Beyond Bounding-Box: Convex-hull Feature Adaptation for Oriented and Densely Packed Object Detection

Zonghao Guo[1], Chang Liu[1], Xiaosong Zhang[1], Jianbin Jiao[1], Xiangyang Ji[2], and Qixiang Ye[1]*
[1]University of Chinese Academy of Sciences, Beijing, China. [2]Tsinghua University, Beijing, China
{guozonghao19,liuchang615,zhangxiaosong18}@mails.ucas.ac.cn
xyji@tsinghua.edu.cn, {jiaojb,qxye}@ucas.ac.cn

Abstract

Detecting oriented and densely packed objects remains challenging for spatial feature aliasing caused by the intersection of reception fields between objects. In this paper, we propose a convex-hull feature adaptation (CFA) approach for configuring convolutional features in accordance with oriented and densely packed object layouts. CFA is rooted in convex-hull feature representation, which defines a set of dynamically predicted feature points guided by the convex intersection over union (CIoU) to bound the extent of objects. CFA pursues optimal feature assignment by constructing convex-hull sets and dynamically splitting positive or negative convex-hulls. By simultaneously considering overlapping convex-hulls and objects and penalizing convex-hulls shared by multiple objects, CFA alleviates spatial feature aliasing towards optimal feature adaptation. Experiments on DOTA and SKU110K-R datasets show that CFA significantly outperforms the baseline approach, achieving new state-of-the-art detection performance. Code is available at github.com/SDL-GuoZonghao/BeyondBoundingBox.

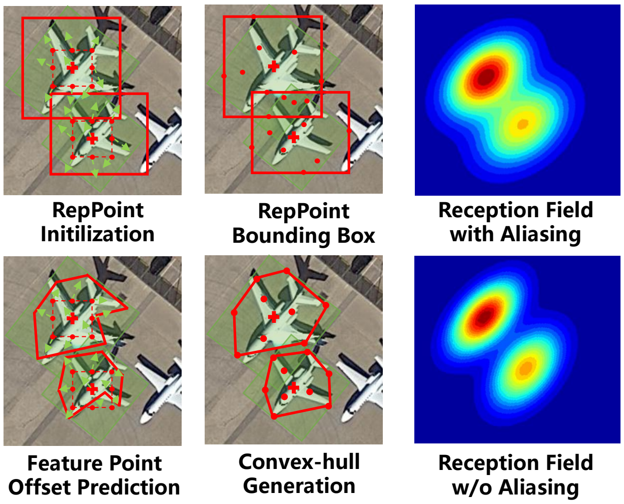
RepPoint Initilization
RepPoint Bounding Box
Reception Field with Aliasing
Feature Point Offset Prediction
Convex-hull Generation
Reception Field w/o Aliasing

Figure 1. Problem illustration. (Upper) When using box representation, oriented and densely packed objects cause feature aliasing for the intersection of reception fields between objects. (Lower) With convex-hull representation, our CFA approach adapts features located on regular convolutional maps to oriented and densely packed objects, solving the feature aliasing issue in a systematic way.

1. Introduction

Over the past decade, we witnessed substantial progress in visual object detection. This is attributed to the availability of deep networks incorporating rich feature representation [16, 15] and large-scale databases [3, 17, 28] for pretraining representative models. However, most detectors encounter problems when objects, such as those in aerial images, are in arbitrary orientations [24], or have dissimilar layouts to the objects utilized during training. The problems become more serious when oriented objects are densely distributed, because this causes spatial feature aliasing at the intersection of reception fields, Fig. 1(above).

One solution for oriented object detection is feature/anchor augmentation [19, 21, 20, 31], which produces features in multiple orientations for detector training. This intuitive solution, however, suffers a significant increase in computational complexity and false detection. The other solution is defining RoI transformers, which apply spatial transformations on RoIs while learning the parameters under the supervision of oriented bounding boxes [4]. Transformers have been promoted as being dynamic [24], attentive [37], and smoothed [36], enabling adaptive receptive fields to adapt to object orientations. However, the problem about how to adapt feature grids to objects of arbitrary layouts remains unsolved, which causes feature aliasing, particularly when objects are densely packed.

In this paper, we propose a convex-hull feature adapta-

*Corresponding Author.

tion (CFA) approach for oriented and densely packed object detection. The objective is to adapt features located within regular convolutional grids to objects with irregular layouts. We model object layouts as convex-hulls, which have natural advantages over rectangular boxes when required to cover the full extent of objects while minimizing background regions, Fig. 1(below). On each convex-hull is a set of feature points (extreme points) which defines the object boundaries and indicates the Convex Intersection over Union (CIoU) to determine object localization. Within the convex-hull, discriminative features represent the object appearance for precise classification.

The proposed detector consists of two stages, following the RepPoint method [38]. Initially, convex-hulls are generated by predicting feature point offsets driven by object localization (CIoU) loss. Then, convex-hulls are refined to cover the full extent of objects while classifying objects with the backgrounds driven by object localization and classification loss. Meanwhile, CFA constructs a convex-hull set for each object so that features at the periphery of objects can be jointly optimized. To adapt convex-hulls to objects, CFA defines a convex-hull set splitting strategy, under the guidance of gradient consistency. By dynamically categorizing convex-hulls into positives or negatives and penalizing convex-hulls shared by neighboring objects, CFA alleviates feature aliasing and pursues optimal feature adaptation. During the inference phase, single convex-hulls are used to localize objects without convex-hull set construction or set splitting, which guarantees detection is efficient.

The contributions of this study include:

- We propose convex-hull representation, which is promising for detecting objects of irregular shapes and/or layouts via learnable feature configuration.

- We propose the convex-hull feature adaptation (CFA) approach, which incorporates CIoU and feature anti-aliasing strategies and defines a systematic way to detect oriented and densely packed objects.

- We significantly improved the performance upon the baseline method, achieving new state-of-the-art on commonly used benchmarks.

2. Related Works

Orientation-encoded Representation. Orientation-robust representation received attention in the era of hand-crafted features. For example, the SIFT features [22], Gabor features [8], and LBP [23] used orientation-encoded feature channels or bit cyclic shift to achieve rotation in-variance. In recent years, orientation-robust representation has been fused with deep feature learning. Spatial transformer network (STN) [9] contributed a general framework for spatial transform by introducing a network module which manipulates the feature maps according to the estimated transform matrix. ORN [44] involved active rotating filters which actively change orientations during convolution and produce feature maps with explicitly encoded locations and orientations.

