Gliding Vertex on the Horizontal Bounding Box for Multi-Oriented Object Detection

Yongchao Xu, Mingtao Fu, Qimeng Wang, Yukang Wang, Kai Chen, Gui-Song Xia, *Senior Member, IEEE*, and Xiang Bai, *Senior Member, IEEE*

Abstract—Object detection has recently experienced substantial progress. Yet, the widely adopted horizontal bounding box representation is not appropriate for ubiquitous oriented objects such as objects in aerial images and scene texts. In this paper, we propose a simple yet effective framework to detect multi-oriented objects. Instead of directly regressing the four vertices, we glide the vertex of the horizontal bounding box on each corresponding side to accurately describe a multi-oriented object. Specifically, We regress four length ratios characterizing the relative gliding offset on each corresponding side. This may facilitate the offset learning and avoid the confusion issue of sequential label points for oriented objects. To further remedy the confusion issue for nearly horizontal objects, we also introduce an obliquity factor based on area ratio between the object and its horizontal bounding box, guiding the selection of horizontal or oriented detection for each object. We add these five extra target variables to the regression head of faster R-CNN, which requires ignorable extra computation time. Extensive experimental results demonstrate that without bells and whistles, the proposed method achieves superior performances on multiple multi-oriented object detection benchmarks including object detection in aerial images, scene text detection, pedestrian detection in fisheye images.

Index Terms—Object detection, R-CNN, multi-oriented object, aerial image, scene text, pedestrian detection

1 Introduction

Object detection has achieved a considerable progress thanks to convolutional neural networks (CNNs). The state-of-the-art methods [1], [2], [3] usually aim to detect objects via regressing horizontal bounding boxes. Yet multi-oriented objects are ubiquitous in many scenarios. Examples are objects in aerial images and scene texts. Horizontal bounding box 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.

Recent advances in multi-oriented object detection are mainly driven by adaption of classical object detection methods using rotated bounding boxes [4], [5] or quadrangles [6], [7], [8] to represent multi-oriented objects. Though these existing adaptions of horizontal object detection methods to multi-oriented object detection have achieved promising results, they still face some limitations. For detection using rotated bounding boxes, the accuracy of angle prediction is critical. A minor angle deviation leads to important IoU drop, resulting in inaccurate object detection. This problem is more prominent for detecting long oriented objects such as bridges and harbors in aerial images and Chinese text lines in scene images. The methods based on quadrangle regression usually have ambiguity in defining the ground-truth order of four vertices, yielding unexpected detection results for objects of some orientations.

Some other methods [9], [10], [11] alternatively detect horizontal object parts followed by a grouping process. Yet, such grouping process step is usually heuristic and time-consuming. Describing an oriented object as its segmentation mask [12] is another alternative solution. However, this often results in split and/or merged components, requiring a heavy and time-consuming post-processing.

In this paper, we propose a simple yet effective framework to deal with multi-oriented object detection. Specifically, we propose to glide each vertex of the horizontal bounding box on the corresponding side to accurately describe a multi-oriented object. This results in a novel representation by adding four gliding offset variables to classical horizontal bounding box representation. Put it simply, we regress four length ratios that characterize the relative gliding offset (see Fig. 1) on each side of horizontal bounding box. Such representation may be less sensitive to offset prediction error than angle prediction error in rotated bounding box representation. By limiting the offset on the corresponding side of horizontal bounding box, we may facilitate offset learning and also avoid the confusion for sequential label points in directly regressing the four vertices of oriented objects. To further get rid of confusion issue for nearly horizontal objects, we also introduce an obliquity factor based on area ratio between the multi-oriented object and its horizontal bounding box. As depicted in Fig. 1, this obliquity factor guides us to select the horizontal detection for nearly horizontal objects and oriented detection for oriented objects. It is noteworthy that the proposed method only introduces five additional target variables, requiring ignorable extra computation time.

In summary, the main contribution of this paper are three folds: 1) We introduce a simple yet effective representation for oriented objects, which is rather robust to offset prediction error and does not have the confusion issue. 2) We propose an obliquity factor that effectively guides the selection of horizontal detection for nearly horizontal objects and oriented detection for others, remedying the confusion issue for nearly horizontal objects. 3) Without bells and whistles (e.g., cascade refinement or attention mechanism), the proposed method outperforms some state-of-the-art methods on multiple multi-oriented object detection benchmarks.

2 Related Work

2.1 Deep General Object Detection

Object detection aims to detect general objects in images with horizontal bounding boxes. Recent mainstream CNN-based methods can be roughly summarized into top-down and bottom-up methods. Top-down methods directly detect entire objects. They can be further categorized into two classes: two-stage and single-stage methods. R-CNN and its variances [1], [3], [13], [14], [15] are representative two-stage methods. They first generate object proposals and then use the features of these proposals to predict object categories and refine the bounding boxes. YOLO and its variances [2], [16], [17], SSD [18], and RetinaNet [19] are representative single-stage methods. They predict bounding boxes directly from deep feature maps instead of region proposals. Bottom-up methods rise recently by predicting object parts followed by a grouping process. CornerNet [20], ExtremeNet [21], and CenterNet [22] are recently proposed in succession. They attempt to predict some keypoints of objects such as corners or extreme points, which are then grouped into bounding boxes. Center points are also used by [21], [22] as supplemental information for grouping.

