

Object Detection in Aerial Images: A Large-Scale Benchmark and Challenges

Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Michael Ying Yang,
Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, Liangpei Zhang

Abstract—In the past decade, object detection has achieved significant progress in natural images but not in aerial images, due to the massive variations in the scale and orientation of objects caused by the bird's-eye view of aerial images. More importantly, the lack of large-scale benchmarks has become a major obstacle to the development of object detection in aerial images (ODAI). In this paper, we present a large-scale *Dataset of Object deTection in Aerial images* (DOTA) and comprehensive baselines for ODAI. The proposed DOTA dataset contains 1,793,658 object instances of 18 categories of oriented-bounding-box annotations collected from 11,268 aerial images. Based on this large-scale and well-annotated dataset, we build baselines covering 10 state-of-the-art algorithms with over 70 configurations, where the speed and accuracy performances of each model have been evaluated. Furthermore, we provide a code library for ODAI and build a website for evaluating different algorithms. Previous challenges run on DOTA have attracted more than 1300 teams worldwide. We believe that the expanded large-scale DOTA dataset, the extensive baselines, the code library and the challenges can facilitate the designs of robust algorithms and reproducible research on the problem of object detection in aerial images.

Index Terms—Object detection, remote sensing, aerial images, oriented object detection, benchmark dataset.

arXiv:2102.12219v2 [cs.CV] 4 Dec 2021

INTRODUCTION

Currently, Earth vision (also known as Earth observation and remote sensing) technologies enable us to observe the

Earth's surface with aerial images¹ with a resolution up to a half meter. Although challenging, developing mathematical tools and numerical algorithms is necessary for interpreting these huge volumes of images, among which object detection refers to localizing objects of interest (e.g., vehicles and ships) on the Earth's surface and predicting their categories. Object detection in aerial images (ODAI) has been an essential step in many real-world applications such as urban management, precision agriculture, emergency rescue and disaster relief [1], [2]. Although extensive studies have been devoted to object detection in aerial images and appreciable breakthroughs have been made [3]–[8], the task has numerous difficulties such as arbitrary orientations, scale variations, extremely nonuniform object densities and large aspect ratios (ARs), as shown in Fig. 1.

Among these difficulties, the arbitrary orientation of objects caused by the overhead view is the main difference between natural images and aerial images, and it complicates the object detection task in two ways. First, rotation-invariant feature representations are preferred in the detection of arbitrarily orientated objects, but they are often beyond the capability of most of current deep neural network models. Although the methods such as those designed in [6], [9], [10] use rotation-invariant convolutional neural networks (CNNs), the problem is far from solved. Second, the horizontal bounding box (HBB) object representation used in conventional object detection [11]–[13] cannot localize the oriented objects precisely, such as ships and large vehicles, as shown in Fig. 1. The oriented bounding box (OBB) object representation is more appropriate for aerial

J. Ding and L. Zhang are with the State Key Lab. LIESMARS, Wuhan University, Wuhan, 430079, China. Email: {jian.ding, zlp62}@whu.edu.cn.

N. Xue is with the National Engineering Research Center for Multimedia Software, School of Computer Science and Institute of Artificial Intelligence, Wuhan University, Wuhan, 430072, China. Email: xuenan@whu.edu.cn.

G.-S. Xia is with the National Engineering Research Center for Multimedia Software, School of Computer Science and Institute of Artificial Intelligence, and also the State Key Lab. LIESMARS, Wuhan University, Wuhan, 430072, China. Email: guisong.xia@whu.edu.cn.

X. Bai is with the School of Electronic Information, Huazhong University of Science and Technology, Wuhan, 430079, China. Email: xbai@hust.edu.cn.

W. Yang is with the School of Electronic Information, and the State Key Lab. LIESMARS, Wuhan University, Wuhan, 430072, China. Email: yangwen@whu.edu.cn.

• M. Y. Yang is with Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, the Netherlands. Email: michael.yang@utwente.nl.

• S. Belongie is with Department of Computer Science, Cornell University and Cornell Tech. Email: sjb344@cornell.edu.

• J. Luo is with Department of Computer Science, University of Rochester, Rochester, NY 14627. Email: jluo@cs.rochester.edu.

• M. Datcu is with Remote Sensing Technology Institute, German Aerospace Center (DLR), 82234, Germany, and also the University POLITEHNICA of Bucharest (UPB), Romania. Email: mihai.datcu@dlr.de.

• M. Pelillo is with DAIS, Ca' Foscari University of Venice, Italy. Email: pelillo@unive.it.

• The studies in this paper have been supported by the NSFC projects under the contracts No.61922065, No.61771350 and No.41820104006. Dr. Nan Xue was also supported by National Post-Doctoral Program for Innovative Talents under Grant BX20200248. Mihai Datcu was supported by the CNCS-UEFISCDI, project number PN-III-P4-ID-PCE-2020-2120, within PNCDI III.

• The corresponding author is Gui-Song Xia (guisong.xia@whu.edu.cn).

1. This paper uses the term "aerial" to refer to any overhead image looking approximately straight down onto the Earth, including both satellite images and airborne images, for simplification unless otherwise indicated. We use the term "airborne" if we do not want to include the satellite images.

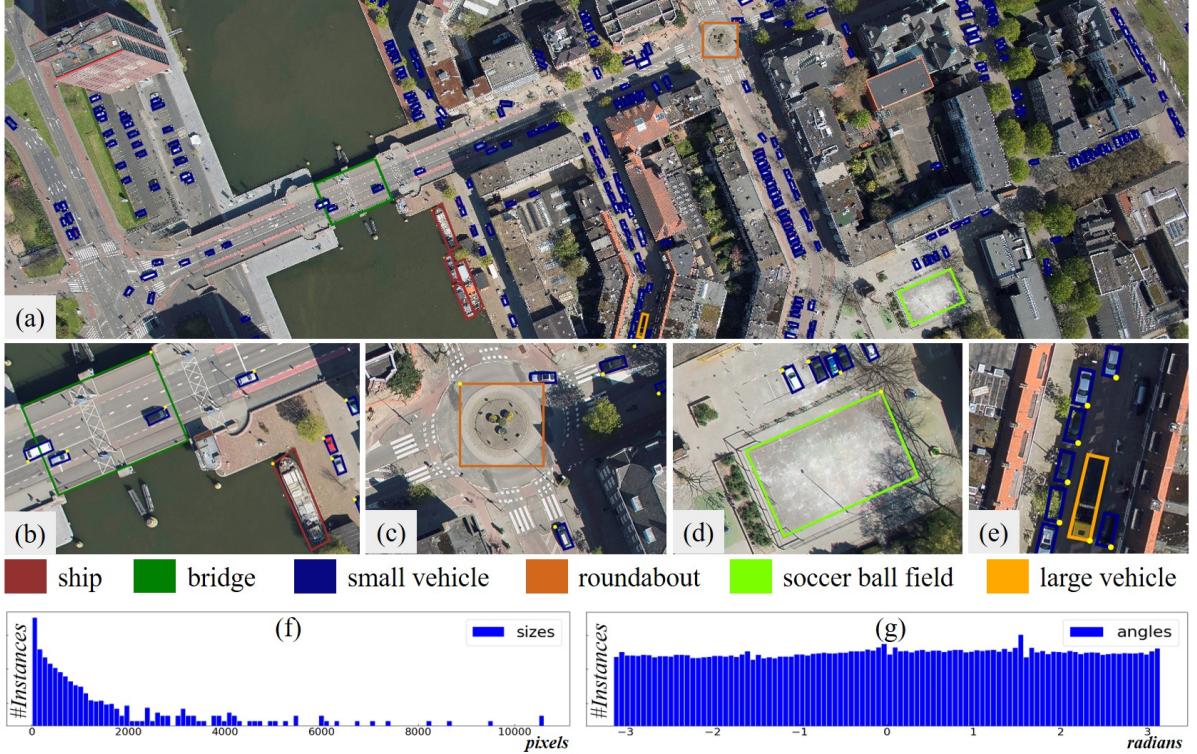


Fig. 1. An example image taken from DOTA. (a) A typical image in DOTA consisting of many instances from multiple categories. (b), (c), (d), (e) are cropped from the source image. We can see that instances such as small vehicles have arbitrary orientations. There is also a massive scale variation across different instances. Moreover, the instances are not distributed uniformly. The instances are sparse in most areas but crowded in some local areas. Large vehicles and ships have large ARs. (f) and (g) exhibit the size and orientation histograms, respectively, for all instances.

images [4], [14]–[17]. It allows us to distinguish densely packed instances (as shown in Fig. 3) and extract rotation-invariant features [4], [18], [19]. The OBB object representation actually introduces a new object detection task, called *oriented object detection*. In contrast with horizontal object detection [8], [20]–[22], oriented object detection is a recently emerging research direction and most of the methods for this new task attempt to transfer successful deep-learning-based object detectors pre-trained on large-scale natural image datasets (*e.g.*, ImageNet [12] and Microsoft Common Objects in Context (MS COCO) [13]) to aerial scenes [18], [19], [23]–[25] due to the lack of large-scale annotated aerial image datasets.

To mitigate the dataset problem, some public datasets of aerial images have been created, see *e.g.* [7], [15]–[17], [26], but they contain a limited number of instances and tend to use images taken under ideal conditions (*e.g.*, clear backgrounds and centered objects), which cannot reflect the real-world difficulties of the problem. The recently released xView [27] dataset provides a wide range of categories and contains large quantities of instances in complicated scenes. However, it annotates the instances with HBBs instead of the more precise OBBs. Thus, a large-scale dataset that has OBB annotations and reflects the difficulties in real-world applications of aerial images is in high demand.

Another issue with ODAI is that the module design and the hyperparameter setting of conventional object detectors learned from natural images are not appropriate for aerial images due to domain differences. Thus, when developing new algorithms, comprehensive baselines and sufficient ablative analyses of models on aerial images are

required. However, comparing different algorithms is difficult due to the diversities in hardware, software platforms, detailed settings and so on. These factors influence both speed and accuracy. Therefore, when building the baselines, implementing the algorithms with a unified code library and keeping the hardware and software platform the same is highly desirable. Nevertheless, current object detection libraries, *e.g.*, MMDetection [28] and Detectron [29], do not support oriented object detection.

To address the above-mentioned problems, in this paper we first extend the preliminary version of DOTA, *i.e.*, DOTA-v1.0 [14], to DOTA-v2.0. Specifically, DOTA-v2.0 collects 11,268 aerial images from various sensors and platforms and contains **approximately 1.8 million object instances** annotated with **OBBs** in 18 common categories, which, to our knowledge, is the largest public Earth vision object detection dataset. Then, to facilitate algorithm developments and comparisons with DOTA, we provide a well-designed code library that supports oriented object detection in aerial images. Based on the code library, we also build more comprehensive baselines than the preliminary version [14], keeping the hardware, software platform, and settings the same. In total, we evaluate 10 algorithms and over 70 models with different configurations. We then provide detailed speed and accuracy analyses to explore the module designs and parameter settings in aerial images to guide future research. These experiments verify the large differences in object detector design between natural and aerial images and provide materials for universal object detection algorithms [30].

The main contributions of this paper are three-fold:

- To the best of our knowledge, the expanded DOTA is the largest dataset for object detection in Earth vision. The OBB annotations of DOTA not only provide a large-scale benchmark for object detection in Earth vision but also pose interesting algorithmic questions and challenges to generalized object detection in computer vision.
- We build a code library for object detection in aerial images. This is expected to facilitate the development and benchmarking of object detection algorithms in aerial images with both HBB and OBB representations.
- With the expanded DOTA, we evaluate 10 representative algorithms over 70 model configurations, providing comprehensive analysis that can guide the designs of object detection algorithms in aerial images.

The dataset, code library, and regular evaluation server are available and maintained on the DOTA website². It is worth noting that the creation and use of DOTA have advanced object detection in aerial images. For instance, the regular DOTA evaluation server and two object detection contests organized at the 2018 International Conference on Pattern Recognition (ICPR' 2018 with DOTA-v1.0)³ and 2019 Conference on Computer Vision and Pattern Recognition (CVPR'2019 with DOTA-v1.5)⁴ have attracted approximately 1300 registrations. We believe that our new DOTA dataset, with a comprehensive code library and an online evaluation platform, will further promote the reproducible research in Earth vision.

2 RELATED WORK

Well-annotated datasets have played an important role in data-driven computer vision research [12], [13], [31]–[35] and have promoted cutting-edge research in a number of tasks such as object detection and classification. In this section, we first review object detection datasets of natural and aerial images. Then we discuss the recent deep learning based object detectors in aerial images. Finally, we briefly review the code libraries for object detection.

2.1 Datasets for Conventional Object Detection

As a pioneer, PASCAL Visual Object Classes (VOC) [11] held challenges on object detection from 2005 to 2012. The computer vision community has widely adopted PASCAL VOC datasets and their evaluation metrics. Specifically, the PASCAL VOC Challenge 2012 dataset contains 11,530 images, 20 classes, and 27,450 annotated bounding boxes. Later, the ImageNet dataset [12] was developed and is an order of magnitude larger than PASCAL VOC, containing 200 classes and approximately 500,000 annotated bounding boxes. However, non-iconic views are not addressed. Then MS COCO [13] was released, containing a total of 328K images, 91 categories, and 2.5 million labeled segmented objects. MS COCO has on average more instances and categories per image and contains more contextual information

than PASCAL VOC and ImageNet. It is worth noticing that, in Earth vision, the image size could be extremely large (*e.g.*, 20,000 × 20,000 pixels), so the number of images cannot reflect the scale of a dataset. In this case, the pixel area would be more reasonable when comparing the scale between the datasets of natural and aerial images. Moreover, the large images include more instances per image and contextual information. Tab. 1 provides the detailed comparisons.