The majority of existing methods focus on locally or globally invariant features, which enhances orientation-encoded representation. However, the problem of how to process oriented object layouts remains.

Orientation-robust Detection. To precisely localize objects in arbitrary orientations and/or with dense distributions [24], early methods [19, 21, 20, 20, 31] used feature/anchor augmentation strategies, *e.g.*, numerous features/anchors/RoIs [21] with multiple orientations, scales, and aspect ratios for object representation. Despite the effectiveness, these methods suffer from substantial increases in computational complexity and risk of false detection caused by additional classification operations. Recent methods defined RoI transformers to configure oriented boxes for oriented objects. RoI-Transformers [4] applied spatial transformations on RoIs and learned parameters under the supervision of oriented bounding box (OBB) annotations. Dynamic Refinement Network (DRN) [24] equipped with a feature selection module (FSM) and a dynamic refinement head (DRH) can adapt receptive fields to objects with various orientations.

R3Det [33] proposed re-encoding the positional information of bounding boxes into the corresponding feature points through feature interpolation. CSL [35] involved a new rotation prediction pipeline, which converts orientation prediction from a regression problem to a classification task. GlidingVertex [32] glided the vertex of a horizontal bounding box on each corresponding side to accurately describe multi-oriented objects. While SCRDet [37] introduced the localization (IoU) loss term to address the boundary issue for oriented bounding boxes.

Existing methods have made encouraging progress in rotation-robust detection; however, the problem of how to adapt features on regular grids to objects of arbitrary layouts remains. We attempt to solve this problem by exploring a novel convex-hull representation beyond rectangular representation.

Feature Spatial Adaptation. Deformable convolutional network (DCN) [2] and ActivateConv [11] contributed general spatial adaptation models by defining learnable spatial transformation upon feature maps. CARAFE [30] defined content-aware re-assembly of features for dense prediction (detection or segmentation) tasks. This not only aggregated contextual information within a larger receptive field but also contributed instance-specific handling, which generates adaptive kernels on-the-fly.

Our approach defines a more feasible way by introducing

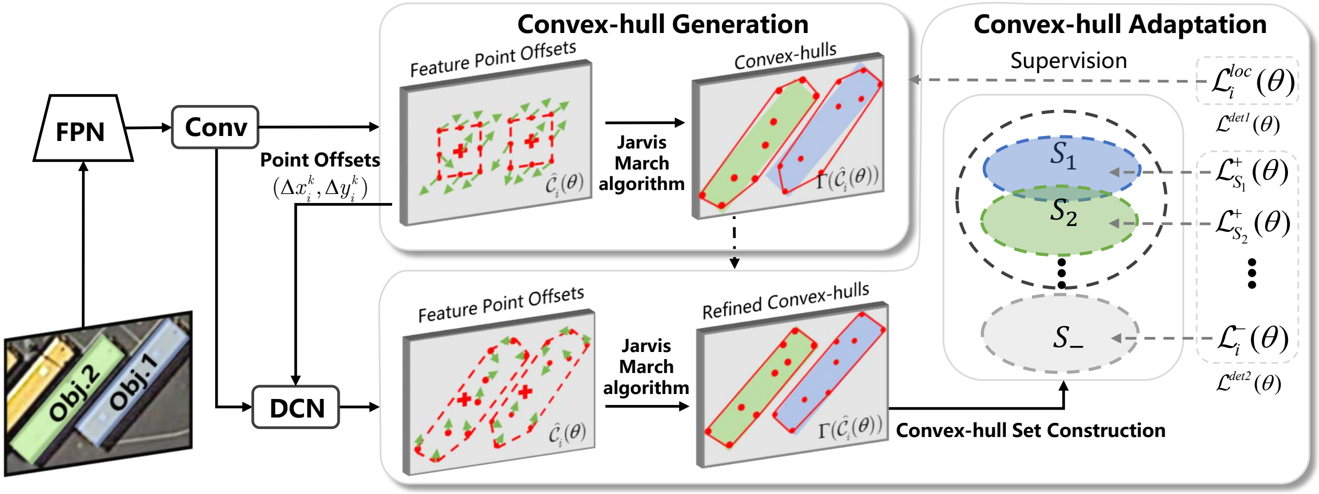
Convex-hull Generation
Feature Point Offsets
Convex-hulls
Jarvis March algorithm
FPN
Conv
Point Offsets
$(\Delta x_i^k, \Delta y_i^k)$
$\hat{C}_i(\theta)$
$\Gamma(\hat{C}_i(\theta))$
Feature Point Offsets
Refined Convex-hulls
Jarvis March algorithm
DCN
Obj.2
Obj.1
$\hat{C}_i(\theta)$
$\Gamma(\hat{C}_i(\theta))$
Convex-hull Adaptation
Supervision
$\mathcal{L}_i^{loc}(\theta)$
$\mathcal{L}^{det1}(\theta)$
S_1
S_2
S_-
$\mathcal{L}_{S_1}^{+}(\theta)$
$\mathcal{L}_{S_2}^{+}(\theta)$
$\mathcal{L}_i^{-}(\theta)$
$\mathcal{L}^{det2}(\theta)$
Convex-hull Set Construction

Figure 2. Flowchart of the proposed CFA detector. The first stage predicts feature point offsets for convex-hull generation, while the second stage refines the predicted convex-hulls, as well as reducing feature aliasing by convex-hull feature adaptation.

convex-hull sets. It is inspired by the FreeAnchor [40, 13], ATSS [39], and PAA detectors [14], which used learning-to-match [41] and continuation optimization [29] to adapt features to objects with various spatial layouts. Beyond these methods, our feature adaption strategy considers features shared with multiple bags/sets, making this the first attempt at handling joint feature optimization.

3. The Proposed Approach

3.1. Overview

The flowchart of the proposed CFA approach is provided in Fig. 2, which uses the RepPoint method [38] as the baseline. As an anchor-free detector, CFA consists of two stages: convex-hull generation and convex-hull adaptation. The first stage predicts convex-hulls for all locations on feature maps and estimates convex-hull layouts. The second stage refines predicted convex-hulls as well as adapting them to densely packed objects. During the inference phase, only convex-hull generation is carried out to localize objects without convex-hull set construction or set splitting, which guarantees the simplicity and efficiency of detection. For each predicted convex-hull, we calculate a minimum bounding rectangle and merge overlapping ones using Non-Maximum Suppression (NMS) detection performance evaluation. In what follows, we first analyze the disadvantages of box representation and present convex-hull representation. We then present convex-hull generation and convex-hull adaptation modules.

