The blue border in the bounding box denotes the predicted head of the object.

TABLE 17
Detection accuracy on each object (AP) and overall performance (mAP) on OHD-SJTU-L. Note our model's performance gain is even pronounced on the challenging AP_{75} and $AP_{50:95}$ metrics.

Method	PL	SH	SV	LV	HA	HC	AP_{50}	AP_{75}	$AP_{50:95}$
R^2CNN [8]	90.02	80.83	63.07	64.16	**66.36**	55.94	70.06	32.70	35.44
RRPN [9]	89.55	82.60	57.36	72.26	63.01	45.27	68.34	22.03	31.12
RetinaNet-H [1]	**90.22**	80.04	63.32	63.49	63.73	53.77	69.10	35.90	36.89
RetinaNet-R [1]	90.00	86.90	63.24	**86.90**	62.85	52.35	**72.78**	40.13	40.58
R^3Det [1]	89.89	**87.69**	**65.20**	78.95	57.06	53.50	72.05	36.51	38.57
OHDet (ours)	89.73	86.63	61.37	78.80	63.76	**54.62**	72.49	**43.60**	**41.29**

mainly include R^2CNN, RRPN, RetinaNet, R^3Det. All experiments are based on the same setting, using ResNet101 as the backbone. Except for data augmentation (include random horizontal, vertical flipping, random graying, and random rotation) is used in OHD-SJTU-S, no other tricks are used. As shown in Table 16, the large number of dense ship with large aspect ratio in the OHD-SJTU-S brings huge challenges to the high-precision detection capabilities of the detector. RetinaNet-R and R^3Det use rotating anchor and cascade structure respectively, which makes them stand out in high-precision indicators, such as AP_{75}. OHDet uses CSL and IoU smooth L1 in combination, which not only reduces the amount of model parameters, but also avoids the side effects of boundary problems. Then OHDet is further combined with the cascade structure to achieve the best detection performance, at about 63.94%. Especially in AP_{75} and $AP_{50:95}$, our method is 3.93% and 2.08% higher than the second-best method. Similar conclusions can also be obtained from OHD-SJTU-L. As shown in Table 17, OHDet achieves a notable advantage in high-precision indicators and achieves the best performance (about 41.29%).

4.4 Object Heading Detection Experiment

For rotation object detection, the expected output includes the object category and the IoU between the predicted bounding box and ground truth, which is often compared with a certain threshold, e.g. 0.5. In contrast, object heading detection additionally outputs the head prediction results. Three indicators are used to measure the performance: OBB mAP, OHD mAP and Head Accuracy. OBB mAP is a detection metric without considering object head prediction, which is consistent with rotation detection. While OHD mAP additionally considers the accuracy of object head prediction. Therefore, for the same model, the upper bound of OHD mAP is OBB mAP when the head prediction accuracy is 100%. The Head Accuracy indicates the accuracy of head prediction in all the detection boxes judged as true positive

TABLE 18
Results of object heading detection on OHD-SJTU dataset, covering six categories under three different settings of IoU.

Tag	Task	PL	SH	SV	LV	HA	HC	IoU_{50}	IoU_{75}	$IoU_{50:95}$
S	OBB	90.73	88.59	–	–	–	–	89.66	75.62	61.49
	OHD	76.89	86.40	–	–	–	–	81.65	65.51	55.09
	Head	90.91	94.87	–	–	–	–	92.89	93.81	94.25
L	OBB	89.62	85.58	48.45	76.55	61.43	33.87	65.92	38.80	37.66
	OHD	59.93	47.57	26.59	35.32	41.29	17.53	38.04	24.86	22.97
	Head	74.43	68.39	60.15	57.79	76.66	49.06	64.41	65.17	64.12

(TP). Figure 15 visualizes the object heading detection on different categories.

Table 18 shows the performance of OHDet on the two sub-data sets of OHD-SJTU. Due to the high image resolution and clear objects in OHD-SJTU-S, a very high head prediction accuracy can be achieved: 94.25%. However, head prediction still faces challenges in complex environments. For example, the head prediction accuracy is only 64.12% on OHD-SJTU-L dataset, and the detection performance drops from 37.66% to 22.97%. It can be seen that the addition of head prediction conditions will greatly reduce the detection performance. The main difficulties of inaccurate head prediction are as follows: extremely similar head and tail (e.g. LV, SH, SV), fuzzy small object (e.g. SV, SH), densely arranged (e.g. LV, SH, SV), and small sample size (e.g. HC). Figure 16 shows some bad cases. We hope that the open source of the OHD-SJTU can promote the research of related methods.