TABLE 1

DOTA vs. general object detection datasets. *BBox* is short for bounding box, *Avg. BBox quantity* indicates the average number of bounding boxes per image. For PASCAL VOC (07++12), we count the whole PASCAL VOC 07 and training and validation (trainval) set of PASCAL VOC 12. DOTA has a comparable scale with the large-scale datasets for object detection in natural images. Note that for the average number of instances per image, DOTA surpasses the other datasets.

Dataset	Classes	Image quantity	Megapixel area	BBox quantity	Avg. BBox quantity
PASCAL VOC (07++12)	20	21,503	5,133	52,090	2.42
MS COCO (2014 trainval)	80	123,287	32,639	886,266	7.19
ImageNet (2014 train)	200	456,567	82,820	478,807	1.05
DOTA-v1.0	15	2,806	19,173	188,282	67.10
DOTA-v1.5	16	2,806	19,173	402,089	143.73
DOTA-v2.0	18	11,268	126,306	1,793,658	159.18

2.2 Datasets for Object Detection in Aerial Images

In aerial object detection, a dataset resembling MS COCO and ImageNet both in terms of the image number and detailed annotations has been missing, which becomes one of the main obstacles to research in Earth vision, especially for developing deep-learning-based algorithms. In Earth vision, many aerial image datasets are prepared for actual demands in a specific category, such as building datasets [7], [36], vehicle datasets [8], [15], [16], [26], [37]–[39], ship datasets [4], [40], and plane datasets [17], [41]. Although some public datasets [17], [42]–[45] have multiple categories, they have only limited number of samples, which are hardly efficient for training robust deep models. For example, NWPU [42] only contains 800 images, 10 classes and 3,651 instances.

To alleviate this problem, our preliminary work DOTA-v1.0 [14] presented a dataset with 15 categories and 188,282 instances, which for the first time enables us to efficiently train robust deep models for ODAI without the help of large-scale datasets of natural images, such as MS COCO and ImageNet. Later, iSAID [46] provided an instance segmentation extension of DOTA-v1.0 [14]. A notable dataset is xView [27], which contains 1,413 images, 16 main categories, 60 fine-grained categories, and 1 million instances. Another dataset DIOR [47] provided a comparable number of instances as DOTA-v1.0 [14]. However, the instances in xView and DIOR are both annotated by HBBs, which are not suitable for precisely detecting objects that are arbitrarily oriented in aerial images. In addition, VisDrone [48] is also a large-scale dataset for drone images but focuses more on video object detection and tracking. The image subset in VisDrone for object detection is not very large. Furthermore, most of the previous datasets are heavily imbalanced in favor of positive samples, whose negative samples are not sufficient to represent the real-world distribution.

2. <https://captain-whu.github.io/DOTA/>

3. <https://captain-whu.github.io/ODAI/results.html>

4. <https://captain-whu.github.io/DOAI2019/challenge.html>

TABLE 2

DOTA vs. object detection datasets in aerial images. *HBB* is horizontal bounding box, and *OBB* is oriented bounding box. *CP* is center point.

Dataset	Source	Annotation	# of main categories	Total # of categories	# of instances	# of images	Image width	Year
TAS [26]	satellite	HBB	1	1	1,319	30	792	2008
SZTAKI-INRIA [7]	multi source	OBB	1	1	665	9	~800	2012
NWPU VHR-10 [42]	multi source	HBB	10	10	3,651	800	~1000	2014
VEDAI [15]	satellite	OBB	3	9	2,950	1,268	512, 1024	2015
DLR 3k [16]	aerial	OBB	2	8	14,235	20	5616	2015
UCAS-AOD [17]	Google Earth	OBB	2	2	14,596	1,510	~1000	2015
COWC [37]	aerial	CP	1	1	32,716	53	2000–19,000	2016
HRSC2016 [4]	Google Earth	OBB	1	26	2,976	1,061	~1100	2016
RSOD [43]	Google Earth	HBB	4	4	6,950	976	~1000	2017
CARPPK [8]	drone	HBB	1	1	89,777	1,448	1280	2017
ITCVD [38]	aerial	HBB	1	1	228	23,543	5616	2018
LEVIR [44]	Google Earth	HBB	3	3	11,000	22,000	800–600	2018
xView [27]	satellite	HBB	16	60	1,000,000	1,413	~3000	2018
VisDrone [48]	drone	HBB	10	10	54,200	10,209	2000	2018
SpaceNet MVOI [36]	satellite	polygon	1	1	126,747	60,000	900	2019
HRRSD [45]	multi source	HBB	13	13	55,740	21,761	152–10569	2019
DIOR [47]	Google Earth	HBB	20	20	190,288	23,463	800	2019
iSAID [46]	multi source	polygon	14	15	655,451	2,806	800–13,000	2019
FGSD [40]	Google Earth	OBB	1	43	5,634	2,612	930	2020
RarePlanes [41]	satellite	polygon	1	110	644,258	50,253	1080	2020
DOTA-v1.0 [14]	multi source	OBB	14	15	188,282	2,806	800–13,000	2018
DOTA-v1.5	multi source	OBB	15	16	402,089	2,806	800–13,000	2019
DOTA-v2.0	multi source	OBB	17	18	1,793,658	11,268	800–20,000	2021

As we stated previously [14], a good dataset for aerial image object detection should have the following properties: 1) substantial annotated data to facilitate data-driven, especially deep-learning-based methods; 2) large images to contain more contextual information; 3) OBB annotation to describe the precise location of objects; and 4) balance in image sources, as pointed in [49]. DOTA is built considering these principles (unless otherwise specified, DOTA refers to DOTA-v2.0). Detailed comparisons of these existing datasets and DOTA are shown in Tab. 2. Compared to other aerial datasets, as we shall see in Sec. 4, DOTA is challenging due to its large number of object instances, arbitrary orientations, various categories, density distribution, and diverse aerial scenes from various image sources. These properties make DOTA helpful for real-world applications.

2.3 Deep Models for Object Detection in Aerial Images

Object detection in aerial images is a longstanding problem. Recently, with the development of deep learning, many researchers in Earth vision have adapted deep object detectors [50]–[54] developed for natural images to aerial images. However, the challenges caused by the domain shift need to be addressed. Here, we highlight some notable works.

Objects in aerial images are often arbitrarily oriented due to the bird’s-eye view, and the scale variations are larger than those in natural images. To handle rotation variations, a simple model [9] plugs an additional rotation-invariant layer into R-CNN [52] relying on rotation data augmentation. The oriented response network (ORN) introduces active rotating filters (ARF) to produce the rotation-invariant feature without using data augmentation, which is adopted by the rotation-sensitive regression detector (RRD) [23]. The deformable modules [55] designed for general object deformation are also widely used in aerial images. The methods mentioned above do not fully utilize the OBB annotations. When OBB annotations are available, a rotation R-CNN (RR-CNN) [56] uses rotation region-of-interest (RRoI) pooling

to extract rotation-invariant region features. However, RR-CNNs [56] generate proposals by hand-crafted way. Then the RoI Transformer [18] tries to use the supervision of OBBs to learn RoI-wise spatial transformation. The later S²A-Net [57] extracts spatially invariant features in one-stage detectors. To solve the challenges of scale variations, feature pyramids [19], [58] and image pyramids [24], [25] are widely used to extract scale-invariant features in aerial images. We evaluate the geometric transformation network modules and geometric data augmentations in Sec. 6.1.

Crowded instances represented by HBBs are difficult to distinguish (see Fig. 3). Traditional HBB-based non maximum suppression (NMS) will fail in such cases. Therefore, these methods [18], [24], [25] use rotated NMS (R-NMS), which requires precise detections to address this problem. Similar to text and face detection in natural scenes, e.g. [23], [59]–[61], precise ODAI can also be modeled as an oriented object detection task. Most of the previous works [14], [23]–[25], [62] consider it as a regression problem and regress the offsets of the OBB ground truth relative to anchors (or proposals). However, the definition of an OBB is ambiguous. For example, there are four permutations of the corner points in a quadrilateral. The Faster R-CNN OBB [14] solves it by using a defined rule to determine the order of points in OBBs. Work in [63] further uses the gliding offset and obliquity factor to eliminate the ambiguity. The circular smooth label (CSL) [64] transforms the regression of angle as a classification problem to avoid the problem. Mask OBB [65] and CenterMap [66] consider object detection as a pixel-level classification problem to avoid ambiguity. Mask-based methods converge more easily but have more floating point operations per second (FLOPS) than regression-based methods. We will give a more detailed comparison between them in one unified code library in Sec. 6.1.1.

The final challenge is detecting objects in large images. Aerial images are usually extremely large (over 20k × 20k pixels). Current GPU memory capacity is insufficient to process large images. Downsampling a large image to a

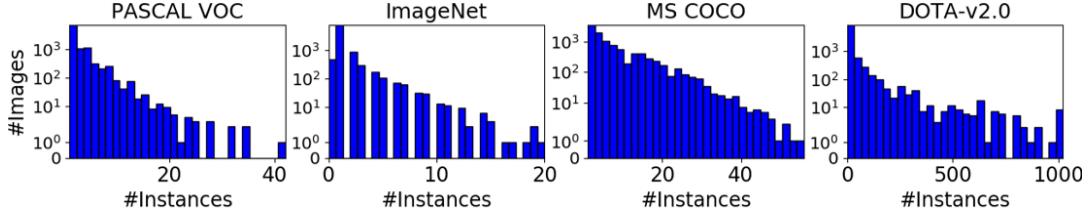


Fig. 2. Number of instances per image among DOTA and general object detection datasets. For PASCAL, ImageNet and MS COCO, we count the statistics of 10,000 random images. As the images in DOTA are very large ($20,000 \times 20,000$), for a fair comparison, we count the statistics of 10,100 image patches with the size of 1024×1024 , which is also the size used for the baselines in Sec. 5.2. DOTA-v2.0 has a wider range of the number of instances per image.

small size would lose the detailed information. To solve this problem [14], [16], the large images can be simply split into small patches. After obtaining the results on these patches, the results are integrated into large images. To speed up inference on large images, these methods [20]–[22], [67] first find regions that are likely to contain instances in the large images and then detect objects in the regions. In this paper, we simply follow the naive solutions [14], [16] to build baselines.

2.4 Code Libraries for Object Detection

The development of object detection algorithms is a sophisticated process. In addition, there are too many design choices and hyperparameter settings, which make comparisons between different methods difficult. Therefore, object detection code libraries such as the Tensorflow Object Detection API [68], Detectron [29], MaskRCNN-Benchmark [69], Detectron2 [70], MMDetection [28] and SimpleDet [71] are developed to facilitate the comparisons of object detection algorithms. These code libraries primarily use a modular design, which makes it easy to develop new algorithms. The current widely used settings, such as the training schedule, are from Detectron [29]. However, these code libraries mainly focus on horizontal object detection. Only Detectron2 [70] has limited support for oriented object detection. In our work, we enriched MMDetection [28] with several crucial operators for oriented object detection and evaluated 10 algorithms for object detection in aerial images.

3 CONSTRUCTION OF DOTA

3.1 Image Collection

In aerial images, the resolution and a variety of sensors are the factors that produce dataset biases [72]. To eliminate these biases, we collect images from various sensors and platforms with multiple resolutions, including Google Earth, the Gaofen-2 (GF-2) Satellite, Jilin-1 (JL-1) Satellite, and airborne images (taken by CycloMedia [73] in Rotterdam). To obtain the DOTA images, we first collected the coordinates of areas of interest (e.g., airports or harbors) from all over the world. Then, according to the coordinates, images are collected from Google Earth, GF-2 and JL-1 (GF&JL) satellites. The airborne images taken by CycloMedia [73] were obtained from five perspectives in Rotterdam, which include both oblique views and nadir views. The tilt angle of the oblique view was approximately 45° .

For the Google Earth images, we collect the images that contain instances of interest with sizes from 800×800 to 4000×4000 pixels. However, for the GF&JL satellite and

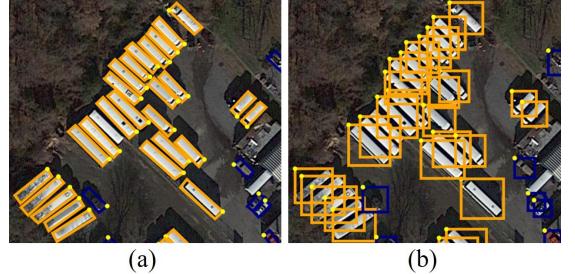


Fig. 3. Comparisons between HBB and OBB representations for objects. (a) OBB representation. (b) HBB representation. The HBB representation cannot distinguish rotated dense objects.

airborne images, we maintained their original sizes. Large images can approach real-world distributions, and also pose a challenge for finding small instances [20]. In DOTA-v2.0, the sizes of newly collected GF-2 satellite images and CycloMedia airborne images are usually $29,200 \times 27,620$ and $7,360 \times 4,912$ pixels, respectively.