3.2. Convex-hull Representation

Considering that the CNN feature grids are axis-aligned, modern object detectors typically use boxes, *e.g.*, rectangles or rotated rectangles, to cover object extent. Recent anchor-free detectors, such as RepPoint [38] and ExtremePoint [43], endow the feature grid deformability, *i.e.*, dynamically arranging feature points in accordance with object bounding boxes. However, they did not consider adapting features to objects of non-rectangle layouts, Fig. 1, which causes feature aliasing when such objects come together. Recent studies including DRN [24] and R3Det [33] attacked this problem by predicting object orientations, but cannot adapt features to object extent well, Fig. 3(above). The drawbacks of existing methods attribute to using rectangle boxes to bound objects of irregular layouts, which degrades the representation capacity of convolutional features.

We propose to represent object extent as convex-hulls, each of which corresponds to a set of sampled points on convolutional feature maps. Driven by object localization and classification loss the detector learns to arrange feature points in a manner that bounds the extent of an object and discriminative features, Fig. 1. Modeling objects as convex-hulls, object extent can be covered more completely, while avoiding orientation ambiguity. Specifically, for i-th location (x_i, y_i) on the feature maps $X \in \mathbb{R}^{H\times W\times C}$, where H, W and C respectively denote the height, width, and channel number of feature maps, we define the convex-hull as

$$\mathcal{C}_i = \{(x_i^k, y_i^k)\}_i^{k=1...K}, \tag{1}$$

where k indexes feature points and $K = 9$ is the feature point number of the convex-hull. In experiments, the feature points are initialized as a 3×3 feature grid, Fig. 3.

During training, convex-hull feature points "move" toward the ground-truth bounding box to maximize the CIoU (defined in next section) between the convex-hull and the ground-truth box. This is a procedure to predict an offset $(\Delta x_i^k, \Delta y_i^k)$ for each feature point using a convolutional operation, which outputs a feature offset map

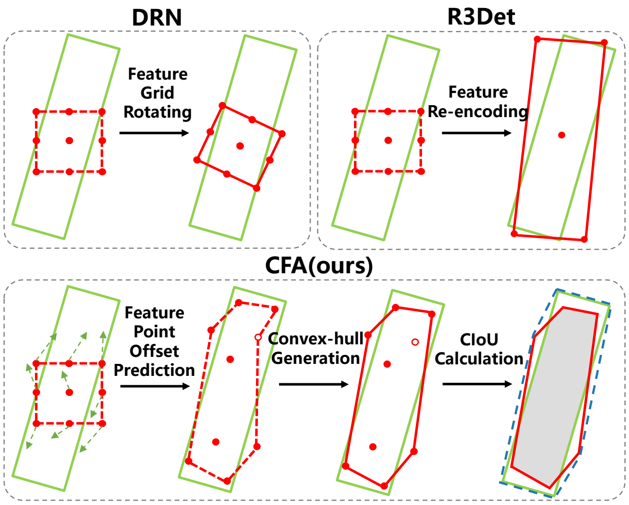

Figure 3. Comparison of oriented box representation (above) with the proposed convex-hull representation (below).

$(O \in \mathbb{R}^{H\times W\times 2K})$. The convex-hull prediction is formulated as

$$\hat{\mathcal{C}}_i(\theta) \leftarrow \{(x_i^k + \Delta x_i^k(\theta), y_i^k + \Delta y_i^k(\theta))\}_i^{k=1...K}, \qquad (2)$$

where θ denotes network parameters.

Considering that the update of feature points can destroy the convex-hull, Fig. 3(below), the Jarvis March algorithm [10], denoted as $\Gamma(\cdot)$, is applied on the sampled feature points to generate a minimal convex-hull after each training iteration, as

$$\mathcal{C}_i(\theta) = \Gamma(\hat{\mathcal{C}}_i(\theta)). \qquad (3)$$

Starting from a leftmost one of the feature points, the Jarvis March algorithm keeps the points in the convex-hull by anticlockwise rotation. From a current point, it selects the next point by checking the orientations of those points from the current point. When the angle is largest, the point is selected. After completing all points and when the next point becomes the start point, the algorithm stops.

3.3. Convex-hull Generation

CIoU. Based on each convex-hull prediction, we can calculate the localization and classification loss for an object. The CIoU between the i-th predicted convex-hull $\mathcal{C}_i(\theta)$ and the ground-truth box $\mathcal{B}_j$ of the j-object is calculated as

$$\mathrm{CIoU}(\mathcal{C}_i(\theta), \mathcal{B}_j) = \frac{|\mathcal{C}_i(\theta) \cap \mathcal{B}_j|}{|\mathcal{C}_i(\theta) \cup \mathcal{B}_j|} - \frac{|\mathcal{R}_j \setminus (\mathcal{C}_i(\theta) \cup \mathcal{B}_j)|}{|\mathcal{R}_j|}, \qquad (4)$$

where $\mathcal{R}_j$ is the minimum bounding polygon of $\mathcal{B}_j$ and $\mathcal{C}_i(\theta)$, Fig. 3. According to [27], CIoU not only can represent the spatial overlap (the shadow area in Fig. 3) between $\mathcal{C}_i(\theta)$ and $\mathcal{B}_j$ but also is continuous and derivable.

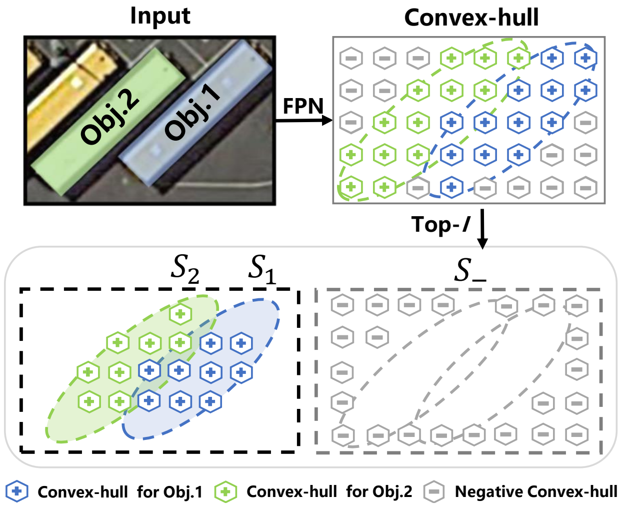

Figure 4. Convex-hull set construction.