- *Y. Xu, M. Fu, Q. Wang, Y. Wang, and X. Bai are with the School of Electronic Information and Communications, Huazhong University of Science and Technology (HUST), Wuhan, Hubei 430074, China. E-mail: {yongchaoxu, mingtaofu, qimengwang, wangyk, xbai}@hust.edu.cn.*
- *G. S. Xia is with the LIEMARS, Wuhan University, Wuhan, Hubei 430072, China. E-mail: guisong.xia@whu.edu.cn.*
- *K. Chen is with the Shanghai Jiaotong University, Shanghai 200240, China, and also with the Onyou Inc., Shenzhen, Guangdong 518057, China. E-mail: kchen@sjtu.edu.cn.*

Manuscript received 4 Sept. 2019; revised 27 Dec. 2019; accepted 11 Feb. 2020. Date of publication 18 Feb. 2020; date of current version 4 Mar. 2021.
(Corresponding author: Xiang Bai.)
Recommended for acceptance by J. Wang.
Digital Object Identifier no. 10.1109/TPAMI.2020.2974745

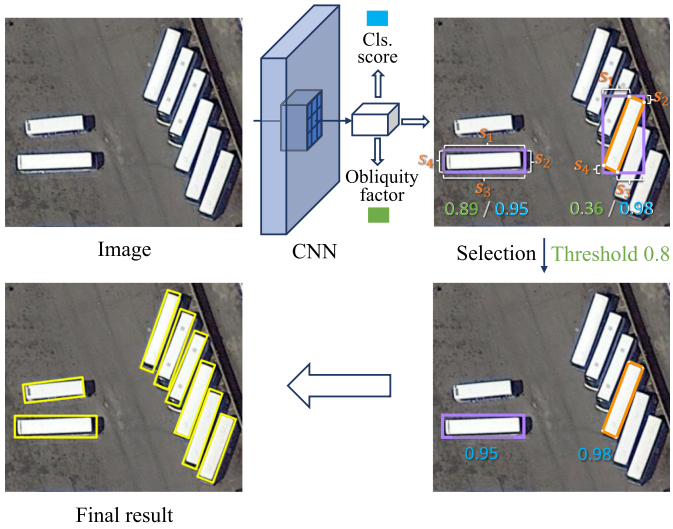

Fig. 1. Pipeline of the proposed method. An image is fed into a CNN, which outputs a classification score (blue value), a horizontal bounding box, four length ratios between each segment s_i and corresponding side, and an obliquity factor (green value) for each detection. Based on obliquity factor, we select horizontal box (in purple) or oriented detection (in orange) as the final result. Best viewed in electronic version.

2.2 Multi-Oriented Object Detection

*Object detection in aerial images*is chanllenging because of huge scale variations and arbitrary orientations. Extensive studies have been devoted to this task. The baselines on the popular dataset DOTA [23] replace horizontal box regression of faster R-CNN with regression of four vertices of quadrangle representation. Many methods resort to rotated bounding box representation. Rotated RPN is exploited in [24], [25], which involves more anchors and thus requires more runtime. Ding *et al.* [5] propose an RoI transformer that transforms horizontal proposals to rotated ones, on which the rotated bounding box regression is performed. Azimi *et al.* [26] adopt an image-cascade network to extract multi-scale features. Yang *et al.* [27] employ multi-dimensional attention to extract robust features, better coping with complex backgrounds. Zhang *et al.* [28] propose to learn global and local contexts together to enhance the features.

*Oriented scene text detection*is a challenging problem due to arbitrary orientations. The mainstream CNN-based detectors can be roughly divided into regression-based and segmentation-based [12], [29] methods. We focus on regression-based methods. Most methods directly predict entire texts using rotated bounding box or quadrangle representation. Ma *et al.* [30] employ rotated RPN in the framework of faster R-CNN [1] to generate rotated proposals and further perform rotated bounding box regression. Liu *et al.* [31] propose to use quadrangle sliding windows to match texts with perspective transformation. TextBoxes++ [6] adopts vertex regression on SSD [18]. RRD [32] further improves TextBoxes++ [6] by decoupling classification and bounding box regression on rotation-invariant and rotation-sensitive features, respectively, making the regression more accurate for long texts. Both EAST [4] and Deep direct regression [7] perform rotated bounding box regression and/or vertex regression at each location.

*Pedestrian detection in fisheye images*is different from general pedestrian detection because pedestrians in fisheye images are often multi-oriented. Seidel *et al.* [33] propose to transform omnidirectional images into perspective ones, on which the detection is applied. Such transformation introduces extra computation time. Based on the prior knowledge that objects in fisheye images are radial, Tamura *et al.* [34] propose to train a general object detector with rotated images and then determine the orientations based on the relative positions of object centers *w.r.t.* the image center.

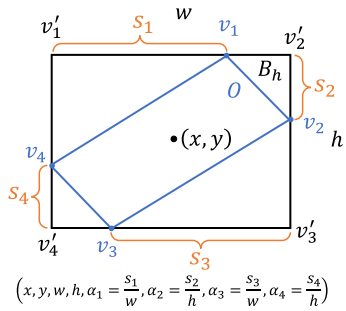

Fig. 2. Illustration of proposed representation for an oriented object O based on four intersecting points $\{v_i\}$ between O and its horizontal bounding box $B_h = (v_1', v_2', v_3', v_4') = (x, y, w, h)$. We adopt $(x, y, w, h, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$ to represent oriented objects.

2.3 Comparison With Related Works

Compared with the related works, the proposed method targets on general and ubiquitous multi-oriented object detection with a simple yet effective framework. By gliding the vertex of horizontal bounding box on each corresponding side and a novel divide-and-conquer selection scheme for nearly horizontal and oriented objects, the proposed method may better learn the offset for accurate multi-oriented object detection and does not suffer from confusion issue. Furthermore, the proposed method may be complementary and easily plugged into many existing methods focusing on enhancing features. To equip them with the proposed approach, we only need to replace rotated bounding box or vertex regression by regressing the four length ratios and obliquity factor in addition to horizontal bounding box. Such modification requires ignorable extra runtime.

3 Proposed Method

3.1 Overview

CNN-based object detectors perform well on detecting horizontal objects but struggle on oriented ones, in particular for long and dense oriented objects. Direct adaption using rotated bounding box B_r regression tends to produce inaccurate results due to high sensitivity to angle prediction error. Regressing the four vertices of quadrangle representation does not suffer from this problem, but also fails on some cases because of the ambiguity in defining the order of four ground truth vertices to be regressed. We attempt to solve the general multi-oriented object detection by introducing a simple representation for oriented objects and a novel detection scheme that divides and conquers nearly horizontal and oriented object detection, respectively. Specifically, we propose to glide the vertex of horizontal bounding box B_h on each corresponding side to accurately describe an oriented object. Put it simply, in addition to B_h, we compute four length ratios that characterize the relative gliding offset on each side of B_h. Besides, We also introduce an obliquity factor based on area ratio between multi-oriented object and its horizontal bounding box B_h. Based on the estimated obliquity factor, we select the horizontal (*resp.* oriented) detection for a nearly horizontal (*resp.* oriented) object. This simple yet effective framework only introduces five target variables compared with classical horizontal object detectors, requiring ignorable extra computation time.