3.2 Category Selection

We choose eighteen categories, *plane*, *ship*, *storage tank*, *baseball diamond*, *tennis court*, *swimming pool*, *ground track field*, *harbor*, *bridge*, *large vehicle*, *small vehicle*, *helicopter*, *roundabout*, *soccer ball field*, *basketball court*, *container crane*, *airport* and *helipad*. We select these categories according to their frequency of occurrence and value for real-world applications. The first 10 categories are common in the existing datasets, e.g., [16], [17], [37], [42]. Other categories are added considering their value in real-world applications. For example, we selected “helicopter” as moving objects are of significant importance in aerial images, and “roundabout” as it plays an essential role in roadway analyses. It is worth discussing whether to take “stuff” categories into account. There are usually no clear definitions for the “stuff” categories (e.g. *harbor*, *airport*, *parking lot*), as shown in the Scene UNderstanding (SUN) dataset [74]. However, their contextual information may be helpful for object detection. Based on this idea, we select the harbor and airport categories because their borders are relatively easy to define and there are abundant harbor and airport instances in our image sources.

3.3 Oriented Object Annotation

In computer vision, many visual concepts, such as region descriptions, objects, attributes, and relationships, are often represented with bounding boxes, as shown in [75]. A common representation of the bounding box is (x_c, y_c, w, h) , where (x_c, y_c) is the center location and w, h are the width and height, respectively, of the bounding box. We call this

type of bounding box an HBB. The HBB can describe objects well in most cases. However, it cannot accurately outline oriented instances such as text and objects in aerial images. As shown in Fig. 3, the HBB cannot differentiate densely-distributed oriented objects. The conventional NMS algorithm fails in such cases. On the other hand, the regional features extracted from HBBs are not rotation invariant. To address these problems, we represent the objects with OBBs. In detail, an OBB is denoted by $\{(x_i, y_i) | i = 1, 2, 3, 4\}$, where (x_i, y_i) denotes the position of the OBB's vertex in the image. The vertices are arranged in clockwise order.

The most straightforward way to annotate an OBB is to draw an HBB and then adjust the angle. However, since there is no reference for HBBs, several adjustments in the center, height, width and angle are usually needed to fit an arbitrarily oriented object well. Clicking on physical points lying on the object [76] could make crowd-sourced annotations more efficient for HBBs, as these points are easy to find. Inspired by this idea, we allow the annotators to click four corners of the OBBs. For most categories, the corners of the OBBs (*e.g.*, tennis court and basketball court) lie on or close to the objects (vehicles), however, there are still some categories whose shapes are very different from OBBs. For these categories, we annotate four key points lying on the object. For example, we annotate the planes with 4 key points, representing the head, two wingtips, and tail. Then we transfer the 4 key points to an OBB.

However, when using OBBs to represent objects, we could obtain four different representations for the same object by changing the order of the points. For example, assume that $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$ represents an object, but we could represent the same object by $(x_2, y_2, x_3, y_3, x_4, y_4, x_1, y_1)$. For categories having differences between the head and tail (*e.g.*, *helicopter*, *large vehicle*, *small vehicle*, *harbor*), we carefully select the first point to imply the “head” of the object. For other categories (*e.g.*, *soccer-ball field*, *swimming pool and bridge*) that do not have visual clues to determine the first point, we choose the top-left point as the starting point.

The detailed pipeline is described in the following. First, we developed a customized annotation tool. Based on this, we asked experts in the interpretation of aerial images to annotate some examples for each category. The annotations by the experts are used to train the volunteers for the large-scale annotation. After the training, we evaluate the annotation ability of volunteers to separate them into the *plain* and *senior* groups. The volunteers in the plain group are asked to yield the initial annotations which are doubly-checked by senior volunteers and the authors. The images that do not pass the checking were sent back to volunteers to improve the annotation quality. The volunteers were mainly recruited from Wuhan University, with a background in remote sensing image interpretation. Some examples of annotated patches are shown in Fig. 4.

Discussion. There are two types of possible errors in the object annotations: 1) missing annotations; 2) inaccurate bounding boxes annotations. Missing annotations is mainly caused by the difficulty in identifying tiny objects. The proportion of missed objects can be ignored and does not influence the training and evaluation of object detectors in DOTA-v2.0. However, if the researchers want to study this

TABLE 3
The statistics for the annotated objects across different data sources in DOTA-v1.5 and DOTA-v2.0. The total image areas, the area of objects and the ratio of foreground pixels of the annotated objects to the image areas are reported.

DOTA-v1.5				
	Google Earth	GF&JL	Aerial	All
# of images	2375	431	/	2806
Images Area (10^6)	11,873	7,301	/	19,173
Objects Area (10^6)	784	20	/	804
Foreground Ratio	0.066	0.003	/	0.042
DOTA-v2.0				
	10186	516	566	11268
Images Area (10^6)	29,991	75,854	20,462	126,306
Objects Area (10^6)	1,111	243	673	2,027
Foreground Ratio	0.037	0.003	0.033	0.016

problem and further improve the performance, we recommend researchers refer to these prior works [77], [78]. The inaccurate bounding boxes annotations exist in all object detection datasets since there exists ambiguity to define the boundary of objects sometimes (*e.g.*, the occlusion). Developing algorithms that modeling the inaccurate bounding boxes annotation has been studied in [79] for natural images. However, He *et al.* [79] only studied the inaccurate HBB annotation. The modeling of inaccurate annotation of the OBBs in DOTA can be researched in the future.

4 PROPERTIES OF DOTA

4.1 Image Sources

The images in DOTA-v2.0 are from three different sources, *i.e.*, Google Earth images, GF-2 and JL-1 (GF&JL) satellite images, and the CycloMedia [73] airborne images. Tab. 3 shows the statistics of three image sources in terms of the images area, objects area, and foreground ratio. We can see that the carefully selected Google Earth images contain the majority of positive samples. Nevertheless, the negative samples are also important to avoid positive sample bias [49]. The object distributions in the collected GF&JL satellite images and CycloMedia airborne images are close to those in real-world applications and provide enough background area. It is worthwhile to notice that DOTA-v2.0 contains both RGB images and grayscale images. More precisely, the images collected from Google Earth and CycloMedia are often RGB-rendered versions of original aerial images, and the images from GF-2, JL-1 are 8-bit per pixel optimally converted from their original panchromatic band in 10-bit. However, during those spectral rendering and bit-length optimization processes, the structure and appearance information of the image content are always consistent and the images are feasible for recognition-oriented tasks [34].

The acquisition dates are available for all the images from GF-2, JL-1, and CycloMedia, and for 27% of the images collected from Google Earth. As the main goal of our task is to recognize objects in aerial images by relying on visual cues, for which the geolocation of an image is insignificant for the process, DOTA-v2.0 does not provide the geolocation of its images.

4.2 Spatial Resolution Information

The ground sample distance (GSD), which indicates the distance between pixel centers measured on Earth, has potential usages. For example, it allows us to calculate the



Fig. 4. Examples of annotated images in DOTA. We show three examples per category.



Fig. 5. Typical examples of images collected from Google Earth (a), GF&JL satellite (b) and CycloMedia (c).

actual sizes of objects, which can be used to filter mislabeled or misclassified outliers since the object sizes of the same category are usually limited to a small range. The GSD can also be directly incorporated into object detectors [80] to improve the classification accuracy of categories that have less physical size variation. Furthermore, we can conduct scale normalization [81] based on the priors of the object size and GSD. In DOTA-v2.0, the GSDs of the images from GF-2, JL-1, and CycloMedia are 0.81, 0.72, and 0.1 meters per pixel, respectively. While the GSDs of the images from Google Earth range from 0.1m to 4.5m per pixel. The statistical distribution of GSDs is shown in Fig. 6. It is noted that only 30% of the images in DOTA-v2.0 have the GSD information. However, the missing of GSDs will not have a big impact on the applications that require GSDs, since a learning-based method can be used to estimate the GSD [82].

4.3 Various Instance Orientations

Objects in the overhead view images have a high diversity of orientations without the restriction of gravity. As shown

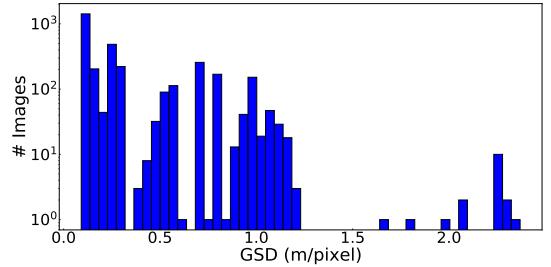


Fig. 6. The statistics of the GSD in 30% of the images in DOTA-v2.0.

in Fig. 1 (g), the objects have equal probabilities of arbitrary angles in $[-\pi, \pi]$. It is worthwhile to note that although objects in scene text detection [83] and face detection [60] also have many orientation variations, the angles of most objects lie within a narrow range (e.g., $[-\pi/2, \pi/2]$) due to gravity. The unique angle distributions of DOTA make it a good dataset for research on rotation-invariant feature extraction and oriented object detection.

4.4 Various Instances Pixel Sizes

Following the convention in [84], we use the height of an HBB to measure the pixel size of the instance. We divide all the instances in our dataset into three splits according to their heights of HBBs: small, with range from 10 to 50, medium, with range from 50 to 300, and large, with range above 300. Tab. 4 illustrates the percentages of these three instance splits in different datasets. It is clear that the PASCAL VOC dataset, NWPU VHR-10 dataset and DLR 3K Munich Vehicle dataset are dominated by medium instances or small instances.

TABLE 4

Comparison of the instance size distributions of aerial and natural images in some datasets.

Dataset	10-50 pixels	50-300 pixels	>300 pixels
PASCAL VOC [11]	0.14	0.61	0.25
MS COCO [13]	0.43	0.49	0.08
NWPU VHR-10 [42]	0.15	0.83	0.02
DLR 3K [16]	0.93	0.07	0
DOTA-v1.0 [14]	0.57	0.41	0.02
DOTA-v1.5	0.79	0.2	0.01
DOTA-v2.0	0.77	0.22	0.01

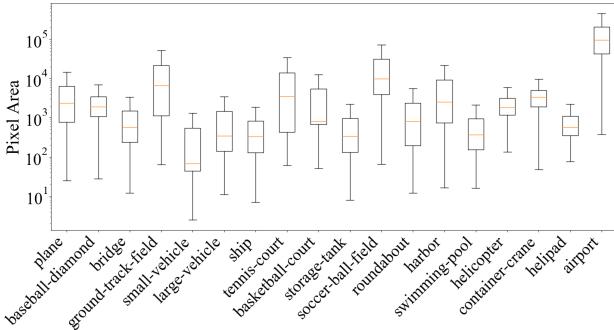


Fig. 7. Size variations for each category in DOTA. The sizes of different categories vary in different ranges.

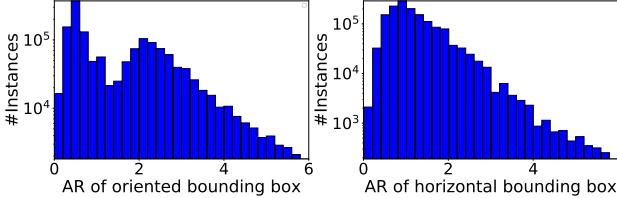


Fig. 8. AR distributions of the instances in DOTA. (a) The ARs of the OBBs. (b) The ARs of the HBBs.

MS COCO and DOTA-v1.0 have a good balance between small instances and medium instances. DOTA-v2.0 has more small instances than DOTA-v1.0. In DOTA-v2.0, some instances that are approximately 10 pixels are annotated.

In Fig. 7, we also show the distribution of instances' pixel sizes for different categories in DOTA. This figure indicates that the scales vary greatly both within and between categories. These large-scale variations among instances make the detection task more challenging.

4.5 Various Instance Aspect Ratios (ARs)

The AR is essential for anchor-based models, such as Faster R-CNN [54] and You Only Look Once (YOLOv2) [51]. We use two kinds of ARs for all the instances in our dataset to guide the model design namely, 1) the ARs of the original OBBs and 2) the AR of HBBs, which are generated by calculating the axis-aligned bounding boxes over the OBBs. Fig. 8 illustrates the distributions of these two types of aspect ratios in DOTA. We can see that instances vary significantly in aspect ratio. Moreover, many instances have a large aspect ratio in our dataset.

4.6 Various Instance Densities of the Images

The number of instances per image is an important property for object detection datasets and varies largely in DOTA. It can be very dense (up to 1000 instances per image patch), or very sparse (only one instance per image patch). We

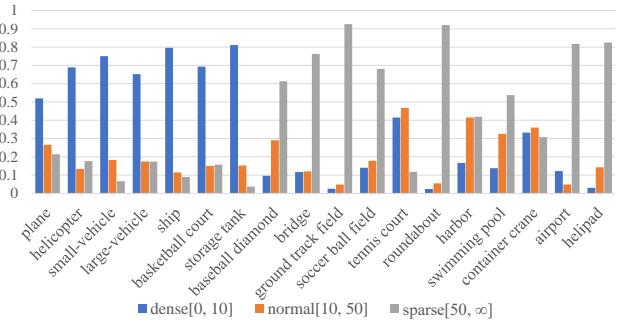


Fig. 9. Densities of the different categories. The density is measured by calculating the distance to the closest instance.

compare this property among DOTA and the general object detection datasets in Fig. 1. The number of instances per image in DOTA varies more widely than in natural image datasets.

Different categories also have different density distributions. To give a quantitative analysis, for each instance, we first measure the distance to the closest instance in the same category. We then bin the distances into three parts, dense $[0, 10]$, normal $[10, 50]$ and sparse $[50, \infty]$ (see Fig. 9). Fig. 9 shows that the storage tank, ship and small vehicle are top-3 dense categories.