Convex-hull Loss. According to Eq. 4, the CIoU loss is defined as

$$\mathcal{L}_i^{loc}(\theta) = 1 - \mathrm{CIoU}(\mathcal{C}_i(\theta), \mathcal{B}_j). \qquad (5)$$

Let $f_i^k(\theta)$ denote the feature of the k-th point, the convex-hull feature $f_i(\theta)$ is calculated by weighted summation of all the features of the points on $\mathcal{C}_i(\theta)$, as $f_i(\theta) = \sum_k w_i^k \cdot f_i^k(\theta)$, where w_i^k denotes learnable feature weights in DCN [2]. Based on the convex-hull feature, the prediction score $\mathcal{S}_i(\theta)$ of $\mathcal{C}_i(\theta)$ is calculated by a convolutional operation, and the classification loss of $\mathcal{C}_i(\theta)$ with respect to object $\mathcal{B}_j$ is defined as

$$\mathcal{L}_i^{cls}(\theta) = \mathrm{FL}(S_i(\theta), Y_j), \qquad (6)$$

where Y_j denotes the binary ground-truth label and $\mathrm{FL}(\cdot)$ the FocalLoss function [16]. As a result, the classification loss for a positive convex-hull (defined in the next section) is calculated as

$$\mathcal{L}_i^{+}(\theta) = \mathcal{L}_i^{cls}(\mathcal{S}_i(\theta), Y_j) + \lambda \mathcal{L}_i^{loc}(\mathcal{C}_i(\theta), \mathcal{B}_j), \qquad (7)$$

where λ is an experimentally determined regularization factor. For a negative convex-hull, classification loss is defined as

$$\mathcal{L}_i^{-}(\theta) = \mathcal{L}_i^{cls}(\mathcal{S}_i(\theta), Y_j). \qquad (8)$$

Optimization. As shown in Fig. 2, during detector training, convex-hulls are generated solely by optimizing the localization (CIoU) loss, as

$$\mathcal{L}^{det1}(\theta) = \frac{1}{J} \sum_i \mathbb{I}_{(x_i,y_i)} \mathcal{L}_i^{loc}(\theta), \qquad (9)$$

where J denotes the number of ground-truth objects, $\mathbb{I}_{(x_i,y_i)}$ an indicator function for whether i-th convex-hull is involved in optimization. The classification loss will be used in the second stage.

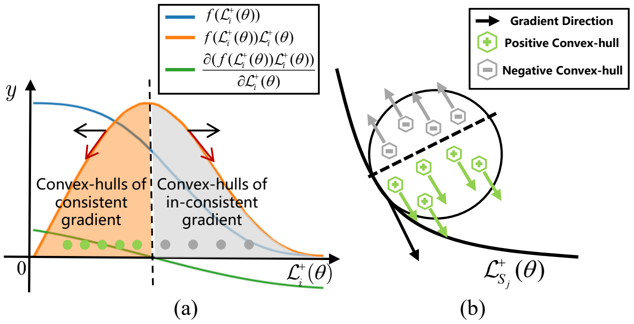
$f(\mathcal{L}_i^+(\theta))$
$f(\mathcal{L}_i^+(\theta))\mathcal{L}_i^+(\theta)$
$\frac{\partial(f(\mathcal{L}_i^+(\theta))\mathcal{L}_i^+(\theta))}{\partial\mathcal{L}_i^+(\theta)}$
y
Convex-hulls of consistent gradient
Convex-hulls of in-consistent gradient
0
$\mathcal{L}_i^+(\theta)$
(a)
Gradient Direction
Positive Convex-hull
Negative Convex-hull
$\mathcal{L}_{S_j}^+(\theta)$
(b)

Figure 5. Convex-hull set splitting guided by the gradient-consistency principle.

3.4. Convex-hull Adaptation

Convex-hull representation facilitates localizing objects in arbitrary layouts. However, how to adapt the features to densely packed objects, which have feature aliasing, remains a problem. Recent learning-to-match methods [40, 39, 14] have taken steps toward feature adaption; nevertheless, all of them are defined for single objects, ignoring joint feature optimization for multiple dense objects.

Convex-hull Set Construction. We propose to construct a convex-hull set for each object so that an object can be matched with multiple convex-hulls, making it possible to jointly optimize features for densely packed objects, Fig. 4. A convex-hull set is constructed by selecting top-I convex-hulls as positive candidates, according to the CIoU between the convex-hulls and ground-truth boxes. It can also be constructed with an experimentally determined CIoU threshold. Convex-hulls not belonging to any convex-hull set are merged to the negative set S_{-}.

Denote the convex-hull set for the j-th object as S_j. The loss for S_j is defined as

$$\mathcal{L}_{S_j}^{+}(\theta) = \frac{1}{|S_j|}\sum_{i\in S_j}\omega_i \mathcal{L}_i^{+}(\theta), \tag{10}$$

where ω_i denotes the confidence of the i-th convex-hull, $\mathcal{L}_i^{+}(\theta)$ the prediction loss of the i-th convex-hull. When multiple objects come together, not all convex-hull features within a convex-hull set are proper to represent the object. The convex-hulls with large feature aliasing should be categorized to negatives while the convex-hulls shared by multiple objects should have low confidence.

Feature Anti-Aliasing. To alleviate feature aliasing, we first propose a convex-hull set splitting strategy, *i.e.*, dynamically evaluating the convex-hulls to select positive and negative ones. To this end, we define the weight in Eq. 10 as $\omega_i = f(\mathcal{L}_i^{+}(\theta))$, and have

$$\mathcal{L}_{s_j}^{+}(\theta) = \frac{1}{|S_j|}\sum_{i\in S_j} f(\mathcal{L}_i^{+}(\theta))\mathcal{L}_i^{+}(\theta), \tag{11}$$

where $f(x)$ is a monotonically decreasing function defined on the Gaussian cumulative error function[1], Fig. 5(a), which implies that the convex-hull of a smaller loss provides a greater object prediction confidence.