3.2 Multi-Oriented Object Representation

The proposed method relies on a simple representation for oriented objects and an effective selection scheme. An intuitive illustration of the proposed representation is depicted in Fig. 2. For a given oriented object O (blue box in Fig. 2) and its corresponding horizontal bounding box B_h (black box in Fig. 2), let $v_i, i \in \{1, 2, 3, 4\}$ denote top, right, bottom, left intersecting point with its horizontal bounding box B_h denoted by $v_i', i \in \{1, 2, 3, 4\}$, respectively. The horizontal bounding box B_h is also usually represented by

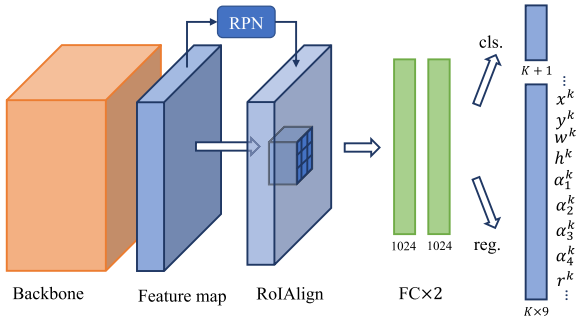

Fig. 3. Network architecture. We simply add five extra target variables (normalized to $[0, 1]$ using the sigmoid funciton) to the head of faster R-CNN [1]. K: number of classes; k: a certain class.

(x, y, w, h), where (x, y) is the center, and w and h are the width and height, respectively. We propose to represent the underlying oriented object by $(x, y, w, h, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$. The extra variables $\alpha_i, i \in \{1, 2, 3, 4\}$ are defined as follows:

$$\begin{aligned} \alpha_{\{1,3\}} &= \|s_{\{1,3\}}\|/w, \\ \alpha_{\{2,4\}} &= \|s_{\{2,4\}}\|/h, \end{aligned} \tag{1}$$

where $\|s_i\| = \|v_i - v_i'\|$ denotes the distance between v_i and v_i', i.e., the length of segment $s_i = (v_i, v_i')$ representing the gliding offset from v_i' to v_i. It is noteworthy that all α_i is set to 1 for horizontal objects.

In addition to the simple representation in terms of $(x, y, w, h, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$ for an oriented object O, we also introduce an obliquity factor characterizing the tilt degree of O. This is given by the area ratio r between O and B_h:

$$r = |O| \ / \ |B_h|, \tag{2}$$

where $|\cdot|$ denotes the cardinality. Nearly horizontal objects have a large obliquity factor r being close to 1, and the obliquity factor r for extremely slender and oriented objects are close to 0. Therefore, we can select the horizontal or oriented detection as the final result based on such obliquity factor r. Indeed, it is reasonable to represent nearly horizontal objects with horizontal bounding boxes. However, oriented detections are required to accurately describe oriented objects.

3.3 Network Architecture

The network architecture (see Fig. 3) is almost the same as faster R-CNN [1]. We simply add five extra target variables (normalized to $[0, 1]$ using the sigmoid funciton) to the head of faster R-CNN [1]. Specifically, The input image is first fed into a backbone network to extract deep features and generate bounding box proposals with RPN [1]. Then the regional features extracted via RoIAlign [35] on proposals are passed through a modified R-CNN head to generate final results, including a horizontal bounding box (x, y, w, h), four variables $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ characterizing the oriented bounding box, and obliquity factor r that indicates whether the object is nearly horizontal or not.

3.4 Ground-Truth Generation

The ground-truth for each object is composed of three components: classical horizontal bounding box representation $(\tilde{x}, \tilde{y}, \tilde{w}, \tilde{h})$, four extra variables $(\tilde{\alpha}_1, \tilde{\alpha}_2, \tilde{\alpha}_3, \tilde{\alpha}_4)$ representing the oriented object, and the obliquity factor $\tilde{r}$. The horizontal bounding box ground-truth follows the pioneer work in [13], which is relative to the proposal. The ground-truth for the four extra variables $(\tilde{\alpha}_1, \tilde{\alpha}_2, \tilde{\alpha}_3, \tilde{\alpha}_4)$ and obliquity factor $\tilde{r}$ depend only on the underlying ground-truth object, and are directly calculated by Eqs. (1) and (2), respectively.

3.5 Training Objective

The proposed method involves loss for RPN stage and R-CNN stage. The loss of RPN is the same as that in [1]. The loss L for R-CNN head contains a classification loss term L_{cls} and a regression loss term L_{reg}. The R-CNN loss L is given by

$$L = \frac{1}{N_{cls}} \sum_i L_{cls} + \frac{1}{N_{reg}} \sum_i p_i^* \times L_{reg}, \tag{3}$$

where N_{cls} and N_{reg} are the number of total proposals and positive proposals in a mini-batch fed into the head, respectively, and i denotes the index of a proposal in a mini-batch. If the ith proposal is positive, p_i^* is 1, otherwise it is 0. The regression loss L_{reg} contains three terms for horizontal bounding box, four length ratios $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$, and obliquity factor r regression, respectively. Put it simply, the regression loss L_{reg} is given by

$$\begin{aligned} L_{reg} &= \lambda_1 \times L_h + \lambda_2 \times L_\alpha + \lambda_3 \times L_r, \\ L_\alpha &= \sum_{i=1}^{4} smooth_{L_1}(\alpha_i - \tilde{\alpha}_i), \\ L_r &= smooth_{L_1}(r - \tilde{r}), \end{aligned} \tag{4}$$

where L_h is the loss for horizontal box regression, which is the same as that in [1], and λ_1, λ_2, and λ_3 are hyper-parameters that balance the importance of each loss term.