4.7 DOTA Versions

It is important to notice the significant improvements from DOTA-v1.0 to DOTA-v2.0. In DOTA-v1.0, tiny objects (below 10 pixels) have not been annotated, and images are mainly from a single domain, *i.e.*, Google Earth images. Moreover, the images from DOTA-v1.0 are usually selected *areas that contain many objects* from large-size images. Although, in the past years, promising progress has been reported in oriented object detection with DOTA-v1.0, following challenging aspects can not be fully addressed by using DOTA-v1.0:

- to benchmark detection models for oriented objects both in tiny size and normal size;
- to address the object detection problem in large-scale images, *e.g.*, images with size larger than $20,000 \times 20,000$ pixels, that only contain a few objects;
- to develop robust oriented object detection models with strong generalization capability for multi-source overhead images.

To address these problems, DOTA-v1.5 added the annotation of the tiny objects in DOTA-v1.0. DOTA-v2.0 further collected many more large-size GF-2 and airborne images, which have a lower foreground ratio, approaching the object distribution in real-world applications, as shown in Tab. 3. The number of objects for each category and the dataset split for three versions of DOTA are summarized in Tab. 5.

4.7.1 DOTA-v1.0

DOTA-v1.0 contains 15 common categories, 2,806 images and 188,282 instances. The proportions of the training set, validation set, and testing set in DOTA-v1.0 are 1/2, 1/6, and 1/3, respectively.

TABLE 5

Comparisons of the three versions of DOTA. We count the number of instances for each category and dataset split.

	DOTA-v1.0	DOTA-v1.5	DOTA-v2.0
Plane	14,085	14,978	23,930
BD	1,130	1,127	3,834
Bridge	3,760	3,804	21,433
GTF	678	689	4,933
SV	48,891	242,276	1,235,658
LV	31,613	39,249	89,353
Ship	52,516	62,258	251,883
TC	4,654	4,716	9,396
BC	954	988	3,556
ST	11,794	12,249	79,497
SBF	720	727	2,404
RA	871	929	6,809
Harbor	12,287	12,377	29,581
SP	3,507	4,652	20,095
HC	822	833	893
CC	0	237	3,887
Airport	0	0	5,905
Helipad	0	0	611
Total	188,282	402,089	1,793,658
Training	98,990	210,631	268,627
Validation	28,853	69,565	81,048
Test/Test-dev	60,439	121,893	353,346
Test-challenge	0	0	1,090,637

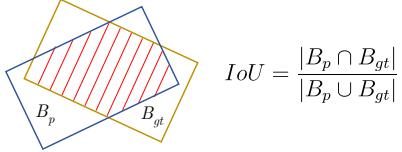


Fig. 10. The computation of the IoU between two OBBs.

4.7.2 DOTA-v1.5

DOTA-v1.5 uses the same images as DOTA-v1.0, but extremely small instances (less than 10 pixels) are also annotated. Moreover, a new category, “container crane” containing 402,089 instances in total is added. The number of images and dataset splits are the same as those in DOTA-v1.0.

4.7.3 DOTA-v2.0

There are 18 common categories, 11,268 images and 1,793,658 instances in DOTA-v2.0. Compared to DOTA-v1.5, it further adds the new categories of “airport” and “helipad”. DOTA-v2.0 are split into *training*, *validation*, *test-dev*, and *test-challenge* subsets. To avoid the problem of overfitting, the proportion of the training and validation sets is smaller than that of the test set. Furthermore, we have two test subsets, namely *test-dev* and *test-challenge*, which are similar to the MS COCO dataset [13].

For the test-dev subset and the test-challenge subset, we release the images without annotations. For evaluation, one can submit the results to the evaluation server⁵. All the DOTA-v2.0 experiments in this paper are evaluated on test-dev.

5 BENCHMARKS

5.1 Evaluation Tasks and Metrics

The task of object detection is to locate and classify the instances in images. We use two location representations

(**HBB** and **OBB**) in our paper. The HBB is a rectangle (x, y, w, h) , and the OBB is a quadrilateral $\{(x_i, y_i)|i = 1, 2, 3, 4\}$. Then, there are two tasks, detection with HBB and detection with OBB. To be more specific, we evaluate these methods on two kinds of ground truths: HBB and OBB ground truths. For the two tasks, each detected bounding box has a corresponding confidence score. We adopt the PASCAL VOC 07 metric [11] for the calculation of the mean average precision (mAP). Average Precision (AP) computes the average precision value for recall value over 0 to 1 (*i.e.*, the area under precision/recall curve). The mean Average Precision (mAP) is the average of AP over all classes. The detailed computation of the precision and recall can refer to [11]. The intersection over union (IoU) is crucial in determining true positives and false positives, which are required to compute precision and recall. It is worthwhile to note that for the OBB task, the intersection over union (IoU) is calculated between OBBs, as shown in Fig. 10. The two OBBs (B_p and B_{gt}), and the intersection between OBBs ($B_p \cap B_{gt}$) are all convex polygons, whose area can be easily computed⁶. The union area of two OBBs can be calculated as $|B_p \cup B_{gt}| = |B_p| + |B_{gt}| - |B_p \cap B_{gt}|$. The code for the mAP and IoU computation between OBBs can be found in our development kit.

5.2 Implementation Details

In the previous benchmarks [14], the algorithms were implemented with different codes and settings, which makes these algorithms hard to compare in DOTA. To this end, we implement and evaluate all the algorithms in one unified code library modified from MMDetection [28].

Since large images cannot be directly fed to CNN-based detectors due to the memory limitations, we crop a series of $1,024 \times 1,024$ patches from the original images with a stride set to 824 (different from the previous stride of 512 [14]). During inference, we first send the patches (same settings as training) to obtain temporary results. Then we map the detected results from the patch coordinates to the original image coordinates. Finally, we apply NMS on these results in the original image coordinates. We set the threshold of NMS to 0.3 for the HBB experiments and 0.1 for the OBB experiments. For multi-scale training and testing, we first scale the original images to $[0.5, 1.0, 1.5]$ and then crop the images into patches of size $1,024 \times 1,024$ and a stride of 824. We use 4 GPUs for training with a total batch size of 8 (2 images per GPU). The learning rate is set to 0.01. Except for RetinaNet [85], which adopts the “ $2\times$ ” schedule, the other algorithms adopt the “ $1\times$ ” [29] training schedule. We set the number of proposals and maximum number of predictions per image patch to 2,000 for all the experiments except when otherwise mentioned. The other hyperparameters follow those of Detectron [29].

5.2.1 Baselines with HBBs

We use two ways to build baselines for the HBB task. The first way directly predicts the HBB results, while the second way first predicts the OBB results and then converts

6. A polygon can be decomposed into a group of non-overlapping triangles. The area of the convex polygon is equal to the sum of all the triangular areas.

5. <https://captain-whu.github.io/DOTA/evaluation.html>

OBBs to HBBs. To directly predict the HBB results, we use RetinaNet [85], Mask R-CNN, Cascade Mask R-CNN, Hybrid Task Cascade and Faster R-CNN [54] as baselines. For the OBB predictions, we will introduce the methods in the following section.

5.2.2 Baselines with OBBs

Most of the state-of-the-art object detection methods are not designed for oriented objects. To enable these methods to predict OBBs, we build the baselines in two ways. The first is to change HBB head to OBB Head, which regresses the offsets of OBBs relative to the HBBs. The second is Mask Head, which considers the OBBs to a coarse mask and predicts the pixel-level classification from each RoI.

OB Head To predict OBB, the previous Faster R-CNN OBB [14] and Textboxes++ [62] modified RoI Head of Faster R-CNN and the Anchor Head of the single-shot detector (SSD), respectively, to regress quadrangles. In this paper, we use the representation (x, y, w, h, θ) instead of $\{(x_i, y_i) | i = 1, 2, 3, 4\}$ for OBB regression. More precisely, rectangular RoIs (anchors) can be written as $\mathcal{B} = (x_{min}, y_{min}, x_{max}, y_{max})$. We can also consider it a special OBB and rewrite it as $\mathcal{R} = (x, y, w, h, \theta)$. For matching, IoUs are calculated between the horizontal RoIs (anchors) and HBBs of the ground truths for computational simplicity. Each OBB, it has four forms: $\mathcal{G} = \{gt_i | i = 1, 2, 3, 4\}$, where $gt_1 = (x_g, y_g, w_g, h_g, \theta_g)$, $gt_2 = (x_g, y_g, w_g, h_g, \theta_g + \pi)$, $gt_3 = (x_g, y_g, h_g, w_g, \theta_g)$, and $gt_4 = (x_g, y_g, h_g, w_g, \theta_g + \pi)$. Before calculating the targets, we choose the best matched ground-truth form. The index of the best matched form is calculated by $\xi = \arg \min_i \mathcal{D}(\mathcal{R}, gt_i)$, where \mathcal{D} is a distance function, which could be Euclidean distance or another distance function. We denote the best matched form by $gt_\xi = (x_b, y_b, w_b, h_b, \theta_b)$. Then the learning target is calculated as

$$\begin{aligned} t_x &= (x_b - x)/w, \quad t_y = (y_b - y)/h, \\ t_w &= \log(w_b/w), \quad t_h = \log(h_b/h), \\ t_\theta &= \theta_b - \theta \end{aligned} \quad (1)$$

We then simply replace the HBB RoI Head of *Faster R-CNN* and anchor head of *RetinaNet* with OBB Head and obtain two models, called *Faster R-CNN OBB* and *RetinaNet OBB*. We also modify the *Faster R-CNN* to predict both the HBB and OBB in parallel, which is similar to *Mask R-CNN* [86]. We call this model *Faster R-CNN H-OBB*. We further evaluate the deformable RoI pooling (Dpool) and RoI Transformer by replacing the RoI Align in *Faster R-CNN OBB*. Then we have two models: *Faster R-CNN OBB + Dpool* and *Faster R-CNN OBB + RoI Transformer*. Note that the RoI Transformer used here is slightly different from the original one. The original RoI Transformer uses the Light Head R-CNN [87] as the base detector while we use *Faster R-CNN*.

Mask Head *Mask R-CNN* [86] was originally used for instance segmentation. Although DOTA does not have pixel-level annotation for each instance, the OBB annotations can be considered coarse pixel-level annotations, so we can apply *Mask R-CNN* [86] to DOTA. During inference, we calculate the minimum OBBs that contain the predicted masks. The original *Mask R-CNN* [86] only applies a mask head to the top 100 HBBs in terms of the score. Due to the

large number of instances per image, as illustrated in Fig. 2, we apply a mask head to all the HBBs after NMS. In this way, we evaluate *Mask R-CNN* [86], *Cascade Mask R-CNN* and *Hybrid Task Cascade* [88].

5.3 Codebase and Development Kit

We also provide an aerial object detection code library⁷ and a development kit⁸ for using DOTA. To construct the comprehensive baselines, we select MMDetection as the fundamental code library since it contains rich object detection algorithms and has the feature of modular design. However, the original MMDetection [28] lacks the modules to support oriented object detection. Therefore, we enriched MMDetection with **OB Head** as described in Sec. 5.2.2 to enable OBB predictions. We also implemented modules such as rotated RoI Align and rotated position-sensitive RoI Align for rotated region feature extraction, which are crucial components in algorithms such as rotated region proposal network (RRPN) [89] and RoI Transformer [18]. These new modules are compatible with the modularly designed MMDetection, so we can easily create new algorithms for oriented object detection not restricted to the baseline methods in this paper. We also provide a development kit containing necessary functions for object detection in DOTA, including:

- *Loading and visualizing the ground truths.*
- *Calculating the IoU between OBBs*, which is implemented in a mixture of Python/C program. We provide both the CPU and GPU versions.
- *Evaluating the results*. The evaluation metrics are described in Sec. 5.1.
- *Cropping and merging images*. When using the large-size images in DOTA, one can utilize this tool kit to split an original image into patches. After testing on the patches, one can use the tools to map the results of patches back to the original image coordinates and apply NMS.

6 RESULTS

6.1 Benchmark Results and Analyses

In this section, we conduct a comprehensive evaluation of over 70 experiments and analyze the results. First, we demonstrate the baseline results of 10 algorithms on DOTA-v.10, DOTA-v1.5 and DOTA-v2.0. The baselines cover both two-stage and one-stage algorithms. For most algorithms, we report the mAPs of HBB and OBB predictions, respectively, except for *RetinaNet* and *Faster R-CNN*, since they do not support oriented object detection. For algorithms with only OBB heads (*RetinaNet OBB*, *Faster R-CNN OBB*, *Faster R-CNN OBB + DPool*, *Faster R-CNN OBB + RoI Transformer*), we obtain their HBB results by transferring from OBB as described in Sec. 5.2.1. For algorithms with both HBB and OBB heads (*Mask R-CNN*, *Cascade Mask R-CNN*, *Hybrid Task Cascade*^{*}, and *Faster R-CNN H-OBB*), the HBB mAP is the maximum of the predicted HBB mAP and the transferred HBB mAP. It can be seen that the OBB mAP

7. <https://github.com/dingjiansw101/AerialDetection>

8. https://github.com/CAPTAIN-WHU/DOTA_devkit

TABLE 6

Baseline results on DOTA. For the evaluation of DOTA-v2.0, we use the DOTA-v2.0 test-dev set. The implementation details are described in Sec. 5.2. All the algorithms in this table adopt the ResNet-50 with an FPN as backbone. The speed refers to the inference speed, which is reported for a single NVIDIA Tesla V100 GPU on DOTA-v2.0 test-dev. The image size is $1,024 \times 1,024$. Hybrid Task Cascade* means that the semantic branch is not used since there are no semantic annotations in DOTA.

method	speed (fps)	DOTA-v1.0		DOTA-v1.5		DOTA-v2.0	
		HBB mAP	OBB mAP	HBB mAP	OBB mAP	HBB mAP	OBB mAP
RetinaNet [85]	16.7	67.45	-	61.64	-	49.31	-
RetinaNet OBB [85]	12.1	69.05	66.28	62.49	59.16	49.26	46.68
Mask R-CNN [86]	9.7	71.61	70.71	64.54	62.67	51.16	49.47
Cascade Mask R-CNN [88]	7.2	71.36	70.96	64.31	63.41	50.98	50.04
Hybrid Task Cascade* [88]	7.9	72.49	71.21	64.47	63.40	50.88	50.34
Faster R-CNN [54]	14.3	70.76	-	64.16	-	50.71	-
Faster R-CNN OBB [14]	14.1	71.91	69.36	63.85	62.00	49.37	47.31
Faster R-CNN OBB + Dpool [55]	12.1	71.83	70.14	63.67	62.20	50.48	48.77
Faster R-CNN H-OBB [14]	13.7	70.37	70.11	64.43	62.57	50.38	48.90
Faster R-CNN OBB + RoI Transformer [18]	12.4	74.59	73.76	66.09	65.03	53.37	52.81

TABLE 7

Baseline results of class-wise AP on DOTA-v1.0. The abbreviations of algorithms are: Mask R-CNN (MR), CMR-Cascade Mask R-CNN (CMR), Hybrid Task Cascade without a semantic branch (HTC*), Faster R-CNN (FR), Deformable RoI Pooling (Dp) and RoI Transformer (RT). The short names for categories are defined as: BD—Baseball diamond, GTF—Ground field track, SV—Small vehicle, LV—Large vehicle, TC—Tennis court, BC—Basketball court, SC—Storage tank, SBF—Soccer-ball field, RA—Roundabout, SP—Swimming pool, HC—Helicopter.