Convex-hull set splitting is guided by a gradient-consistency principle. Specifically, taking the derivative of Eq. 11, we have the gradient for the convex-hull set, as

$$\frac{\partial \mathcal{L}_{s_j}^{+}(\theta)}{\partial\theta} = \frac{1}{|S_j|}\sum_{i\in S_j}\frac{\partial(f(\mathcal{L}_i^{+}(\theta))\mathcal{L}_i^{+}(\theta))}{\partial\mathcal{L}_i^{+}(\theta)}\frac{\partial\mathcal{L}_i^{+}(\theta)}{\partial\theta}. \tag{12}$$

The gradient-consistency principle requires that the gradient $\frac{\partial\mathcal{L}_i^{+}(\theta)}{\partial\theta}$ of each positive convex-hull has the same direction with that of the convex-hull set $\frac{\partial\mathcal{L}_{s_j}^{+}(\theta)}{\partial\theta}$, Fig. 5(b). In other words, the convex-hulls of inconsistent gradient directions are supposed to cause feature aliasing. This means that when $\frac{\partial(f(\mathcal{L}_i^{+}(\theta))\mathcal{L}_i^{+}(\theta))}{\partial\mathcal{L}_i^{+}(\theta)}$ is positive, $\mathcal{C}_i$ is a positive convex-hull, and otherwise a negative one. As shown in Fig. 5, when sorting $\mathcal{L}_i^{+}(\theta)$ in an increasing order, $f(\mathcal{L}_i^{+}(\theta))\mathcal{L}_i^{+}(\theta)$ defines an upper convex function with a single extreme value, Fig. 5(a). Function $\frac{\partial(f(\mathcal{L}_i^{+}(\theta))\mathcal{L}_i^{+}(\theta))}{\partial\mathcal{L}_i^{+}(\theta)}$ has a single zero point and the convex-hulls are split to positives or negatives by this zero point. According to the gradient consistency principle, convex-hulls are dynamically partitioned to the positive set S_j or the negative set S_{-}.

On the other hand, to handle feature aliasing, we introduce an anti-aliasing coefficient

$$p_i = \gamma\cdot\frac{\mathrm{CIoU}(\mathcal{C}_i,\mathcal{B}_j)}{\sum_{m=1}^{M}\mathrm{CIoU}(\mathcal{C}_i,\mathcal{B}_m)}, \tag{13}$$

which indicates the degree that $\mathcal{C}_i$ belongs to a single object when it overlaps M objects. γ is an anti-aliasing factor. With the anti-aliasing coefficient, Eq. 11 is updated to

$$\mathcal{L}_{s_j}^{+}(\theta) = \frac{1}{|S_j|}\sum_{i\in S_j} p_i f(\mathcal{L}_i^{+}(\theta))\mathcal{L}_i^{+}(\theta). \tag{14}$$

Optimization. As shown in Fig. 2, the optimization of the second stage is driven by the joint classification and localization loss defined on convex-hull sets, as

$$\begin{aligned}\mathcal{L}^{det2}(\theta) =& \frac{1}{J}\sum_{j=1}^{J}\frac{1}{|S_j|}\sum_{i\in S_j} p_i f(\mathcal{L}_i^{+}(\theta))\mathcal{L}_i^{+}(\theta)\\ &+\frac{1}{|S_{-}|}\sum_{i\in S_{-}}\mathcal{L}_i^{-}(\theta),\end{aligned} \tag{15}$$

where S_j denotes the positive convex-hull set for the j-th object and S_{-} the negative convex-hull set. Eq. 15 jointly

[1] $f(x) = 1.0 - \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$

Table 1. Detection performance on DOTA. The category names are abbreviated as follows: 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, SF-Swimming Pool, and HC-HeliCopter. $(\cdot)^*$ indicates multi-scale test. $(\cdot)^\dagger$ indicates single-scale test performance provided by the authors.

Method	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
one-stage method																	
SSD [6]	-	44.74	11.21	6.22	6.91	2.00	10.24	11.34	15.59	12.56	17.94	14.73	4.55	4.55	0.53	1.01	10.94
YOOLOv2 [25]	-	76.90	33.87	22.73	34.88	38.73	32.02	52.37	61.65	48.54	33.91	29.27	36.83	36.44	38.26	11.61	39.20
FR-O [31]	ResNet101	79.09	69.12	17.17	63.49	34.20	37.16	36.20	89.19	69.60	58.96	49.40	52.52	46.69	44.80	46.30	52.93
R3Det [33]	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
two-stage method																	
R-DFPN [34]	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
R2CNN [12]	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
ICN [1]	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
RoI-Transformer [4]	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
SCRDet [37]	ResNet101	89.41	78.83	50.02	65.59	69.96	57.63	72.26	90.73	81.41	84.39	52.76	63.62	62.01	67.62	61.16	69.83
SCRDet* [37]	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
CSL† [35]	ResNet152	**90.14**	83.97	54.25	67.84	70.44	73.51	77.62	90.71	85.90	86.45	63.30	65.78	73.83	70.24	**68.93**	74.86
Gliding Vertex* [32]	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
CSL* [35]	ResNet152	90.25	85.53	54.64	75.31	70.44	73.51	77.62	90.84	86.15	86.69	69.60	68.04	73.83	71.10	68.93	76.17
anchor-free method																	
IE-Net [18]	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
CenterNet [42]	Hourglass104	89.02	69.71	37.62	63.42	65.23	63.74	77.28	90.51	79.24	77.93	44.83	54.64	55.93	61.11	45.71	65.04
DRN [24]	Hourglass104	88.91	80.22	43.52	63.35	73.48	70.69	84.94	90.14	83.85	84.11	50.12	58.41	67.62	68.60	52.50	70.70
DRN* [24]	Hourglass104	89.45	83.16	48.98	62.24	70.63	74.25	83.99	90.73	84.60	85.35	55.76	60.79	71.56	68.82	63.92	72.95
CFA(ours)	ResNet101	89.26	81.72	51.81	67.17	79.99	78.25	84.46	90.77	83.40	85.54	54.86	67.75	73.04	70.24	64.96	75.05
CFA(ours)	ResNet152	89.08	83.20	**54.37**	66.87	**81.23**	**80.96**	**87.17**	90.21	84.32	86.09	52.34	69.94	**75.52**	**80.76**	67.96	**76.67**

Table 2. Detection performance on SKU110K-R.

Method	mAP	AP_{75}	AR_{300}
YoloV3-Rotate	49.1	51.1	58.2
CenterNet-4point [42]	34.3	19.6	42.2
CenterNet [42]	54.7	61.1	62.2
DRN [24]	55.9	63.1	63.3
CFA(ours)	**57.0**	**63.5**	**63.9**

considers feature correspondence to multiple objects and penalizing convex-hull shared by multiple objects, alleviating feature aliasing towards optimal feature adaptation. Combing the Eq. 9 and Eq. 15, we have the overall loss function $\mathcal{L}^{det1}(\theta) + \mathcal{L}^{det2}(\theta)$ for CFA detector training.