3.6 Inference

During testing phase, for a given image, the forward pass generates a set of $(x, y, w, h, \alpha_1, \alpha_2, \alpha_3, \alpha_4, r)$ representing horizontal bounding boxes, four length ratios, and obliquity factors. For each candidate, if its obliquity factor r is larger than a threshold t_r, indicating that the underlying object is nearly horizontal, we select the horizontal bounding box (x, y, w, h) as the final detection. Otherwise, we select the oriented one given by $(x, y, w, h, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$. The non-maximum suppression (NMS) process is also performed. Specifically, we first adopt the efficient horizontal NMS (with 0.5 IoU threshold) to get rid of some candidate proposals, followed by an oriented NMS (with 0.1 IoU threshold) on the significantly reduced number of candidate proposals.

4 Experiments

4.1 Datasets and Evaluation Protocols

DOTA [23] is a large-scale and challenging dataset for object detection in aerial images with quadrangle annotations. It contains 2806 4000×4000 images and 188, 282 instances of 15 object categories: plane, baseball diamond (BD), bridge, ground field track (GTF), small vehicle (SV), large vehicle (LV), ship, tennis court (TC), basketball court (BC), storage tank (ST), soccer-ball field (SBF), roundabout (RA), harbor, swimming pool (SP) and helicopter (HC). The official evaluation protocol of DOTA in terms of mAP is used.

HRSC2016 [36] is dedicated for ship detection in aerial images, containing 1061 images annotated with rotated rectangles. We conduct experiments for the level-1 task which detects ship from backgrounds. The standard evaluation protocol of HRSC2016 in terms of mAP is used.

MSRA-TD500 [37] is proposed for detecting long and oriented texts. It contains 300 training and 200 test images annotated in terms of text lines. Since the training set is rather small, following other methods, we also use HUST-TR400 [38] during training. The standard evaluation protocol of MSRA-TD500 based on F-measure is used.

RCTW-17 [39] is also a long text detection dataset, consisting of 8034 training images and 4229 test images annotated with text lines. This dataset is very challenging due to very large text scale variances.

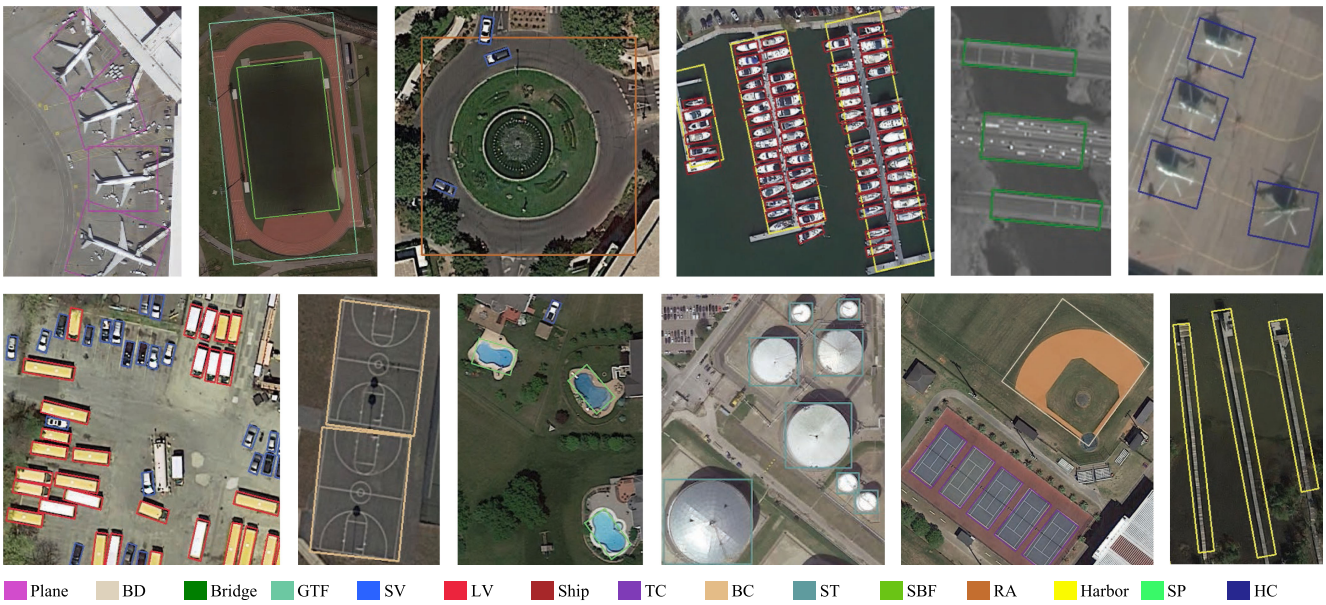

Fig. 4. Some detection results of the proposed method on DOTA [23]. The arbitrary-oriented objects are correctly detected.

We evaluate the proposed method via the online evaluation platform in terms of F-measure.

MW-18Mar [40] is a multi-target horizontal pedestrian tracking dataset, in which images are taken with fisheye cameras. The authors of [34] extracted some frames and annotated the pedestrians with rotated rectangles for omnidirectional pedestrian detection. The standard miss rates at every false positive per image (FPPI) and log average miss rates (LAMRs) [41] are adopted for benchmarking.

4.2 Implementation Details

The proposed method is implemented based on the project of "maskrcnn_benchmark"[1] using 3 Titan Xp GPUs. For a fair comparison with other methods, we adopt ResNet101 [42] for object detection in aerial images, where the batch size is set to 6 due to limited GPU memory. For the other experiments, ResNet50 is adopted, and the batch size is set to 12. In all experiments, the network is trained by SGD optimizer with momentum and weight decay set to 0.9 and 5×10^{-4}, respectively. The learning rate is initialized with 7.5×10^{-3} and divided by 10 at each learning rate decay step. The hyper-parameters λ_1, λ_2, and λ_3 in Eq. (4) are set to 1, 1, and 16, respectively. Without explicitly specifying, the hyper-parameter t_r on obliquity factor guiding the selection of horizontal or oriented detection is set to 0.8. Some other application related settings are depicted in the corresponding sections.