OBB Results																
method	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	mAP
RetinaNet [85]	86.54	77.45	42.8	64.87	71.06	58.5	73.53	90.72	80.97	66.67	52.42	62.16	60.79	64.84	40.83	66.28
MR [86]	88.7	74.13	50.75	63.66	73.64	73.98	83.68	89.74	78.92	80.26	47.43	65.09	64.79	66.09	59.79	70.71
CMR [88]	88.93	75.21	51.55	64.9	74.39	75.37	84.74	90.23	77.48	81.51	46.57	63.49	65.39	67.63	56.96	70.96
HTC* [88]	89.17	75.05	51.95	64.5	74.19	76.3	86.05	90.55	79.51	77.18	50.3	61.23	65.89	68.29	58.01	71.21
FR OBB [14]	88.42	74.24	45.31	61.49	73.53	70.03	77.76	90.87	81.8	82.64	48.75	60.14	63.42	67.65	54.31	69.36
FR OBB + Dp [55]	88.82	74.12	45.44	63.07	73.13	73.59	84.39	90.71	82.28	83.59	42.76	58.49	63.52	68.25	59.89	70.14
FR H-OBB [14]	88.41	79.35	45.39	63.16	73.91	72.11	83.86	90.25	77.34	81.04	48.5	60.53	63.88	66.43	57.53	70.11
FR OBB + RT [18]	88.34	77.07	51.63	69.62	77.45	77.15	87.11	90.75	84.9	83.14	52.95	63.75	74.45	68.82	59.24	73.76
HBB Results																
method	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	mAP
RetinaNet [85]	88.28	77.76	47.47	59.07	73.83	63.49	77.69	90.43	78.57	65.87	48.67	61.82	68.92	71.59	38.22	67.45
RetinaNet OBB [85]	88.51	78.21	47.9	64.46	75.17	72.9	78.5	90.72	82.22	67.11	51.33	62.16	69.47	69.79	37.3	69.05
MR [86]	88.79	79.06	53.04	63.14	78.22	65.23	77.94	89.6	81.99	81.06	47.17	65.19	72.72	68.67	62.27	71.61
CMR [88]	88.93	81.5	52.85	64.01	79.18	65.85	78.12	90.08	77.48	81.87	46.45	62.88	72.99	69.95	58.32	71.36
HTC* [88]	89.11	79.76	53.87	64.4	79.06	76.23	86.57	90.5	79.51	81.66	50.44	63.75	73.33	70.61	48.59	72.49
FR [54]	89.02	75.8	53.47	60.8	78.02	65.56	78.01	90.1	77.3	81.7	47.12	61.48	72.59	70.89	59.54	70.76
FR OBB [14]	88.37	75.45	52.11	59.98	78.08	73.42	85.65	90.81	83.22	83.18	51.45	60.01	72.44	71.09	53.44	71.91
FR OBB + Dp [55]	88.86	77.4	51.6	63.12	77.62	74.73	86.2	90.72	82.73	83.75	44.35	58.81	71.97	70.98	54.68	71.83
FR H-OBB [14]	88.67	78.95	52.63	57.34	78.55	65.22	78.08	90.69	82.01	82.06	46.08	61.46	72.43	70.34	51.12	70.37
FR OBB + RT [18]	88.47	81.0	54.1	69.19	78.42	81.16	87.35	90.75	84.9	83.55	52.63	62.97	75.89	71.31	57.22	74.59

TABLE 8

Baseline results of class-wise AP on DOTA-v1.5. The abbreviations of algorithms are: Mask R-CNN (MR), CMR-Cascade Mask R-CNN (CMR), Hybrid Task Cascade without a semantic branch (HTC*), Faster R-CNN (FR), Deformable RoI Pooling (Dp) and RoI Transformer (RT). The short names for categories are defined as: BD—Baseball diamond, GTF—Ground field track, SV—Small vehicle, LV—Large vehicle, TC—Tennis court, BC—Basketball court, SC—Storage tank, SBF—Soccer-ball field, RA—Roundabout, SP—Swimming pool, HC—Helicopter, CC—Container Crane.

OBB Results																	
method	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	CC	mAP
RetinaNet [85]	71.43	77.64	42.12	64.65	44.53	56.79	73.31	90.84	76.02	59.96	46.95	69.24	59.65	64.52	48.06	0.83	59.16
MR [86]	76.84	73.51	49.9	57.8	51.31	71.34	79.75	90.46	74.21	66.07	46.21	70.61	63.07	64.46	57.81	9.42	62.67
CMR [88]	77.77	74.62	51.09	63.44	51.64	72.9	79.99	90.35	74.9	67.58	49.54	72.85	64.19	64.88	55.87	3.02	63.41
HTC* [88]	77.8	73.67	51.4	63.99	51.54	73.31	80.31	90.48	75.12	67.34	48.51	70.63	64.84	64.48	55.87	5.15	63.4
FR OBB [14]	71.89	74.47	44.45	59.87	51.28	68.98	79.37	90.78	77.38	67.5	47.75	69.72	61.22	65.28	60.47	1.54	62.0
FR OBB + Dp [55]	71.79	73.61	44.76	61.99	51.34	70.04	79.67	90.78	76.58	67.73	44.58	70.51	61.8	65.49	64.35	0.15	62.2
FR H-OBB [14]	71.57	74.71	46.39	63.4	51.54	70.11	79.09	90.63	76.81	67.4	48.66	70.9	63.1	65.67	56.66	4.55	62.57
FR OBB + RT [18]	71.92	76.07	51.87	69.24	52.05	75.18	80.72	90.53	78.58	68.26	49.18	71.74	67.51	65.53	62.16	9.99	65.03
HBB Results																	
method	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	CC	mAP
RetinaNet [85]	74.05	77.75	48.75	59.94	49.23	61.43	77.31	90.38	75.46	61.19	47.29	69.99	67.99	74.15	40.88	10.4	61.64
RetinaNet OBB [85]	71.66	77.22	48.71	65.16	49.48	69.64	79.21	90.84	77.21	61.03	47.3	68.69	67.22	74.48	46.16	5.78	62.49
MR [86]	78.36	77.41	53.36	56.94	52.17	63.6	79.74	90.31	74.28	66.41	45.49	71.32	70.77	73.87	61.49	17.11	64.54
CMR [88]	78.61	75.43	54.0	63.76	52.55	63.93	79.88	90.06	75.05	67.83	45.76	72.48	72.1	74.11	53.9	9.59	64.31
HTC* [88]	78.41	74.41	53.41	63.17	52.45	63.56	79.89	90.34	75.17	67.64	48.44	69.94	72.13	74.02	56.42	12.14	64.47
FR [54]	71.88	74.06	52.69	62.35	52.08	63.22	79.69	90.55	76.91	66.86	47.84	70.72	70.61	68.55	63.34	15.26	64.16
FR OBB [14]	71.91	71.6	50.58	61.95	51.99	71.05	80.16	90.78	77.16	67.66	47.93	69.35	69.51	74.4	60.33	5.17	63.85
FR OBB + Dp [55]	71.9	72.8	50.84	61.99	51.97	72.15	80.13	90.74	76.53	67.95	45.09	69.93	70.43	74.72	58.28	3.19	63.67
FR H-OBB [14]	71.6	74.14	52.69	62.85	52.13	71.32	79.75	90.64	76.29	67.52	49.31	71.27	72.11	73.91	55.11	10.26	64.43
FR OBB + RT [18]	71.92	75.21	54.09	68.1	52.54	74.87	80.79	90.46	78.58	68.41	51.57	71.48	74.91	74.84	56.66	13.01	66.09

TABLE 9

Baseline results of class-wise AP on DOTA-v2.0. The abbreviations of algorithms are: Mask R-CNN (MR), CMR-Cascade Mask R-CNN (CMR), Hybrid Task Cascade without a semantic branch (HTC*), Faster R-CNN (FR), Deformable RoI Pooling (Dp) and RoI Transformer (RT). The short names for categories are defined as: *BD*—Baseball diamond, *GTF*—Ground field track, *SV*—Small vehicle, *LV*—Large vehicle, *TC*—Tennis court, *BC*—Basketball court, *SC*—Storage tank, *SBF*—Soccer-ball field, *RA*—Roundabout, *SP*—Swimming pool, *HC*—Helicopter, *CC*—Container Crane, *Air*—Airport, *Heli*—HeliPad.

method	OBB Results																		
	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	CC	Air	Heli	mAP
RetinaNet [85]	70.63	47.26	39.12	55.02	38.1	40.52	47.16	77.74	56.86	52.12	37.22	51.75	44.15	53.19	51.06	6.58	64.28	7.45	46.68
MR [86]	76.2	49.91	41.61	60.0	41.08	50.77	56.24	78.01	55.85	57.48	36.62	51.67	47.39	55.79	59.06	3.64	60.26	8.95	49.47
CMR [88]	77.01	47.54	41.79	58.02	41.58	51.74	57.86	78.2	56.75	58.5	37.89	51.23	49.38	55.98	54.59	12.31	67.33	3.01	50.04
HTC* [88]	77.69	47.25	41.15	60.71	41.77	52.79	58.87	78.74	55.22	58.49	38.57	52.48	49.58	56.18	54.09	4.2	66.38	11.92	50.34
FR OBB [14]	71.61	47.2	39.28	58.7	35.55	48.88	51.51	78.97	58.36	58.55	36.11	51.73	43.57	55.33	57.07	3.51	52.94	2.79	47.31
FR OBB + Dp [55]	71.55	49.74	40.34	60.4	40.74	50.67	55.56	79.03	58.22	58.24	34.73	51.95	44.33	55.1	53.14	7.21	59.53	6.38	48.77
FR H-OBB [14]	71.39	47.59	39.82	59.01	41.51	49.88	57.17	78.36	56.87	58.24	37.66	51.86	44.61	55.49	54.74	7.56	61.88	6.6	48.9
FR OBB + RT [18]	71.81	48.39	45.88	64.02	42.09	54.39	59.92	82.7	63.29	58.71	41.04	52.82	53.32	56.18	57.94	25.71	63.72	8.7	52.81
method	HBB Results																		
	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	CC	Air	Heli	mAP
RetinaNet [85]	71.86	48.69	42.2	53.12	41.16	45.64	55.9	77.74	56.14	52.0	37.68	51.46	53.27	57.51	46.76	15.66	67.76	12.97	49.31
RetinaNet OBB [85]	70.99	46.77	43.76	55.08	41.55	51.06	58.01	77.78	57.72	53.5	37.66	51.72	53.56	57.32	49.18	12.01	64.53	4.48	49.26
MR [86]	77.61	51.35	44.89	60.12	42.51	48.1	57.93	77.84	57.55	57.88	36.53	51.71	54.79	58.93	60.01	14.42	60.32	8.43	51.16
CMR [88]	78.12	47.89	46.43	57.8	42.97	48.23	59.11	78.19	57.17	58.88	37.42	51.32	53.66	58.07	55.5	17.67	67.37	1.81	50.98
HTC* [88]	78.28	47.95	45.7	59.95	42.97	48.7	59.14	78.58	55.91	58.77	37.75	52.46	53.34	58.64	55.53	9.78	66.67	5.74	50.88
FR [54]	76.14	49.93	44.97	57.8	42.4	47.86	57.76	77.7	56.57	58.65	39.24	52.6	54.94	58.92	56.62	12.88	61.64	6.24	50.71
FR OBB [14]	71.68	45.8	45.56	58.7	42.18	51.28	59.28	79.01	58.74	58.75	37.26	51.93	52.36	58.08	54.12	8.48	53.01	2.4	49.37
FR OBB + Dp [55]	71.58	47.68	46.16	60.48	42.34	52.55	59.48	79.07	59.61	58.46	35.35	53.73	53.12	58.33	52.06	12.56	59.76	6.38	50.48
FR H-OBB [14]	77.14	50.54	45.6	57.53	42.27	48.09	57.6	78.4	59.78	57.8	36.64	52.13	52.51	58.42	48.91	14.99	60.0	8.47	50.38
FR OBB + RT [18]	71.84	48.2	47.84	63.94	42.97	54.79	60.74	82.88	63.51	58.89	40.63	52.83	55.7	58.87	57.94	27.04	64.27	7.68	53.37

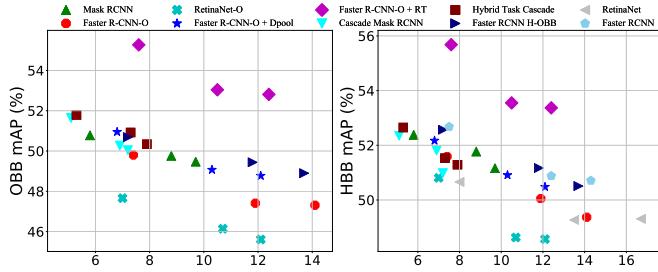


Fig. 11. Results of using different backbones. The algorithms are tested on DOTA-2.0 test-dev. For each algorithm, we choose 3 different backbones, i.e., ResNet-50 with an FPN, ResNet-101 with an FPN, and 64×4d ResNeXt-101 with an FPN. Faster R-CNN-O is the Faster R-CNN OBB in this work. RetinaNet-O stands for RetinaNet OBB. Dpool means the Deformable RoI Pooling, and RT means the RoI Transformer. The speeds are tested on a single Tesla V100.

is usually slightly lower than the HBB mAP for the same algorithm since the OBB task needs a more precise location than the HBB task.