5 Conclusion

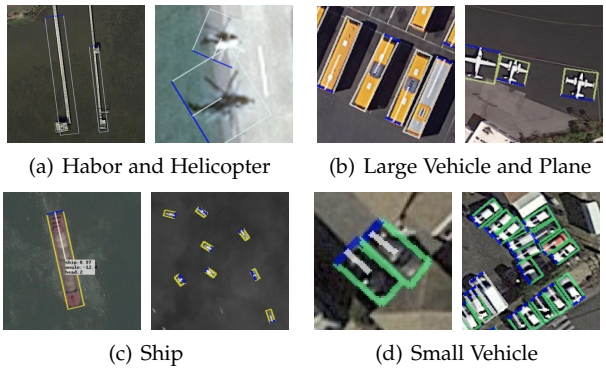
In this paper, we have particularly identified the boundary problems as faced by different regression-based rotation detection methods. The main cause of boundary problems based on regression methods is that the ideal predictions are beyond the defined range. Therefore, considering the prediction of the object angle as a classification problem to better limit the prediction results, and then we design a Circular Smooth Label (CSL) to adapt to the periodicity of the angle and increase the tolerance of classification between adjacent angles with little accuracy error. We also introduce 窗函数 four window functions in CSL and explore the effect of 半径 different window radius sizes on detection performance.

To reduce the excessive model parameters caused by CSL, we further design a Densely Coded Label (DCL), which greatly reduces the length of the encoding while ensuring that the angle prediction accuracy is not reduced. An angle fine-tuning mechanism is also devised to eliminate the theoretical prediction errors caused by angle dispersion which has been a common issue in whatever CSL and DCL. Our devised angle high-precision classification is also the first application in rotation detection.

We further fulfill the function of object heading detection, called OHDet, which is used to find the head of object. We annotate and release a dataset for rotation detection and object heading detection, called OHD-SJTU. Extensive experiments and visual analysis on different detectors and datasets show the effectiveness of our approach.

Acknowledgment

This work was partly supported by National Key Research and Development Program of China (2020AAA0107600), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102) and National Natural Science Foundation of China (U20B2068, 61972250). Xue Yang is partly supported by Wu Wen Jun Honorary Doctoral Scholarship, AI Institute, Shanghai Jiao Tong University.



(a) Habor and Helicopter (b) Large Vehicle and Plane

(c) Ship (d) Small Vehicle

Fig. 16. Illustration for the main failure cases of head prediction: extremely similar head and tail, fuzzy small object, densely arranged, and small sample size, etc.

References

[1] X. Yang, J. Yan, Z. Feng, and T. He, "R3det: Refined single-stage detector with feature refinement for rotating object," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, 2021, pp. 3163–3171.

[2] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning roi transformer for oriented object detection in aerial images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2849–2858.

[3] X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang, Z. Guo, X. Sun, and K. Fu, "Scrdet: Towards more robust detection for small, cluttered and rotated objects," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8232–8241.

[4] S. M. Azimi, E. Vig, R. Bahmanyar, M. Körner, and P. Reinartz, "Towards multi-class object detection in unconstrained remote sensing imagery," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 150–165.

[5] X. Yang, H. Sun, K. Fu, J. Yang, X. Sun, M. Yan, and Z. Guo, "Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks," *Remote Sensing*, vol. 10, no. 1, p. 132, 2018.

[6] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5551–5560.

[7] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "Fots: Fast oriented text spotting with a unified network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5676–5685.

[8] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo, "R2cnn: rotational region cnn for orientation robust scene text detection," *arXiv preprint arXiv:1706.09579*, 2017.

[9] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3111–3122, 2018.

[10] M. Liao, Z. Zhu, B. Shi, G.-s. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5909–5918.

[11] M. Liao, B. Shi, and X. Bai, "Textboxes++: A single-shot oriented scene text detector," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3676–3690, 2018.

[12] X. Shi, S. Shan, M. Kan, S. Wu, and X. Chen, "Real-time rotation-invariant face detection with progressive calibration networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2295–2303.

[13] C. Huang, H. Ai, Y. Li, and S. Lao, "High-performance rotation invariant multiview face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 671–686, 2007.

[14] H. A. Rowley, S. Baluja, and T. Kanade, "Rotation invariant neural network-based face detection," in *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*. IEEE, 1998, pp. 38–44.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[16] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[19] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 379–387.

[20] Y. Liu, S. Zhang, L. Jin, L. Xie, Y. Wu, and Z. Wang, "Omnidirectional scene text detection with sequential-free box discretization," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2019.

[21] W. Qian, X. Yang, S. Peng, J. Yan, and Y. Guo, "Learning modulated loss for rotated object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 2458–2466.

[22] Y. Xu, M. Fu, Q. Wang, Y. Wang, K. Chen, G.-S. Xia, and X. Bai, "Gliding vertex on the horizontal bounding box for multi-oriented object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 4, pp. 1452–1459, 2020.