We compare the proposed method with two baseline methods using rotated bounding box representation (denoted by RBox Reg.) and quadrangle representation (denoted by Vertex Reg.). For the RBox reg., based on horizontal prior boxes, similar with [5], [24], [30], we regress the object center (x, y), long and short side length (w', h'), and the angle θ between the long side and X-axis. For Vertex Reg., we follow [6] by regressing the one-to-one vertex offset between each vertex of the prior box and its corresponding ground-truth vertex, which is ordered by minimizing the sum of vertex-wise euclidean distances between the ground-truth oriented object and its horizontal bounding box. For a fair comparison, both baseline methods are implemented using similar settings with the proposed method.

1. https://github.com/facebookresearch/maskrcnn-benchmark

4.3 Object Detection in Aerial Images

For the experiments on DOTA [23], we train the model for $50k$ steps, and the learning rate decays at $\{38k, 46k\}$ steps. Random rotation with angle among $\{0, \pi/2, \pi, 3\pi/2\}$ and class balancing are adopted for data augmentation. For the experiments on HRSC2016 [36], we train the model for $3.2k$ steps and decay the learning rate at $2.8k$ steps. Horizontal flipping is applied for data augmentation. For a fair comparison, the size of training/test images and the anchor settings on both datasets are kept the same as [5].

Overall Results. Some qualitative results on DOTA and HRSC2016 are shown in Figs. 4 and 7a, respectively. We show all detected objects with classification scores above 0.6. As illustrated, the proposed method accurately detects both horizontal and oriented objects even under dense distribution and/or being long. The quantitative comparisons with other methods on DOTA [23] and HRSC2016 [36] are depicted in Tables 1 and 2, respectively. Without any extra network design such as cascade refinement and attention mechanism, the proposed method outperforms some state-of-the-art methods on both DOTA and HRSC2016 and is more efficient in runtime. Specifically, For the experiment on DOTA, the proposed method without FPN [3] achieves 73.39 percent mAP, outperforming the state-of-the-art method [5] by 5.65 percent mAP. FPN [3] that exploits better multi-scale features is also beneficial for the proposed method, boosting the performance to 75.02 percent. The proposed method using FPN [3] improves the state-of-the-art method [27] by 3.86 percent mAP. For HRSC2016 dataset, the proposed method achieves 88.2 percent mAP, improving state-of-the-art methods by 2 percent.

Experiments on Different Network Architectures. To further demonstrate the versatility of the proposed method, we evaluate the proposed method on different networks. Concretely, we replace the faster R-CNN head by light-head R-CNN [43] head. As depicted in Table 1, using the same network on DOTA [23], the proposed method improves [5] by 4.49 and 4.75 percent mAP with and without FPN, respectively. The proposed method outperforms [5] by 1.2 percent mAP on HRSC2016 [36].

Ablation Study. We conduct ablation study on DOTA [23]. The proposed method relies on a novel multi-oriented object representation composed of three components: horizontal bounding box (x, y, w, h), gliding offsets $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$, and obliquity factor r. We begin with analyzing the quality of each individual component using Faster R-CNN head with FPN. First, the proposed method achieves a good performance with 76.22 percent mAP under horizontal bounding

TABLE 1
Quantitative Comparison With Other Methods on DOTA

Methods	FPN	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	mAP	FPS
FR-O [23]	-	79.42	77.13	17.70	64.05	35.30	38.02	37.16	89.41	69.64	59.28	50.30	52.91	47.89	47.40	46.30	54.13	-
RoI Trans.* [5]	-	88.53	77.91	37.63	74.08	66.53	62.97	66.57	90.50	79.46	76.75	**59.04**	56.73	62.54	61.29	55.56	67.74	5.9
Ours*	-	**89.95**	**86.37**	45.79	73.44	**71.44**	68.20	75.96	**90.72**	**79.63**	**85.03**	58.56	**70.19**	68.28	**71.34**	54.45	72.49	8.4
Ours-r	-	89.93	85.78	45.90	73.66	70.07	69.10	76.78	90.62	79.08	83.94	57.75	67.57	67.53	70.85	56.46	72.33	**9.8**
Ours	-	89.89	85.99	**46.09**	**78.48**	70.32	**69.44**	**76.93**	90.71	79.36	83.80	57.79	68.35	**72.90**	71.03	**59.78**	**73.39**	**9.8**
Azimi *et al.* [26]	✓	81.36	74.30	47.70	70.32	64.89	67.82	69.98	90.76	79.06	78.20	53.64	62.90	67.02	64.17	50.23	68.16	-
RoI Trans.* [5]	✓	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	-
CADNet [28]	✓	87.80	82.40	49.40	73.50	71.10	63.50	76.60	**90.90**	79.20	73.30	48.40	60.90	62.00	67.00	**62.20**	69.90	-
R^2CNN++ [27]	✓	89.66	81.22	45.50	75.10	68.27	60.17	66.83	**90.90**	80.69	86.15	**64.05**	63.48	65.34	68.01	62.05	71.16	-
RBox reg.	✓	89.37	75.96	35.43	69.57	68.35	63.78	74.92	90.76	**84.70**	85.26	62.43	62.40	52.97	60.32	54.61	68.72	9.2
Vertex reg.	✓	80.16	76.77	43.31	69.38	55.71	56.52	72.25	88.10	28.95	86.31	63.66	62.23	61.62	68.18	41.65	63.65	9.8
Ours*	✓	**90.02**	84.41	49.80	**77.93**	72.23	72.52	85.81	90.85	79.21	86.61	59.01	69.15	66.30	71.22	55.67	74.05	7.1
Ours-r	✓	89.40	**85.08**	52.00	77.40	72.68	72.89	86.41	90.74	78.80	86.79	57.84	70.42	67.73	**71.64**	56.63	74.43	**10.0**
Ours	✓	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**	**10.0**
RBox reg.†	✓	42.52	21.76	10.47	36.53	26.57	26.91	32.39	63.20	36.56	33.54	33.04	15.63	11.16	10.05	12.98	27.56	9.2
Vertex reg.†	✓	67.94	50.51	14.28	47.46	29.79	27.92	40.66	72.75	14.29	67.59	33.47	40.87	22.04	17.91	15.13	37.51	9.8
Ours*†	✓	**77.98**	53.21	12.52	**68.87**	47.25	46.07	54.83	**90.45**	68.00	68.45	**56.44**	40.12	28.59	22.47	19.13	50.29	7.1
Ours-r†	✓	67.66	50.37	**17.07**	60.60	48.74	49.00	61.59	88.98	68.84	**74.83**	48.30	48.03	32.58	23.78	23.75	50.94	**10.0**
Ours†	✓	77.32	**59.75**	15.95	67.63	**50.02**	**50.25**	**63.62**	90.38	**69.04**	74.56	51.58	**50.16**	**32.73**	**24.19**	**25.18**	**53.49**	**10.0**