Tab. 6 shows that the performance on DOTA-v1.0, DOTA-v1.5 and DOTA-v2.0 are declining, indicating the increased difficulty of the datasets. The class-wise AP are reported in Tab. 7, Tab. 8 and Tab. 9. To give more detailed comparisons of speed *vs.* accuracy, we evaluate all algorithms using different backbones (as shown in Fig. 11). From the speed-accuracy curve, the Faster R-CNN OBB + RoI Transformer outperforms the other methods. To explore the properties of DOTA and provide guidelines for future research, we evaluate the module design and the hyperparameter setting. Then, we analyze the influence of data augmentation in detail. Finally, we visualize the results to show the difficulties of ODAI.

6.1.1 Mask Head vs. OBB Head

The OBB head tackles oriented object detection as a regression problem, while the mask head tackles oriented object detection as a pixel-level classification problem. The mask head more easily converges and achieves better results but is more computationally expensive. Taking the results on the

DOTA-v2.0 test-dev set as an example, Mask R-CNN still outperforms Faster R-CNN H-OBB by 0.57 points in OBB mAP. Nevertheless, Mask R-CNN is slower than Faster R-CNN H-OBB by 4 fps. Note that the process of transferring the mask to the OBB is not considered here. Otherwise, Mask R-CNN should be slower.

6.1.2 RoI Transformer vs. Deformable RoI Pooling

Geometric variations are still challenging in object detection. In this part, we evaluate RoI Transformer and Dpool by replacing RoI Align in Faster R-CNN OBB. We call these two models Faster R-CNN OBB + RoI Transformer and Faster R-CNN OBB + Dpool. Tab. 6 and Fig. 11 show that Dpool improves the performance of Faster R-CNN OBB at most times, while RoI Transformer performs better than Dpool. This finding verifies that carefully designed geometry transformation modules such as RoI Transformer are better than general geometry transformation modules such as Dpool for aerial images.

6.1.3 Excluding Small Instances

During the training on DOTA-v1.5 and DOTA-v2.0, many extremely small instances will cause numerical instability. For the experiments in DOTA-v1.5 and DOTA-v2.0, we set a threshold to exclude too small instances. We try to explore the influence of different thresholds on DOTA-v2.0. We filter the small instances by two rules: 1) the area of instance is below a certain threshold, and 2) $\max(w, h)$ is below a threshold, where the w and h are the width and height, respectively, of the corresponding HBB. The results in Tab. 10 show that small instances have little influence on the results.

6.1.4 Number of Proposals

The number of proposals is an important hyperparameter in modern detectors. As mentioned before, the possible number of instances in aerial images is quite different from that in natural images. In DOTA, one $1,024 \times 1,024$ image

TABLE 10

Results after excluding extremely small instances by different thresholds in DOTA-v2.0. There are 642,601 instances before filtering.

# of filtered Instance	Filtering strategy	HBB mAP
99,317	area \leq 50 and max(w, h) \leq 10	51.08
157,287	area \leq 80 and max(w, h) \leq 10	51.35
158,629	area \leq 80 and max(w, h) \leq 12	50.71

may contain more than 1,000 instances. There is no doubt that the parameters that perform well for natural images are not optimal for aerial images. Here we explore the optimal settings for aerial images. As shown in Tab. 11, the number of proposals with the highest performance for Faster R-CNN OBB + RoI Transformer is 8,000. For Faster R-CNN OBB, the increase in the mAP slows at approximately 8,000 proposals. Furthermore, from 1,000 to 10,000 proposals, the improvements in Faster R-CNN + RoI Transformer and Faster R-CNN OBB are 2.2 and 1.39 points in mAP, respectively. However, the increased number of proposals bring more computation. Therefore, for the other experiments in this paper, we choose 2,000 proposals. The optimal number of proposals in DOTA is quite a bit larger than that in PASCAL VOC, where 300 is the optimal number. This finding confirms that the difference between aerial and natural images is again significant.

6.1.5 Data Augmentation

In this section, we explore the influence of data augmentation in detail. We followed the multi-scale training, testing, and rotation training strategies in [25] and further conduct rotation testing. Note that as data augmentation often produces a huge number of patches and will dramatically increase the time complexity of experiments, we conduct our ablation study of data augmentation on DOTA-v1.5, which is similar to DOTA-v2.0 in data distribution while much smaller. The model we select is Faster R-CNN OBB + RoI Transformer. We choose R-50-FPN as the backbone and adopt five data augmentation strategies. The first is the high patch overlap. We change the overlap between patches from 200 to 512 since the large instances may be cut off at the edge. The second and third are multi-scale training and testing, respectively. We resize the original images by factors of [0.5, 1.0, 1.5] and then crop the original images into patches of size $1,024 \times 1,024$. The fourth is the rotation training. For images with roundabouts and storage tanks, we rotate the patches randomly by four angles $[\pi/2, \pi, -\pi/2, -\pi]$. For images with the other categories, we rotate the angle randomly from $[-\pi, \pi]$ continuously during training. We also rotate the images at four angles $([0, \pi/2, \pi, 3\pi/2])$ during testing. When performing test time augmentation, the results from images at different angles and scales are merged through the Non-Maximum Suppression (NMS) process in the original image coordinates. As shown in Tab. 12, both scale and rotation data augmentations improve the performance of object detection by a large margin, which is consistent with the large scale and orientation variations in DOTA. Furthermore, this baseline model already used a feature pyramid network (FPN) and RoI Transformer. This indicates that the FPN and RoI Transformer do not completely solve the problem of scale and rotation variations, and geometric modeling with CNNs is still an open problem.

6.1.6 Class-Wise Results

The baseline results of class-wise AP on DOTA-v1.0, DOTA-v1.5, and DOTA-v2.0 are reported in Tab. 7, Tab. 8 and Tab. 9. In contrast with DOTA-v1.0, DOTA-v1.5 additionally annotated the tiny objects (most of them are small vehicles below 10 pixels). Therefore, by comparing the AP of small vehicles of the same detector on DOTA-v1.0 and DOTA-v1.5, we can see the challenges in the detection of tiny objects. Taking Faster R-CNN OBB with RoI Transformer as an example, the AP on small vehicles decrease by **25.4 points** from **77.45** to **52.05**. The challenge on tiny object detection can also be checked in the last row of Fig. 12. The advantage of OBBs over HBBs in detecting the densely packed objects can be demonstrated by comparing the mAP of OBB detectors and HBB detectors. For example, Faster R-CNN **OBB** outperforms Faster R-CNN by **8 points** in AP on large vehicles in DOTA-v1.0. Some examples of the detection of densely packed objects are shown in the first row of Fig. 12.

6.1.7 Visualization of the Results

We show the performance of Faster R-CNN [54], Faster R-CNN OBB, RetinaNet OBB, Mask R-CNN and Faster R-CNN OBB + RoI Transformer on difficult scenes in Fig. 12: 1) The first row demonstrates densely packed large vehicles. Faster R-CNN misses many instances due to the high overlaps between neighboring large vehicles in the HBBs. Those large vehicles are suppressed through NMS. Faster R-CNN OBB, Mask R-CNN and Faster R-CNN OBB + RT perform well, while RetinaNet OBB has lower location precision due to feature misalignment. 2) The second and third rows show long shape instances with a large ARs. These instances are self-similar, which means that each part of the instance has a similar feature as the whole instance. For example, the second row shows that all methods have at least two predictions on a single bridge. The third row also reveals this problem. There exist several predictions on a single ship. 3) The second and third rows also indicate that several different categories have very similar features. Bridges are easily classified as airports and harbors while the ships are easily to be classified as harbors and bridges. 4) The last row shows the difficulty of detecting extremely small instances (less than or approximately 10 pixels). The recall of the extremely small instances is very low.

6.2 State-of-the-Art Results on DOTA-v1.0

In this section, we compare the performance of Faster R-CNN OBB + RoI Transformer with the state-of-the-art algorithms on DOTA-v1.0 [14]. As shown in Tab. 13, Faster R-CNN OBB + RoI Transformer achieves an OBB mAP of 73.76 for DOTA-v1.0, and it outperforms all the previous state-of-the-art methods except that proposed by Li et al. [25]. Note that the method of Li et al. [25], SCRDet [24] and the image cascade network (ICN) [19] all use multiple scales for training and testing to achieve high performance. The method in [25] further used rotation data augmentation during training as described in Sec. 6.1.5. When using the same data augmentation, we achieve an mAP of 79.82. It outperforms the method in [25] by 3.46 points in OBB mAP and 1.96 points in HBB mAP. In addition, there is a significant improvement in densely packed small instances. (e.g.,

TABLE 11

Results using different number of proposals on DOTA-v2.0 test-dev. The speeds are tested on a single Tesla V100 GPU. The other settings are the same with those in Tab. 6.

Method	# of proposals	1,000	2,000	3,000	4,000	5,000	6,000	7,000	8,000	9,000	10,000
Faster R-CNN OBB + ROI Transformer	OBB mAP (%)	51.72	52.81	52.81	53.24	53.29	53.51	53.70	53.94	53.93	53.92
	HBB* mAP	52.56	53.37	53.37	54.63	54.86	55.07	55.08	55.09	55.08	55.06
	speed (fps)	14.4	12.4	12.2	9.1	8.7	7.8	7.5	6.5	6	5.7
Faster R-CNN OBB	OBB mAP (%)	47.10	47.31	48.03	48.09	48.32	48.35	48.48	48.49	48.49	48.49
	HBB* mAP	48.44	49.37	49.46	49.71	49.74	50.09	50.37	50.39	50.38	50.47
	speed (fps)	15.8	14.1	12.5	11.9	10.9	9.9	9.3	9.1	8.4	7.8

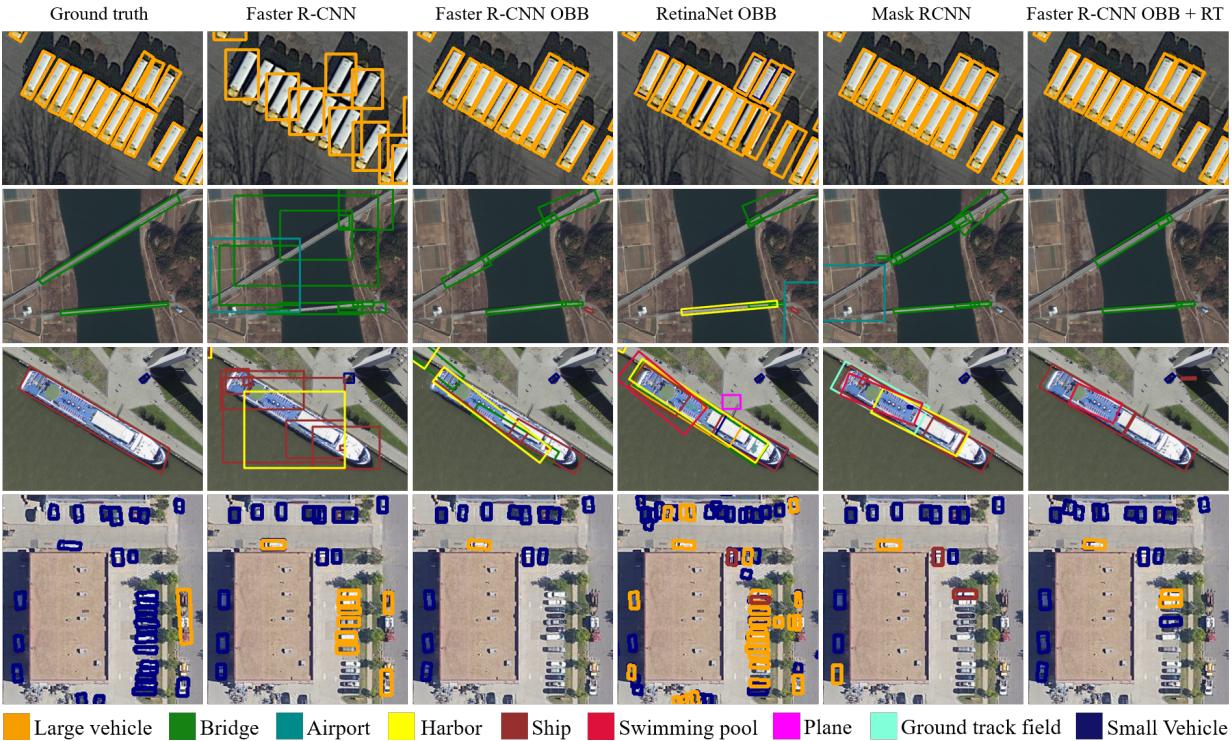


Fig. 12. **Visualization of the results on DOTA-v2.0 test-dev.** The five models are from the DOTA-v2.0 models in Tab. 6. The predictions with scores above 0.1 are shown. The results illustrate the performance in cases of orientation variations, density variations, large ARs and small ARs.