[23] X. Yang and J. Yan, "Arbitrary-oriented object detection with circular smooth label," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 677–694.

[24] X. Yang, L. Hou, Y. Zhou, W. Wang, and J. Yan, "Dense label encoding for boundary discontinuity free rotation detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 819–15 829.

[25] X. Yang, Y. Zhou, and J. Yan, "Alpharotate: A rotation detection benchmark using tensorflow," *arXiv preprint arXiv:2111.06677*, 2021.

[26] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 21–37.

[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[29] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 734–750.

[30] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6569–6578.

[31] X. Zhou, J. Zhuo, and P. Krahenbuhl, "Bottom-up object detection by grouping extreme and center points," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 850–859.

[32] Z. Tian, M. Shu, P. Lyu, R. Li, C. Zhou, X. Shen, and J. Jia, "Learning shape-aware embedding for scene text detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4234–4243.

[33] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 474–11 481.

[34] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.

[35] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, "Solo: Segmenting objects by locations," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 649–665.

[36] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *European Conference on Computer Vision*. Springer, 2020, pp. 282–298.

[37] J. Wang, J. Ding, H. Guo, W. Cheng, T. Pan, and W. Yang, "Mask obb: A semantic attention-based mask oriented bounding box representation for multi-category object detection in aerial images," *Remote Sensing*, vol. 11, no. 24, p. 2930, 2019.

[38] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "Dota: A large-scale dataset for object detection in aerial images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3974–3983.

[39] K.-R. Kim, W. Choi, Y. J. Koh, S.-G. Jeong, and C.-S. Kim, "Instance-level future motion estimation in a single image based on ordinal regression," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 273–282.

[40] X. Yang, H. Sun, X. Sun, M. Yan, Z. Guo, and K. Fu, "Position detection and direction prediction for arbitrary-oriented ships via multitask rotation region convolutional neural network," *IEEE Access*, vol. 6, pp. 50 839–50 849, 2018.

[41] K. Liu and G. Mattyus, "Fast multiclass vehicle detection on aerial images," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 9, pp. 1938–1942, 2015.

[42] L. Liu, Z. Pan, and B. Lei, "Learning a rotation invariant detector with rotatable bounding box," *arXiv preprint arXiv:1711.09405*, 2017.

[43] Q. An, Z. Pan, L. Liu, and H. You, "Drbox-v2: An improved detector with rotatable boxes for target detection in sar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 8333–8349, 2019.

[44] S. M. Azimi, R. Bahmanyar, C. Henry, and F. Kurz, "Eagle: Large-scale vehicle detection dataset in real-world scenarios using aerial imagery," in *2020 25th International Conference on Pattern Recognition*. IEEE, 2021, pp. 6920–6927.

[45] F. Heath, "Origins of the binary code," *Scientific American*, vol. 227, no. 2, pp. 76–83, 1972.

[46] G. Frank, "Pulse code communication," Mar. 17 1953, uS Patent 2,632,058.

[47] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[49] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu *et al.*, "Icdar 2015 competition on robust reading," in *2015 13th International Conference on Document Analysis and Recognition*. IEEE, 2015, pp. 1156–1160.

[50] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon *et al.*, "Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt," in *2017 14th IAPR International Conference on Document Analysis and Recognition*, vol. 1. IEEE, 2017, pp. 1454–1459.

[51] Z. Liu, L. Yuan, L. Weng, and Y. Yang, "A high resolution optical satellite image dataset for ship recognition and some new baselines," in *Proceedings of the International Conference on Pattern Recognition Applications and Methods*, vol. 2, 2017, pp. 324–331.

[52] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," 2010.

[53] T. L. Berg, A. C. Berg, J. Edwards, and D. A. Forsyth, "Who's in the picture," in *Advances in neural information processing systems*, 2005, pp. 137–144.

[54] Z. Zhang, W. Guo, S. Zhu, and W. Yu, "Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 11, pp. 1745–1749, 2018.

[55] J. Yi, P. Wu, B. Liu, Q. Huang, H. Qu, and D. Metaxas, "Oriented object detection in aerial images with box boundary-aware vectors," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2021, pp. 2150–2159.

[56] X. Pan, Y. Ren, K. Sheng, W. Dong, H. Yuan, X. Guo, C. Ma, and C. Xu, "Dynamic refinement network for oriented and densely packed object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 207–11 216.

[57] J. Wang, W. Yang, H.-C. Li, H. Zhang, and G.-S. Xia, "Learning center probability map for detecting objects in aerial images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 4307–4323, 2020.

[58] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 56–72.

[59] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep direct regression for multi-oriented scene text detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 745–753.

[60] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[61] W. Wang, E. Xie, X. Song, Y. Zang, W. Wang, T. Lu, G. Yu, and C. Shen, "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8440–8449.