*Ours-r means that the divide and conquer detection scheme based on obliquity factor r is not used. * indicates that the backbone network is light-head R-CNN [43]. † stands for evaluation using IoU threshold 0.7. Note that the runtime for oriented NMS is not included for all methods on this dataset. Otherwise, the proposed method using FPN runs at 9.4 FPS instead of 10.0 FPS.*

box evaluation. The small performance gap (i.e., 1.2 percent mAP) between oriented and horizontal object detection implies that the gliding offset regression is also quite accurate. We also explicitly evaluate the accuracy of gliding offset regression in terms of mean absolute error (MAE) for the correctly detected objects. As depicted in Fig. 5, the gliding offset regression is quite accurate for oriented objects, but is less precise for nearly horizontal objects (e.g., $\tilde{r} > 0.8$) for which potential confusion issue remains. This motivates us to regress the obliquity factor r to guide the selection of horizontal or oriented detection as the final detection result, helping to remedy the remaining confusion issue for nearly horizontal objects. Indeed, as shown in Fig. 5, the obliquity factor r regression is in general very accurate (MAE $< 5.3\%$). This quality analysis of each individual component of the proposed multi-oriented object representation confirms the effectiveness of the proposed method.

Some qualitative comparison can be found in Fig. 6. We rotate an image with several different angles and test the proposed method and two baseline methods on the rotated images. The RBox reg. produces inaccurate results due to the imprecise angle regression. The Vertex reg. have difficulty for tilted objects at some orientations due to the confusion in defining the vertex order in training. The proposed method is able to accurately detect objects of any orientations.

The quantitative comparison with baseline methods is depicted in the middle of Table 1. The proposed method outperforms the two baseline methods by a large margin. Specifically, the proposed method outperforms the RBox reg. and Vertex reg. by 6.30 and 11.37 percent mAP at the cost of ignorable runtime. In fact, as depicted in Table 1, the proposed method is more efficient than both baseline methods producing more false detections. To further demonstrate the accuracy of the proposed method, we also conduct a benchmark using larger IoU threshold 0.7 in the evaluation system. As shown in Table 1, the improvement is even more significant, changing from 6.30 percent (*resp.* 11.37 percent) to 25.93 percent (*resp.* 15.98 percent). This further demonstrates the accuracy of the proposed method in detecting oriented objects.

We then assess the individual contribution of the proposed vertex gliding and divide-and-conquer detection scheme in the proposed method for multi-oriented object detection. To this end, we evaluate an alternative of the proposed method by discarding the divide-and-conquer detection scheme based on obliquity factor r. As depicted in Table 1, the proposed representation in terms of $(x, y, w, h, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$ contributes a lot to the improvement. The proposed detection scheme brings 0.59 and 1.06 percent mAP improvement with and without FPN [3], respectively. When larger IoU threshold 0.7 is used, the selection scheme yields 2.55 percent mAP improvement, confirming the effectiveness of the selection scheme based on obliquity factor r. Without the selection scheme, some nearly horizontal objects with inaccurate predicted gliding offsets (see Fig. 5) may be considered as correct (*resp.* incorrect) detection under evaluation with 0.5 (*resp.* 0.7) IoU threshold. This explains the more significant improvement of the selection scheme when a larger IoU threshold is used for evaluation.

TABLE 2
Quantitative Comparison With Some State-of-the-Art Methods on HRSC2016

Methods	RC2 [44]	R^2PN [25]	RRD [32]	RoI Trans.* [5]	Ours*	Ours
mAP	75.7	79.6	84.3	86.2	87.4	**88.2**

** indicates that Light-head R-CNN is adopted.*

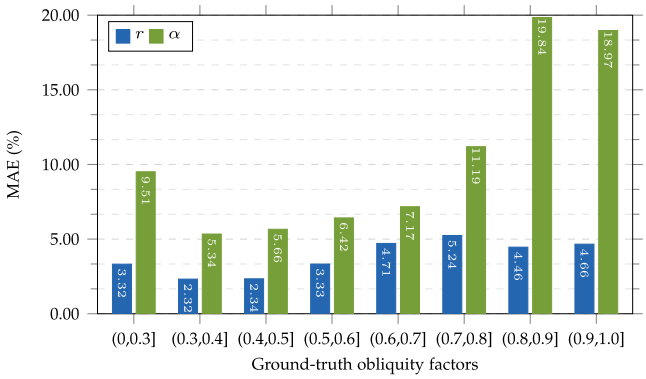

Fig. 5. Mean absolute error (MAE) of obliquity factor r and gliding offset α regression with respect to different ranges of ground-truth obliquity factors for the proposed method on DOTA.

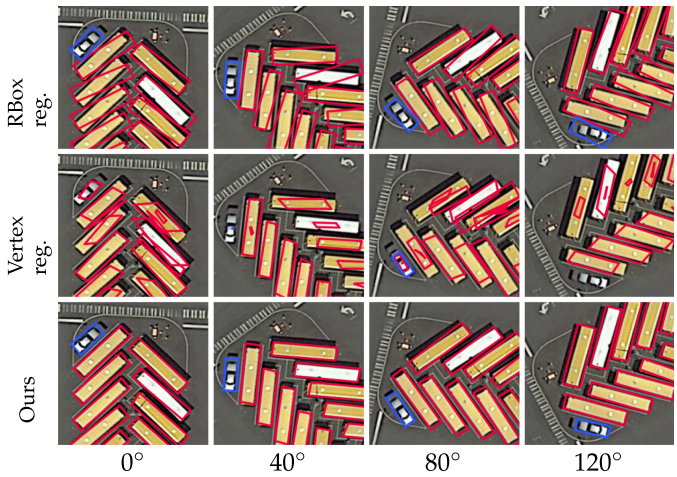

Fig. 6. Qualitative comparison with baseline methods in detecting objects of different orientations (by rotating an input image with different angles). The meaning of colors is the same as that in Fig. 4.