TABLE 12

Data augmentation experiments on DOTA-v1.5. Each column in this table indicates an experiment configuration. The first column represents our baseline method without additional data augmentations, while the other columns gradually add augmentation. We use Faster R-CNN OBB + ROI Transformer as the baseline. High overlap means an overlap of 512 between patches instead of 200 as in Tab. 6.

	Baseline	Data augmentation				
High overlap		✓	✓	✓	✓	✓
Multi scale Train		✓	✓	✓	✓	✓
Multi scale Test			✓	✓	✓	
Rotation Train				✓	✓	
Rotation Test					✓	
OBB mAP	65.03	67.57	69.44	73.62	76.43	77.60
HBB mAP	66.09	67.94	70.63	74.63	77.24	78.88

the small vehicles, large vehicles, and ships). For example, the detection performance for the large vehicle category gains an improvement of 12.18 points compared to the previous results.

6.3 DOAI 2019 Challenge Results

DOTA-v1.5 was first used to hold the DOAI Challenge-2019 in conjunction with CVPR 2019⁹. There were 173 registrations in total, 13 teams submitted valid results on the OBB

Task, and 22 teams submitted valid results on the HBB Task. The detailed leaderboards for the two tasks can be found on the DOAI Challenge-2019 website¹⁰, and the top 3 results are listed in Tab. 14. Notice that most of these results have been achieved by using an ensemble of detection models, except [25] which used a single model and reported 74.9 and 77.9 in terms of mAP on the OBB and HBB tasks, respectively. Both in the training and testing phase, multi-scaling and rotation strategies were used for data augmentations. With the same settings, our single model [18] achieved 76.43 and 77.24 in terms of mAP on the OBB and HBB tasks respectively, as shown in Tab. 12, which was the best results reported on the OBB task.

7 CONCLUSION

DOAI is challenging. To advance future research, we introduce a large-scale dataset, DOTA, containing 1,793,658 instances annotated by OBBs. The DOTA statistics show that it can well represent the real world well. Then, we build a code library for both oriented and horizontal ODAI to conduct a comprehensive evaluation. We hope these experiments can act as benchmarks for fair comparisons between

9. <https://captain-whu.github.io/DOAI2019/>

10. <https://captain-whu.github.io/DOAI2019/results.html>

TABLE 13

State-of-the-art results on DOTA-v1.0 [14]. The short names for categories are defined as: *BD*—*Baseball diamond*, *GTF*—*Ground field track*, *SV*—*Small vehicle*, *LV*—*Large vehicle*, *TC*—*Tennis court*, *BC*—*Basketball court*, *SC*—*Storage tank*, *SBF*—*Soccer-ball field*, *RA*—*Roundabout*, *SP*—*Swimming pool*, and *HC*—*Helicopter*. FR-O indicates the Faster R-CNN OBB detector, which is the previous official baseline provided by DOTA-v1.0 [14]. ICN [19] is the *image cascade network*. The LR-O + RT means *Light Head R-CNN + Rol Transformer*. DR-101-FPN means *deformable ResNet-101 with an FPN*. SCRDet means *small, cluttered and rotated object detector*. R-101-SF-MDA means ResNet-101 with sampling fusion network (SF-Net) and multi-dimensional attention network (MDA-Net). RT means *Rol Transformer*. Aug. means the data augmentations in Sec. 6.1.5. FR-O* means the re-implemented Faster R-CNN OBB detector, which is slightly different from the original FR-O [14].

		OBB Results																	
method	backbone	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	mAP		
FR-O [14]	R-101	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		
ICN [19]	DR-101-FPN	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		
LR-O + RT [18]	R-101-FPN	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 [24]	R-101-SF-MDA	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		
DRN [90]	H-104	89.71	82.34	47.22	64.10	76.22	74.43	85.84	90.57	86.18	84.89	57.65	61.93	69.30	69.63	58.48	73.23		
Gliding Vertex [63]	R-101-FPN	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		
CenterMap [66]	R-101-FPN	89.83	84.41	54.60	70.25	77.66	78.32	87.19	90.66	84.89	85.27	56.46	69.23	74.13	71.56	66.06	76.03		
CSL [91]	R-152-FPN	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		
Li et al. [25]	R-101-FPN	90.41	85.21	55.00	78.27	76.19	72.19	82.14	90.70	87.22	86.87	66.62	68.43	75.43	72.70	57.99	76.36		
S ² A-Net [57]	R-50-FPN	88.89	83.60	57.74	81.95	79.94	83.19	89.11	90.78	84.87	87.81	70.30	68.25	78.30	77.01	69.58	79.42		
FR-O* + RT [18]	R-50-FPN	88.34	77.07	51.63	69.62	77.45	77.15	87.11	90.75	84.90	83.14	52.95	63.75	74.45	68.82	59.24	73.76		
FR-O* + RT (Aug.) [18]	R-50-FPN	87.89	85.01	57.83	78.55	75.22	84.37	88.04	90.88	87.28	85.79	71.04	69.67	79.00	83.29	73.43	79.82		

		HBB Results																	
method	backbone	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	mAP		
ICN [19]	DR-101-FPN	89.97	77.71	53.38	73.26	73.46	65.02	78.22	90.79	79.05	84.81	57.20	62.11	73.45	70.22	58.08	72.45		
SCRDet [24]	R-101-SF-MDA	90.18	81.88	55.30	73.29	72.09	77.65	78.06	90.91	82.44	86.39	64.53	63.45	75.77	78.21	60.11	75.35		
CenterMap [91]	R-101-FPN	89.70	84.92	59.72	67.96	79.16	80.66	86.61	90.47	84.47	86.19	56.42	69.00	79.33	80.53	64.81	77.33		
Li et al. [25]	ResNet101	90.41	85.77	61.94	78.18	77.00	79.94	84.03	90.88	87.30	86.92	67.78	68.76	82.10	80.44	60.43	78.79		
FR-O* + RT [18]	R-50-FPN	88.47	81.00	54.10	69.19	78.42	81.16	87.35	90.75	84.90	83.55	52.63	62.97	75.89	71.31	57.22	74.59		
FR-O* + RT (Aug.) [18]	R-50-FPN	87.91	85.11	62.65	77.73	75.83	85.03	88.18	90.88	87.28	86.18	71.49	70.37	84.94	84.11	73.61	80.75		

TABLE 14

DOAI 2019 Challenge Results. CC is the *container crane* for short. The other abbreviations for categories are the same as those in Tab. 13. The USTC-NELSLIP, pca_lab and czh, AICyber are the top 3 participants in the OBB and HBB Tasks. The FR-O means Faster R-CNN OBB. RT means the Rol Transformer. Aug. means the data augmentation method described in Sec. 6.1.5. Note that FR-O + RT and FR-O + RT (Aug.) are single models, while others are ensembles of multiple models.

		OBB results																	
team (method)		Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	CC	mAP	
USTC-NELSLIP [92]	89.19	85.32	57.27	80.86	73.87	81.26	89.5	90.84	85.94	85.62	69.5	76.73	76.34	76	77.84	57.33	78.34		
pca_lab [25]	89.11	83.83	59.55	82.8	66.93	82.51	89.78	90.88	85.36	84.22	71.95	77.89	78.47	74.27	74.77	53.22	77.84		
czh	89	83.22	54.47	73.79	72.61	80.28	89.32	90.83	84.36	85	68.68	75.3	74.22	74.41	73.45	42.13	75.69		
FR-O + RT [18]	71.92	76.07	51.87	69.24	52.05	75.18	80.72	90.53	78.58	68.26	49.18	71.74	67.51	65.53	62.16	9.99	65.03		
FR-O + RT (Aug.) [18]	87.54	84.34	62.22	79.77	67.29	83.16	89.93	90.86	83.85	77.74	73.91	75.31	78.61	77.07	75.20	54.77	77.60		

		HBB results																	
team (method)		Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	CC	mAP	
pca_lab [25]	88.26	86.55	65.68	79.83	74.59	79.35	88.12	90.86	85.45	84.15	73.9	77.44	84.1	81.07	76.07	57.07	79.53		
USTC-NELSLIP [92]	89.26	85.6	59.61	80.86	75.2	81.13	89.58	90.84	85.94	85.71	69.5	76.34	81.7	81.84	76.53	57.09	79.17		
AICyber	89.2	85.56	64.44	74.07	77.45	81.5	89.65	90.83	85.72	86.03	69.82	76.34	82.89	82.95	74.64	44.02	78.44		
FR-O + RT [18]	71.92	75.21	54.09	68.10	52.54	74.87	80.79	90.46	78.58	68.41	51.57	71.48	74.91	74.84	56.66	13.01	66.09		
FR-O + RT (Aug.) [18]	87.79	84.33	63.75	79.13	72.92	83.08	90.04	90.86	83.85	77.80	73.30	75.66	84.84	82.16	75.20	57.39	78.88		

ODAI algorithms. The results show that hyperparameter selection and module design of the algorithms (*e.g.*, number of proposals) for aerial images are very different from those for natural images. It indicates that DOTA can be used as a supplement to natural scene images to facilitate universal object detection.

In the future, we will continue to extend the dataset, host more challenges, and integrate more algorithms for oriented object detection into our code library. We believe that DOTA, challenges and code library will not only promote the development of object detection in Earth vision but also pose interesting algorithmic questions for general object detection in computer vision.

ACKNOWLEDGMENT

We thank the support of CycloMedia B.V. for providing the airborne images in DOTA-v2.0. We thank Huan Yi, Zhipeng Lin, Fan Hu, Pu Jin, Xinyi Tong, Xuan Hu, Zhipeng Dong, Liang Wu, Jun Tang, Linyan Cui, Duoyou Zhou, Tengteng Huang, and all the others who involved in the annotations of DOTA. We also thank Zhen Zhu for his

advice on running Faster R-CNN, and Jinwang Wang for valuable discussions in details of parameter settings. The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

REFERENCES

- [1] E. J. Sadgrove, G. Falzon, D. Miron, and D. W. Lamb, "Real-time object detection in agricultural/remote environments using the multiple-expert colour feature extreme learning machine (mecelm)," *Computers in Industry*, vol. 98, pp. 183–191, 2018.
- [2] V. Reilly, H. Idrees, and M. Shah, "Detection and tracking of large number of targets in wide area surveillance," in *ECCV*. Springer, 2010, pp. 186–199.
- [3] J. Porway, Q. Wang, and S. C. Zhu, "A hierarchical and contextual model for aerial image parsing," *IJCV*, vol. 88, no. 2, pp. 254–283, 2010.
- [4] 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, 2016.
- [5] T. Moranduzzo and F. Melgani, "Detecting cars in uav images with a catalog-based approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6356–6367, 2014.