4. Experiments

In this section, we first describe the experimental settings. We then report the performance of the proposed CFA detector and compare it with the state-of-the-art methods. We finally present visualization analysis and ablation studies.

4.1. Datasets

DOTA. The dataset is specified for objects in aerial scenarios, with 2,806 images and 15 object categories from remote sensing platforms. Objects are of various scales, orientations and layouts. The image sizes range from around 800×800 to 4,000×4,000 pixels. The objects are annotated by oriented bounding boxes, each of which has four vertexes. Half of the images are randomly selected for training, $1/6$ for validation, and $1/3$ for testing.

SKU110K-R. SKU110K [7] involves 11,762 images captured from supermarkets, with 1,733,678 objects in various scales, orientations, lighting conditions and crowdedness. In experiments, 8,233 images are used for training, 584 for validation, and 2941 for test. A recent work [24] relabelled the dataset (SKU110K-R) by augmenting the images to six orientations ($-45°$, $-30°$, $-15°$, $15°$, $30°$, $45°$).

4.2. Experimental Settings

Evaluation Protocols. After establishing settings in [24], we divided large-resolution images into sub-images with resolution 1024×1024 and an overlap of 200 pixels between sub-images. Detection results from sub-images were merged to the final detection results. For performance evaluation, the mAP metric [5] is applied for DOTA and the AP metric [17] (which reports a mean average precision at IoU=0.5:0.05:0.95.) is applied for SKU110K-R. The definition of the recall rate AR_{300} follows [7].

Implementation details. When training detectors, random horizontal flip and multi-scale variations in range of [768, 1280] are used for data augmentation. The detectors are trained with the SGD optimizer with a batch size 16 on eight Tesla V100 GPUs. The weight decay and momentum are 0.0001 and 0.9, respectively. On DOTA, the detectors are trained with 40 epochs in total. The learning rate is initialized as $8e$-3 and reduced by a magnitude after 24th, 32nd and 38th epochs. On SKU110K-R, the detectors are trained with 24 epochs in total. The learning rate is initialized as

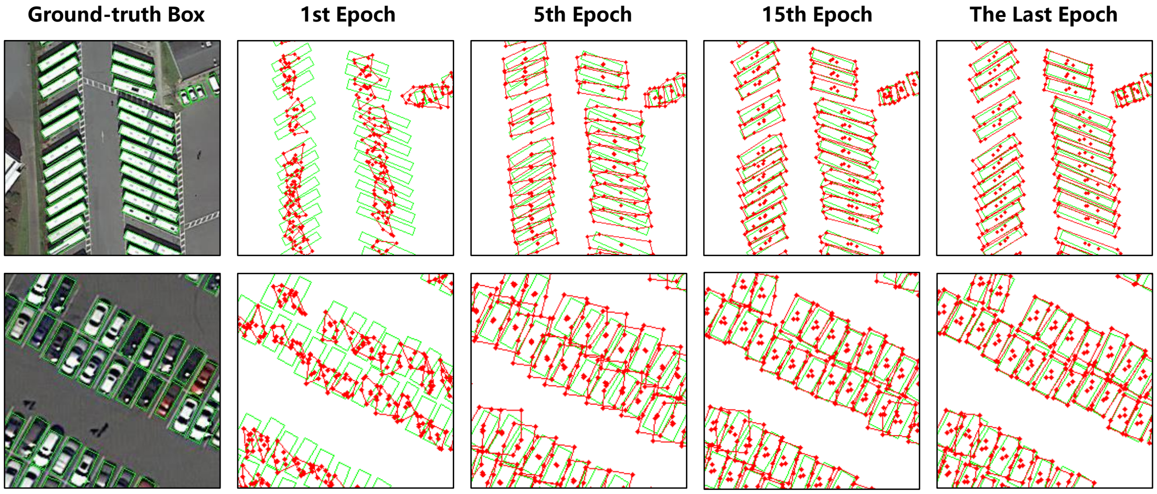

Figure 6. Convex-hull evolution during training. Convex-hulls are in red and ground-truth boxes are in green. (Best viewed in color)

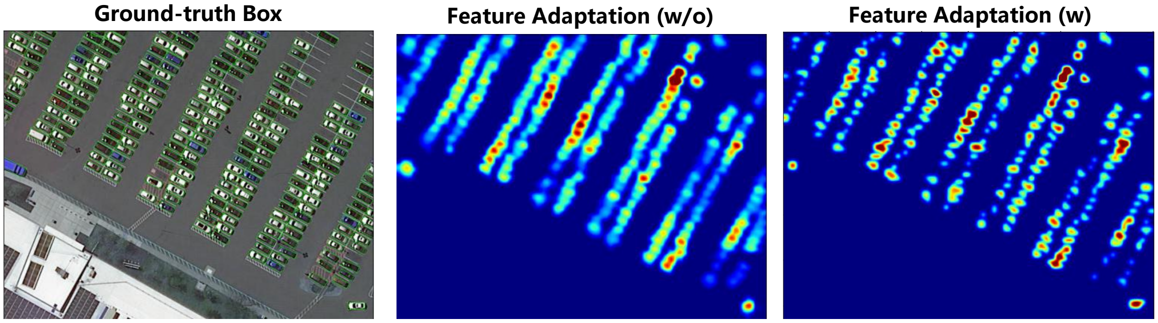

Figure 7. Heatmap comparison of CFA without (middle) and with (right) feature adaptation. Higher values correspond to small positive convex-hull loss $(1/\mathcal{L}_i^+(\theta))$ in the training phase. (Best viewed in color)

$8e$-3 and reduced by a magnitude after 16 and 22 epochs.

In test, after NMS the IoU threshold 0.4 and score threshold 0.05 are applied to obtain the detection results. For a fair comparison with most existing methods, we report only the single-scale test results. While the ResNet-50 backbone network is used for ablation study, the ResNet-101 and ResNet-152 are used to compare with the state-of-the-art detectors.

4.3. Performance

In Table 1, the proposed CFA detector is compared with the state-of-the-art detectors on DOTA for the oriented bounding box (OBB) task[31]. As an anchor-free detector, CFA outperforms the state-of-the-art DRN [24] detector by 5.97% (76.67% vs 70.70%), which is a large margin. Particularly, for the SV, LV, RA, HA and SP categories, it outperforms DRN by 7.75% (81.23% vs. 73.48%), 10.27% (80.96% vs. 70.69%), 11.53% (69.94% vs. 58.41%), 7.90%(75.52% vs. 67.62%), and 12.16%(80.76% vs. 68.60%). The reason lies in that these object categories are of irregular shapes and our CFA approach with a convex-hull representation can be more adaptive to such irregular object shapes and layouts. As an anchor-free detector, our CFA is comparable to, if not better than, most anchor based detectors including RoI-Transformer [4], SCRDet [37], Gliding Vertex [32] and CSL [35].