TABLE 3
Ablation Study on Different Thresholds t_r of Obliquity Factor r

t_r	0.65	0.70	0.75	0.80	0.85	0.90	0.95
w FPN	73.29	74.30	74.72	75.02	**75.06**	**75.06**	74.44
w/o FPN	71.76	72.42	73.24	**73.39**	73.37	72.59	72.47

We also analyze the effect of different thresholds t_r of obliquity factor r on DOTA dataset using Faster R-CNN head with FPN. As depicted in Table 3, the performance is rather stable, especially for $t_r \in [0.75, 0.85]$. The performance slightly decreases for smaller and larger t_r. Indeed, with a very small threshold t_r, horizontal bounding boxes are selected to represent some oriented objects, which leads to inaccurate detection. When a large threshold t_r is adopted, the potential confusion issue for nearly horizontal objects remains, also resulting in decreased performance.

4.4 Long Text Detection in Natural Scenes

For oriented scene text detection on MSRA-TD500 [37] and RCTW-17 [39], we apply the same data augmentation as SSD [18]. Besides, we also randomly rotate the images with $\pi/2$ to better handle vertical texts. The training images are randomly cropped and resized to some specific sizes. For MSRA-TD500, we randomly resize the short side of cropped images to $\{512, 768, 864\}$. For RCTW-17 [39] containing many small texts, the short side is randomly resized to $\{960, 1200, 1400\}$. We first pre-train the model on SynthText [45] for one epoch. Then we fine-tune the model for $4k$ (*resp.* $14k$) and decay the learning rate at $3k$ (*resp.* $10k$) steps for MSRA-TD500 (*resp.* RCTW-17). During test, the short side of MSRA-TD500 images is resized to 768. For RCTW-17, the short side is set to 1200 for single scale test. We add extra scales of $\{512, 1024, 1280, 1560\}$ for multi-scale test.

TABLE 4
Quantitative Comparison With Other Methods on MSRA-TD500 [37]

Methods	Precision	Recall	F-measure	FPS
Zhang *et al.* [12]	83.0	67.0	74.0	0.5
SegLink [9]	86.0	70.0	77.0	8.9
RRD [32]	87.0	73.0	79.0	10.0
EAST [4]	87.3	67.4	76.1	13.2
Border MS [47]	83.0	73.3	76.8	-
TextField [29]	87.4	75.9	81.3	5.2
Lyu *et al.* [10]	87.6	76.2	81.5	5.7
CRAFT [48]	88.2	78.2	82.9	8.6
MCN [11]	88.0	79.0	83.0	-
Wang *et al.* [46]	85.2	82.1	83.6	10.0
Direct MS [7]	**91.0**	81.0	86.0	-
Ours	88.8	**84.3**	**86.5**	**15.0**

MS stands for multi-scale test.

TABLE 5
Quantitative Comparison With Other Methods on RCTW-17 [39]

Methods	Precision	Recall	F-measure	FPS
Official baseline [39]	76.0	40.4	52.8	8.9
RRD [32]	72.4	45.3	55.7	10.0
RRD MS	77.5	59.1	67.0	-
Direct MS [7]	76.7	57.9	66.0	-
Border MS [47]	78.2	58.8	67.1	-
LOMO [8]	**80.4**	50.8	62.3	4.4
LOMO MS	79.1	60.2	68.4	-
Ours	77.0	61.0	68.1	7.8
Ours MS	77.6	**62.7**	**69.3**	-

MS stands for multi-scale test.

Some qualitative illustrations are given in Fig. 7b, 7c, 7d, and 7e. The proposed method correctly detect texts of arbitrary orientations. The quantitative comparisons with some state-of-the-art methods on MSRA-TD500 and RCTW-17 are depicted in Tables 4 and 5, respectively. The proposed method outperforms other competing methods and is more efficient on both datasets. Specifically, on MSRA-TD500, the proposed method under single scale test outperforms the multi-scale version of [7] using larger extra training images by 0.5 percent, and improves [46] by 2.9 percent. On RCTW-17, the proposed method outperforms the state-of-the-art method [8] by 5.8 percent (*resp.* 0.9 percent) under single-scale (*resp.* multi-scale) test while being much more efficient.

4.5 Pedestrian Detection in Fisheye Images

We compare the proposed method with the two baseline methods RBox reg. and Vertex reg., classical horizontal box regression (denoted by HBox reg.), and the method in [34] on MW-18Mar [40].

Fig. 7. Some detection results of the proposed method on HRSC2016 [36] in (a), MSRA-TD500 [37] in (b-c), and RCTW-17 [39] in (d-e).

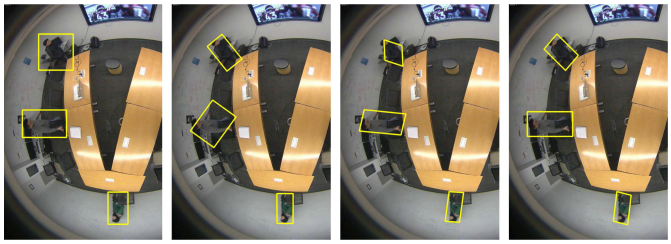

Fig. 8. Qualitative illustrations of different methods on MW-18Mar [40].

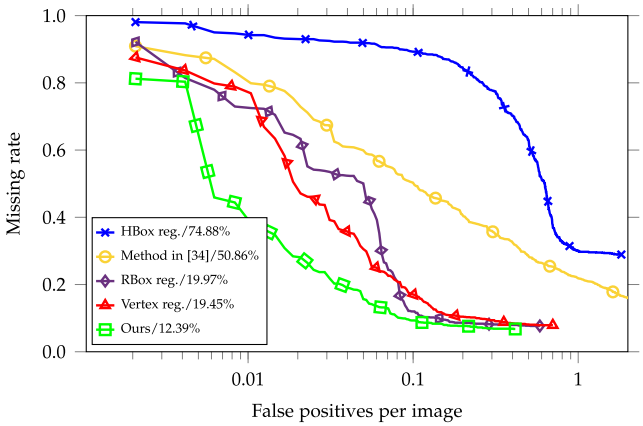

Fig. 9. Evaluation on MW-18Mar [40]. The numbers are the LAMRs.