[62] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai, "Textfield: Learning a deep direction field for irregular scene text detection," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5566–5579, 2019.

[63] W. Feng, W. He, F. Yin, X.-Y. Zhang, and C.-L. Liu, "Textdragon: An end-to-end framework for arbitrary shaped text spotting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9076–9085.

[64] W. Wang, E. Xie, X. Li, X. Liu, D. Liang, Y. Zhibo, T. Lu, and C. Shen, "Pan++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[65] E. Xie, W. Wang, D. Mingyu, Z. Ruimao, and P. Luo, "Polarmask++: Enhanced polar representation for single-shot instance segmentation and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[66] Y. Lin, P. Feng, and J. Guan, "Ienet: Interacting embranchment one stage anchor free detector for orientation aerial object detection," *arXiv preprint arXiv:1912.00969*, 2019.

[67] P. Feng, Y. Lin, J. Guan, G. He, H. Shi, and J. Chambers, "Toso: Student'st distribution aided one-stage orientation target detection in remote sensing images," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 4057–4061.

[68] Z. Chen, K. Chen, W. Lin, J. See, H. Yu, Y. Ke, and C. Yang, "Piou loss: Towards accurate oriented object detection in complex environments," in *European Conference on Computer Vision*. Springer, 2020, pp. 195–211.

[69] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2403–2412.

[70] Z. Xiao, L. Qian, W. Shao, X. Tan, and K. Wang, "Axis learning for orientated objects detection in aerial images," *Remote Sensing*, vol. 12, no. 6, p. 908, 2020.

[71] Y. Li, Q. Huang, X. Pei, L. Jiao, and R. Shang, "Radet: Refine feature pyramid network and multi-layer attention network for arbitrary-oriented object detection of remote sensing images," *Remote Sensing*, vol. 12, no. 3, p. 389, 2020.

[72] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1492–1500.

[73] L. Zhou, H. Wei, H. Li, W. Zhao, Y. Zhang, and Y. Zhang, "Arbitrary-oriented object detection in remote sensing images based on polar coordinates," *IEEE Access*, vol. 8, pp. 223 373–223 384, 2020.

[74] G. Zhang, S. Lu, and W. Zhang, "Cad-net: A context-aware detection network for objects in remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 10 015–10 024, 2019.

[75] H. Wei, Y. Zhang, Z. Chang, H. Li, H. Wang, and X. Sun, "Oriented objects as pairs of middle lines," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 169, pp. 268–279, 2020.

[76] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 483–499.

[77] F. Zou, W. Xiao, W. Ji, K. He, Z. Yang, J. Song, H. Zhou, and K. Li, "Arbitrary-oriented object detection via dense feature fusion and attention model for remote sensing super-resolution image," *Neural Computing and Applications*, pp. 1–14, 2020.

[78] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng, "Dual path networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 4467–4475.

[79] L. Hou, K. Lu, J. Xue, and L. Hao, "Cascade detector with feature fusion for arbitrary-oriented objects in remote sensing images," in *2020 IEEE International Conference on Multimedia and Expo*. IEEE, 2020, pp. 1–6.

[80] Y. Wang, Y. Zhang, Y. Zhang, L. Zhao, X. Sun, and Z. Guo, "Sard: Towards scale-aware rotated object detection in aerial imagery," *IEEE Access*, vol. 7, pp. 173 855–173 865, 2019.

[81] C. Li, B. Luo, H. Hong, X. Su, Y. Wang, J. Liu, C. Wang, J. Zhang, and L. Wei, "Object detection based on global-local saliency constraint in aerial images," *Remote Sensing*, vol. 12, no. 9, p. 1435, 2020.

[82] C. Li, C. Xu, Z. Cui, D. Wang, T. Zhang, and J. Yang, "Feature-attentioned object detection in remote sensing imagery," in *2019 IEEE International Conference on Image Processing*. IEEE, 2019, pp. 3886–3890.

[83] F. Yang, W. Li, H. Hu, W. Li, and P. Wang, "Multi-scale feature integrated attention-based rotation network for object detection in vhr aerial images," *Sensors*, vol. 20, no. 6, p. 1686, 2020.

[84] K. Fu, Z. Chang, Y. Zhang, G. Xu, K. Zhang, and X. Sun, "Rotation-aware and multi-scale convolutional neural network for object detection in remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 161, pp. 294–308, 2020.

[85] Y. Zhu, J. Du, and X. Wu, "Adaptive period embedding for representing oriented objects in aerial images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 10, pp. 7247–7257, 2020.

[86] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[87] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2315–2324.

[88] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[89] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2550–2558.

[90] Q. Yang, M. Cheng, W. Zhou, Y. Chen, M. Qiu, and W. Lin, "Inceptext: a new inception-text module with deformable psroi pooling for multi-oriented scene text detection," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 1071–1077.