In Table 2, CFA is compared with the state-of-the-art detector (DRN) on SKU110K-R. CFA achieves 57.0% AP and improves the state-of-the-art by 1.1% (57.0% vs. 55.9%), despite that DRN uses the larger backbone network (Hourglass 104).

4.4. Visualization Analysis

Convex-hull Layout. In Fig. 6, we show the evolution of convex-hulls during training. One can see that after initialization the convex-hulls gradually approach the ground-truth boxes. Most of the feature points, either in or on the convex-hulls, localize within object extent, which facilitates both object localization and classification.

Feature Adaptation. In Fig. 7, the heatmaps with and without feature adaptation are compared. One can see that with feature adaptation, the heatmaps of densely packed objects are clearly separated. This validates the feature anti-aliasing effect of CFA.

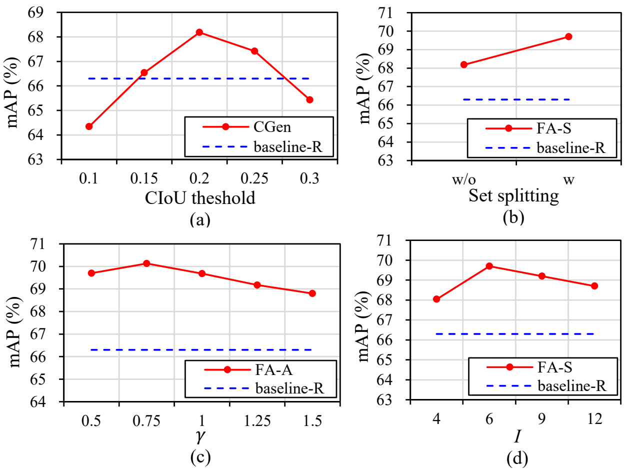

Figure 8. Evaluation of hyperparameters and modules. (a) CIoU threshold. (b) Convex-hull set size (I). (c) Anti-aliasing factor γ. (d) Feature adaptation.

Table 3. Ablation studies of modules in the proposed approach. "CGen" denotes convex-hull generation, "FA-S" denotes feature adaption by set splitting, and "FA-A" feature adaption with anti-aliasing. The backbone network is ResNet-50.

CIoU	CGen	FA-S	FA-A	mAP	Δ	$\sum\Delta$
				65.35		
✓				66.30	+0.95	0.95
✓	✓			68.18	+1.88	2.84
✓	✓	✓		69.70	+1.52	4.36
✓	✓	✓	✓	**70.13**	+0.43	4.79

Table 4. Comparison of the CIoU loss and Smoothed-L1 loss.

Method	CIoU	Smoothed-L1
mAP	66.30	65.35

4.5. Ablation Study

We conducted ablation studies using the DOTA validation set to verify the effect of CFA. RepPoint [38] is selected as the baseline detector. To detect oriented objects, we improved the RepPoint detector by adding an orientation prediction branch at the refinement stage and term the detector "baseline-R".

CIoU. As analyzed in Sec. 3.3, the CIoU loss reflects the overlap of two oriented bounding boxes. Minimizing CIoU loss drives optimizing object bounding boxes. As shown in Table 4, compared with the Smoothed-L1 loss [26], CIoU loss improves the mAP by 0.95% (66.30% vs 65.35%).

Convex-hull Generation. As shown in Fig. 6, by modeling objects as convex-hulls, we alleviated feature aliasing from both the background and other neighboring objects. During training, the convex-hulls are generated and are adaptive to the object extent, in a progressive fashion. In Table 3, convex-hull generation (CGen) improves the performance by 1.88% (68.18% vs 66.30%). In Fig. 8(a) under an experimentally determined CIoU threshold in the assignment process in second stage. Through the ablation study, we found the best CIoU threshold 0.2.

Convex-hull Set Splitting. As discussed in Sec. 3.4, convex-hull set construction is a process to define candidate features while convex-hull set splitting implements feature selection. As shown in Table 3 and Fig. 8(b), by using convex-hull set splitting, we improve the detection performance by 1.52% (69.70% vs 68.18%), validating the proposed gradient-consistency principle for feature anti-aliasing. In Fig. 8(d), we experimentally validated that assigning six convex-hulls per feature pyramid level (I=6) to each set can achieve the best performance.

Anti-aliasing Coefficient. By introducing feature anti-aliasing coefficient ("FA-A" in Table 3), feature adaptation for multiple objects is implemented and the performance is further improved by 0.43% (70.13% vs. 69.70%) under the best anti-aliasing factor (γ=0.75) in Fig. 8(c). In total, the CFA detector improves the baseline by 4.79% mAP.

Computational Cost. With a ResNet-50 backbone on a single Tesla V100 GPU, CFA spends 0.080s to process an image with 1024 × 1024 resolution while the baseline detector spends 0.075s. The DRN detector with an Hourglass-52, which is larger than ResNet-50, spends 0.102s. As CFA does not involve additional network architecture and the loss is only applied in the training phase, the computational cost overhead in the inference phase is negligible.

5. Conclusion

We proposed convex-hull feature adaptation (CFA), which is an elegant and effective approach to configure convolutional features for objects with irregular layouts. By introducing convex-hulls, CFA implemented adaptive feature representation in accordance with the layouts of oriented and densely packed objects. With convex-hull set splitting and feature anti-aliasing strategies, CFA implemented feature adaptation towards optimal feature assignment around objects. Extensive experiments on commonly used benchmarks validated CFA's superiors performance. This is in striking contrast with the state-of-the-art anchor-free detectors. CFA provides a fresh insight for detecting objects of irregular layouts.

Acknowledgement. This work was supported by Natural Science Foundation of China (NSFC) under Grant 61836012, 61620106005 and 61771447, the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDA27000000, CAAI-Huawei MindSpore Open Fund and MindSpore deep learning computing framework at www.mindspore.cn.