For a fair comparison with [34], we follow similar training and test settings with [34]. Specifically, in all experiments, FPN is not used. All images are resized to 416×416 during training and test. During training, We randomly rotate the images for data augmentation. The model is trained in total for $4k$ steps and the learning rate decays at $3k$ steps.

Some qualitative results are illustrated in Fig. 8. The proposed method achieves more accurate results than all the baseline methods. The curve of missing rate with respect to the number of false positives per image is depicted in Fig. 9. The proposed method achieves lower missing rate than all the other methods.

5 Conclusion

In this paper, we propose a simple yet effective representation for oriented objects and a divide-and-conquer strategy to detect multi-oriented objects. Based on this, we build a robust and fast multi-oriented object detector. It accurately detects ubiquitous multi-oriented objects such as objects in arial images, scene texts, and pedestrians in fisheye images. Extensive experiments demonstrate that the proposed method outperforms some state-of-the-art methods on multiple benchmarks while being more efficient. In the future, we would like to explore the complementary of the proposed method with other approaches focusing on feature enhancement. One-stage multi-oriented object detector is also another direction which is worthy of exploitation.

Acknowledgments

This work was supported in part by the Major Project for New Generation of AI under Grant no. 2018AAA0100400, in part by NSFC 61703171, and in part by NSF of Hubei Province of China under Grant 2018CFB199, Dr. Yongchao Xu was supported by the Young Elite Scientists Sponsorship Program by CAST and Dr. Xiang Bai was supported by the Program for HUST Academic Frontier Youth Team 2017QYTD08.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[2] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *preprint: 1804.02767*.
[3] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.
[4] X. Zhou *et al.*, "EAST: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2642–2651.
[5] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning roi transformer for oriented object detection in aerial images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2849–2858.
[6] M. Liao, B. Shi, and X. Bai, "Textboxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, Aug. 2018.
[7] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Multi-oriented and multi-lingual scene text detection with direct regression," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5406–5419, Nov. 2018.
[8] C. Zhang *et al.*, "Look more than once: An accurate detector for text of arbitrary shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10 552–10 561.
[9] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3482–3490.
[10] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7553–7563.
[11] Z. Liu, G. Lin, S. Yang, J. Feng, W. Lin, and W. L. Goh, "Learning markov clustering networks for scene text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6936–6944.
[12] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4159–4167.
[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.
[14] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.
[15] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Advances Neural Inf. Process. Syst.*, 2016, pp. 379–387.
[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.
[17] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7263–7271.
[18] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.
[20] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 734–750.
[21] X. Zhou, J. Zhuo, and P. Krahenbuhl, "Bottom-up object detection by grouping extreme and center points," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 850–859.
[22] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, 2019, pp. 6569–6578.
[23] G.-S. Xia *et al.*, "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3974–3983.
[24] L. Liu, Z. Pan, and B. Lei, "Learning a rotation invariant detector with rotatable bounding box," 2017, *arxiv: 1711.09405*.
[25] Z. Zhang, W. Guo, S. Zhu, and W. Yu, "Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks," *IEEE Geosci. Remote Sens. Lett.*, vol. vol. 15, no. 11, pp. 1745–1749, Nov. 2018.
[26] S. M. Azimi, E. Vig, R. Bahmanyar, M. Körner, and P. Reinartz, "Towards multi-class object detection in unconstrained remote sensing imagery," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 150–165.
[27] X. Yang *et al.*, "R2CNN++: Multi-dimensional attention based rotation invariant detector with robust anchor strategy," 2018, *arxiv: 1811.07126*.
[28] G. Zhang, S. Lu, and W. Zhang, "CAD-Net: A context-aware detection net-570 work for objects in remote sensing imagery," *IEEE Trans. Geosci. Remote Sensing*, vol. 57, no. 12, pp. 10015–10024, 2019.
[29] 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 Trans. Image Process.*, vol. 28, no. 11, pp. 5566–5579, Nov. 2019.
[30] J. Ma *et al.*, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, Nov. 2018.
[31] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3454–3461.

[32] M. Liao, Z. Zhu, B. Shi, G. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5909–5918.
[33] R. Seidel, A. Apitzsch, and G. Hirtz, "Omnidetector: With neural networks to bounding boxes," 2018, *arxiv: 1805.08503*.
[34] M. Tamura, S. Horiguchi, and T. Murakami, "Omnidirectional pedestrian detection by rotation invariant training," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2019, pp. 1989–1998.
[35] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.
[36] Z. Liu, H. Wang, L. Weng, and Y. Yang, "Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 8, pp. 1074–1078, Aug. 2016.
[37] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1083–1090.
[38] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4737–4749, Nov. 2014.
[39] B. Shi *et al.*, "ICDAR2017 competition on reading chinese text in the wild (RCTW-17)," in *Proc. Int. Conf. Document Anal. Recognit.*, 2017, vol. 1, pp. 1429–1434.
[40] Mirror worlds challenge, 2017. [Online]. Available: https://icat.vt.edu/mirrorworlds/challenge/index.html
[41] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Apr. 2012.
[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[43] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Light-Head R-CNN: In defense of two-stage object detector," 2017, *arxiv: 1711.07264*.
[44] Z. Liu, J. Hu, L. Weng, and Y. Yang, "Rotated region based CNN for ship detection," in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 900–904.
[45] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2315–2324.
[46] X. Wang, Y. Jiang, Z. Luo, C.-L. Liu, H. Choi, and S. Kim, "Arbitrary shape scene text detection with adaptive text region representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6449–6458.
[47] C. Xue, S. Lu, and F. Zhan, "Accurate scene text detection through border semantics awareness and bootstrapping," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 355–372.
[48] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9365–9374.