References

[1] Shiqi Chen, Ronghui Zhan, and Jun Zhang. Geospatial object detection in remote sensing imagery based on multiscale single-shot detector with activated semantics. *IEEE Geosci. Remote. Sens. Lett.*, page 820, 2018. 6

[2] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *IEEE ICCV*, pages 764–773, 2017. 2, 4

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009. 1

[4] Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu. Learning roi transformer for oriented object detection in aerial images. In *IEEE CVPR*, pages 2849–2858, 2019. 1, 2, 6, 7

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

[6] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C. Berg. DSSD : Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017. 6

[7] Eran Goldman, Roei Herzig, Aviv Eisenschtat, Jacob Goldberger, and Tal Hassner. Precise detection in densely packed scenes. In *IEEE CVPR*, pages 5227–5236, 2019. 6

[8] Ju Han and Kai-Kuang Ma. Rotation-invariant and scale-invariant gabor features for texture image retrieval. *Image Vis. Comput.*, 25(9):1474–1481, 2007. 2

[9] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NeuralIPS*, pages 2017–2025, 2015. 2

[10] R. A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*, 2(1):18–21, 1973. 4

[11] Yunho Jeon and Junmo Kim. Active convolution: Learning the shape of convolution for image classification. In *IEEE CVPR*, pages 1846–1854, 2017. 2

[12] Yingying Jiang, Xiangyu Zhu, Xiaobing Wang, Shuli Yang, Wei Li, Hua Wang, Pei Fu, and Zhenbo Luo. R2CNN: rotational region CNN for orientation robust scene text detection. *arXiv preprint arXiv:1706.09579*, 2017. 6

[13] Wei Ke, Tianliang Zhang, Zeyi Huang, Qixiang Ye, Jianzhuang Liu, and Dong Huang. Multiple anchor learning for visual object detection. In *IEEE CVPR*, pages 10203–10212, 2020. 3

[14] Kang Kim and Hee Seok Lee. Probabilistic anchor assignment with iou prediction for object detection. *arXiv preprint arXiv:2007.08103*, 2020. 3, 5

[15] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *IEEE CVPR*, pages 936–944, 2017. 1

[16] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE ICCV*, pages 2999–3007, 2017. 1, 4

[17] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014. 1, 6

[18] Youtian Lin, Pengming Feng, and Jian Guan. Ienet: Interacting embranchment one stage anchor free detector for orientation aerial object detection. *arXiv preprint arXiv:1912.00969*, 2019. 6

[19] Kang Liu and Gellért Máttyus. Fast multiclass vehicle detection on aerial images. *IEEE Geosci. Remote. Sens. Lett.*, pages 1938–1942, 2015. 1, 2

[20] Zikun Liu, Jingao Hu, Lubin Weng, and Yiping Yang. Rotated region based CNN for ship detection. In *IEEE ICIP*, pages 900–904, 2017. 1, 2

[21] Zikun Liu, Hongzhen Wang, Lubin Weng, and Yiping Yang. Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds. *IEEE Geosci. Remote. Sens. Lett.*, pages 1074–1078, 2016. 1, 2

[22] David G. Lowe. Object recognition from local scale-invariant features. In *IEEE ICCV*, pages 1150–1157, 1999. 2

[23] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002. 2

[24] Xingjia Pan, Yuqiang Ren, Kekai Sheng, Weiming Dong, Haolei Yuan, Xiaowei Guo, Chongyang Ma, and Changsheng Xu. Dynamic refinement network for oriented and densely packed object detection. In *IEEE CVPR*, pages 11204–11213, 2020. 1, 2, 3, 6, 7

[25] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE CVPR*, pages 779–788, 2016. 6

[26] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeuralIPS*, pages 91–99, 2015. 8

[27] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *IEEE CVPR*, pages 658–666, 2019. 4

[28] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *IEEE ICCV*, pages 8429–8438, 2019. 1

[29] Fang Wan, Chang Liu, Wei Ke, Xiangyang Ji, Jianbin Jiao, and Qixiang Ye. C-mil: Continuation multiple instance learning for weakly supervised object detection. In *IEEE CVPR*, pages 2199–2208, 2019. 3

[30] Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. CARAFE: content-aware re-assembly of features. In *IEEE ICCV*, pages 3007–3016, 2019. 2

[31] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge J. Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. DOTA: A large-scale dataset for object detection in aerial images. In *IEEE CVPR*, pages 3974–3983, 2018. 1, 2, 6, 7

[32] Yongchao Xu, Mingtao Fu, Qimeng Wang, Yukang Wang, Kai Chen, Gui-Song Xia, and Xiang Bai. Gliding vertex on the horizontal bounding box for multi-oriented object detection. *arXiv preprint arXiv:1911.09358*, 2019. 2, 6, 7

[33] Xue Yang, Qingqing Liu, Junchi Yan, and Ang Li. R3det: Refined single-stage detector with feature refinement for rotating object. *arXiv preprint arXiv:1908.05612*, 2019. 2, 3, 6

[34] Xue Yang, Hao Sun, Kun Fu, Jirui Yang, Xian Sun, Menglong Yan, and Zhi Guo. Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks. *IEEE Geosci. Remote. Sens. Lett.*, page 132, 2018. 6

[35] Xue Yang and Junchi Yan. Arbitrary-oriented object detection with circular smooth label. In *ECCV*, 2020. 2, 6, 7

[36] Xue Yang, Junchi Yan, Xiaokang Yang, Jin Tang, Wenlong Liao, and Tao He. Scrdet++: Detecting small, cluttered and rotated objects via instance-level feature denoising and rotation loss smoothing. *arXiv preprint arXiv:2004.13316*, 2020. 1

[37] Xue Yang, Jirui Yang, Junchi Yan, Yue Zhang, Tengfei Zhang, Zhi Guo, Xian Sun, and Kun Fu. Scrdet: Towards more robust detection for small, cluttered and rotated objects. In *IEEE ICCV*, pages 8231–8240, 2019. 1, 2, 6, 7

[38] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *IEEE ICCV*, pages 9656–9665, 2019. 2, 3, 8

[39] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *IEEE CVPR*, pages 9756–9765, 2020. 3, 5

[40] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. Freeanchor: Learning to match anchors for visual object detection. In *Neural Information Processing Systems*, pages 147–155, 2019. 3, 5

[41] Xiaosong Zhang, Fang Wan, Chang Liu, Xiangyang Ji, and Qixiang Ye. Learning to match anchors for visual object detection. *IEEE Trans. Pattern Anal. Machine Intell., 10.1109/TPAMI.2021.3050494*, pages 1–14, 2021. 3

[42] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 6

[43] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *IEEE CVPR*, pages 850–859, 2019. 3

[44] Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Oriented response networks. In *IEEE CVPR*, pages 4961–4970, 2017. 2