- [6] G. Cheng, J. Han, P. Zhou, and D. Xu, "Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection," *IEEE TIP*, vol. 28, no. 1, pp. 265–278, 2018.
- [7] C. Benedek, X. Descombes, and J. Zerubia, "Building development monitoring in multitemporal remotely sensed image pairs with stochastic birth-death dynamics," *IEEE TPAMI*, vol. 34, no. 1, pp. 33–50, 2012.
- [8] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, "Drone-based object counting by spatially regularized regional proposal network," in *ICCV*, 2017, pp. 4145–4153.
- [9] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, 2016.
- [10] Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao, "Oriented response networks," in *CVPR*. IEEE, 2017, pp. 4961–4970.
- [11] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *IJCV*, vol. 88, no. 2, pp. 303–338, 2010.
- [12] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.
- [13] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *ECCV*, 2014, pp. 740–755.
- [14] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "Dota: A large-scale dataset for object detection in aerial images," in *CVPR*, 2018.
- [15] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery: A small target detection benchmark," *J Vis. Commun. Image R.*, vol. 34, pp. 187–203, 2016.
- [16] K. Liu and G. Mátyus, "Fast multiclass vehicle detection on aerial images," *IEEE Geosci. Remote Sensing Lett.*, vol. 12, no. 9, pp. 1938–1942, 2015.
- [17] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye, and J. Jiao, "Orientation robust object detection in aerial images using deep convolutional neural network," in *ICIP*, 2015, pp. 3735–3739.
- [18] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning roi transformer for oriented object detection in aerial images," in *CVPR*, 2019, pp. 2849–2858.
- [19] S. M. Azimi, E. Vig, R. Bahmanyar, M. Körner, and P. Reinartz, "Towards multi-class object detection in unconstrained remote sensing imagery," in *ACCV*. Springer, 2018, pp. 150–165.
- [20] J. Pang, C. Li, J. Shi, Z. Xu, and H. Feng, "R²-cnn: Fast tiny object detection in large-scale remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, 2019.
- [21] R. LaLonde, D. Zhang, and M. Shah, "Clusternet: Detecting small objects in large scenes by exploiting spatio-temporal information," in *CVPR*, June 2018.
- [22] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling, "Clustered object detection in aerial images," in *CVPR*, 2019, pp. 8311–8320.
- [23] M. Liao, Z. Zhu, B. Shi, G.-s. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *CVPR*, 2018, pp. 5909–5918.
- [24] X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang, Z. Guo, X. Sun, and K. Fu, "Scrdet: Towards more robust detection for small, cluttered and rotated objects," in *ICCV*, 2019, pp. 8232–8241.
- [25] C. Li, C. Xu, Z. Cui, D. Wang, Z. Jie, T. Zhang, and J. Yang, "Learning object-wise semantic representation for detection in remote sensing imagery," in *CVPR Workshops*, 2019, pp. 20–27.
- [26] G. Heitz and D. Koller, "Learning spatial context: Using stuff to find things," in *ECCV*, 2008, pp. 30–43.
- [27] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, and B. McCord, "xview: Objects in context in overhead imagery," *arXiv:1802.07856*, 2018.
- [28] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv:1906.07155*, 2019.
- [29] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, "Detectron," <https://github.com/facebookresearch/detectron>, 2018.
- [30] X. Wang, Z. Cai, D. Gao, and N. Vasconcelos, "Towards universal object detection by domain attention," in *CVPR*, 2019, pp. 7289–7298.
- [31] B. Yao, X. Yang, and S.-C. Zhu, "Introduction to a large-scale general purpose ground truth database: Methodology, annotation tool and benchmarks," in *EMMCVPR*, 2007, pp. 169–183.
- [32] B. Zhou, Á. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *NIPS*, 2014, pp. 487–495.
- [33] Q. You, J. Luo, H. Jin, and J. Yang, "Building a large scale dataset for image emotion recognition: The fine print and the benchmark," in *AAAI*, 2016, pp. 308–314.
- [34] G. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, 2017.
- [35] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Mallochi, T. Duerig *et al.*, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *arXiv:1811.00982*, 2018.
- [36] N. Weir, D. Lindenbaum, A. Bastidas, A. V. Etten, S. McPherson, J. Shermeyer, V. Kumar, and H. Tang, "Spacenet mvoi: A multi-view overhead imagery dataset," in *ICCV*, October 2019.
- [37] T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye, "A large contextual dataset for classification, detection and counting of cars with deep learning," in *ECCV*, 2016, pp. 785–800.
- [38] M. Y. Yang, W. Liao, X. Li, and B. Rosenhahn, "Deep learning for vehicle detection in aerial images," in *ICIP*. IEEE, 2018, pp. 3079–3083.
- [39] M. Y. Yang, W. Liao, X. Li, Y. Cao, and B. Rosenhahn, "Vehicle detection in aerial images," *Photogrammetric engineering and remote sensing : PE&RS*, vol. 85, no. 4, pp. 297–304, 2019.
- [40] K. Chen, M. Wu, J. Liu, and C. Zhang, "Fgsd: A dataset for fine-grained ship detection in high resolution satellite images," *arXiv:2003.06832*, 2020.
- [41] J. Shermeyer, T. Hossler, A. Van Etten, D. Hogan, R. Lewis, and D. Kim, "Rareplanes: Synthetic data takes flight," *arXiv:2006.02963*, 2020.
- [42] G. Cheng, J. Han, P. Zhou, and L. Guo, "Multi-class geospatial object detection and geographic image classification based on collection of part detectors," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 98, pp. 119–132, 2014.
- [43] Y. Long, Y. Gong, Z. Xiao, and Q. Liu, "Accurate object localization in remote sensing images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2486–2498, 2017.
- [44] Z. Zou and Z. Shi, "Random access memories: A new paradigm for target detection in high resolution aerial remote sensing images," *IEEE TIP*, vol. 27, no. 3, pp. 1100–1111, 2017.
- [45] Y. Zhang, Y. Yuan, Y. Feng, and X. Lu, "Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection," *IEEE Trans. Geosci. Remote Sens.*, 2019.
- [46] S. Waqas Zamir, A. Arora, A. Gupta, S. Khan, G. Sun, F. Shahbaz Khan, F. Zhu, L. Shao, G.-s. Xia, and X. Bai, "isaid: A large-scale dataset for instance segmentation in aerial images," in *CVPR Workshops*, 2019, pp. 28–37.
- [47] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object detection in optical remote sensing images: A survey and a new benchmark," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 159, pp. 296–307, 2020.
- [48] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu, "Vision meets drones: A challenge," *arXiv:1804.07437*, 2018.
- [49] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR*, 2011, pp. 1521–1528.
- [50] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *ECCV*, 2016, pp. 21–37.
- [51] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *CVPR*, 2017, pp. 7263–7271.
- [52] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580–587.
- [53] R. Girshick, "Fast r-cnn," in *CVPR*, 2015, pp. 1440–1448.
- [54] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE TPAMI*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [55] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," *CoRR, abs/1703.06211*, vol. 1, no. 2, p. 3, 2017.
- [56] Z. Liu, J. Hu, L. Weng, and Y. Yang, "Rotated region based cnn for ship detection," in *ICIP*. IEEE, 2017, pp. 900–904.
- [57] J. Han, J. Ding, J. Li, and G.-S. Xia, "Align deep features for oriented object detection," *arXiv:2008.09397*, 2020.

- [58] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *CVPR*, vol. 1, no. 2, 2017, p. 3.
- [59] F. Wang, L. Zhao, X. Li, X. Wang, and D. Tao, "Geometry-aware scene text detection with instance transformation network," in *CVPR*, 2018, pp. 1381–1389.
- [60] C. Huang, H. Ai, Y. Li, and S. Lao, "High-performance rotation invariant multiview face detection," *IEEE TPAMI*, vol. 29, no. 4, pp. 671–686, 2007.
- [61] X. Shi, S. Shan, M. Kan, S. Wu, and X. Chen, "Real-time rotation-invariant face detection with progressive calibration networks," in *CVPR*, 2018, pp. 2295–2303.
- [62] M. Liao, B. Shi, and X. Bai, "Textboxes++: A single-shot oriented scene text detector," *IEEE TIP*, vol. 27, no. 8, pp. 3676–3690, 2018.
- [63] Y. Xu, M. Fu, Q. Wang, Y. Wang, K. Chen, G.-S. Xia, and X. Bai, "Gliding vertex on the horizontal bounding box for multi-oriented object detection," *IEEE TPAMI*, 2020.
- [64] X. Yang and J. Yan, "Arbitrary-oriented object detection with circular smooth label," *ECCV*, 2020.
- [65] J. Wang, J. Ding, H. Guo, W. Cheng, T. Pan, and W. Yang, "Mask obb: A semantic attention-based mask oriented bounding box representation for multi-category object detection in aerial images," *Remote Sensing*, vol. 11, no. 24, p. 2930, 2019.
- [66] J. Wang, W. Yang, H.-C. Li, H. Zhang, and G.-S. Xia, "Learning center probability map for detecting objects in aerial images," *IEEE Trans. Geosci. Remote Sens.*, 2020.
- [67] B. Uzkent, C. Yeh, and S. Ermon, "Efficient object detection in large images using deep reinforcement learning," in *WACV*, 2020, pp. 1824–1833.
- [68] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *CVPR*, 2017, pp. 7310–7311.
- [69] F. Massa and R. Girshick, "maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch," <https://github.com/facebookresearch/maskrcnn-benchmark>, 2018.
- [70] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," <https://github.com/facebookresearch/detectron2>, 2019.
- [71] Y. Chen, C. Han, Y. Li, Z. Huang, Y. Jiang, N. Wang, and Z. Zhang, "Simpledet: A simple and versatile distributed framework for object detection and instance recognition," *JMLR*, vol. 20, no. 156, pp. 1–8, 2019.
- [72] A.-M. de Oca, R. Bahmanyar, N. Nistor, and M. Datcu, "Earth observation image semantic bias: A collaborative user annotation approach," *J-STARS*, 2017.
- [73] Z. Zhang, G. Vosselman, M. Gerke, C. Persello, D. Tuia, and M. Y. Yang, "Detecting building changes between airborne laser scanning and photogrammetric data," *Remote sensing*, vol. 11, no. 20, p. 2417, 2019.
- [74] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *CVPR*, 2010, pp. 3485–3492.
- [75] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *IJCV*, vol. 123, no. 1, pp. 32–73, 2017.
- [76] D. P. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari, "Extreme clicking for efficient object annotation," in *ICCV*, 2017, pp. 4930–4939.
- [77] Z. Wu, N. Bodla, B. Singh, M. Najibi, R. Chellappa, and L. S. Davis, "Soft sampling for robust object detection," *arXiv:1806.06986*, 2018.
- [78] H. Zhang, F. Chen, Z. Shen, Q. Hao, C. Zhu, and M. Savvides, "Solving missing-annotation object detection with background recalibration loss," in *ICASSP*. IEEE, 2020, pp. 1888–1892.
- [79] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *CVPR*, 2019, pp. 2888–2897.
- [80] W. Li, W. Wei, and L. Zhang, "Gsdet: Object detection in aerial images based on scale reasoning," *IEEE TIP*, vol. 30, pp. 4599–4609, 2021.
- [81] B. Singh and L. S. Davis, "An analysis of scale invariance in object detection snip," in *CVPR*, 2018, pp. 3578–3587.
- [82] wiki, "gsd," https://en.wikipedia.org/wiki/Ground_sample_distance.
- [83] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. K. Ghosh, A. D. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chan-
- drasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny, "ICDAR 2015 competition on robust reading," in *ICDAR*, 2015.
- [84] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A face detection benchmark," in *CVPR*, 2016, pp. 5525–5533.
- [85] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017, pp. 2980–2988.
- [86] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*. IEEE, 2017, pp. 2980–2988.
- [87] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Light-head r-cnn: In defense of two-stage object detector," *arXiv:1711.07264*, 2017.
- [88] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang *et al.*, "Hybrid task cascade for instance segmentation," in *CVPR*, 2019, pp. 4974–4983.
- [89] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," *TMM*, 2018.
- [90] X. Pan, Y. Ren, K. Sheng, W. Dong, H. Yuan, X. Guo, C. Ma, and C. Xu, "Dynamic refinement network for oriented and densely packed object detection," in *CVPR*, 2020, pp. 11207–11216.
- [91] X. Yang and J. Yan, "Arbitrary-oriented object detection with circular smooth label," *arXiv:2003.05597*, 2020.
- [92] Y. Zhu, X. Wu, and J. Du, "Adaptive period embedding for representing oriented objects in aerial images," *arXiv:1906.09447*, 2019.



Jian Ding is currently pursuing his Ph.D degree at the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University. He received the B.S. degree in Aircraft Design and Engineering from Northwestern Polytechnical University, Xian, China in 2017. His research interests include object detection, instance segmentation and remote sensing.



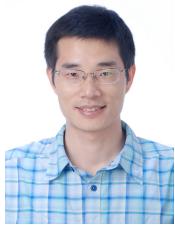
Nan Xue is currently a Research Associate Professor in the School of Computer Science, Wuhan University. He received the B.S., and Ph.D. degrees from Wuhan University in 2014 and 2020 respectively. He was a visiting scholar at North Carolina State University from Sep. 2018 to June 2020. His research interests include geometric structure analysis in computer vision.



Gui-Song Xia received his Ph.D. degree in image processing and computer vision from CNRS LTCI, Télécom ParisTech, Paris, France, in 2011. From 2011 to 2012, he has been a Post-Doctoral Researcher with the Centre de Recherche en Mathématiques de la Decision, CNRS, Paris-Dauphine University, Paris, for one and a half years. He is currently working as a full professor at Wuhan University. He has also been working as Visiting Scholar at DMA, École Normale Supérieure (ENS-Paris) for two months in 2018. His current research interests include mathematical modeling of images and videos, structure from motion, perceptual grouping, and remote sensing image understanding. He serves on the Editorial Boards of several journals, including Pattern Recognition, Signal Processing: Image Communications, EURASIP Journal on Image & Video Processing, Journal of Remote Sensing, and Frontiers in Computer Science: Computer Vision.



Xiang Bai received the B.S., M.S., and Ph.D. degrees from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2003, 2005, and 2009, respectively, all in electronics and information engineering. He is currently a Professor with the School of Electronic Information and Communications and the Vice-Director of the National Center of AntiCounterfeiting Technology, HUST. His research interests include object recognition, shape analysis, and scene text recognition.



Wen Yang received the Ph.D. degree in communication and information system from Wuhan University, Wuhan, China, in 2004. From 2008 to 2009, he was a visiting Scholar with the Laboratoire Jean Kuntzmann (LJK), Grenoble, France. From 2010 to 2013, he was a Post-Doctoral Researcher with the State Key Laboratory of Information Engineering, Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University. Since then, he has been a Full Professor with the School of Electronic Information, Wuhan University. His research interests include object detection and recognition, semantic segmentation, and change detection.



Michael Ying Yang is currently Assistant Professor in the Department of Earth Observation Science at ITC - Faculty of Geo-Information Science and Earth Observation, University of Twente, The Netherlands, heading a group working on scene understanding. He received the PhD degree (summa cum laude) from University of Bonn (Germany) in 2011. He received the *venia legendi* in Computer Science from Leibniz University Hannover in 2016. His research interests are in the fields of computer vision and photogrammetry with specialization on scene understanding and semantic interpretation from imagery. He serves as Associate Editor of ISPRS Journal of Photogrammetry and Remote Sensing, Co-chair of ISPRS working group II/5 Dynamic Scene Analysis, Program Chair of ISPRS Geospatial Week 2019, and recipient of ISPRS President's Honorary Citation (2016), Best Science Paper Award at BMVC (2016), and The Willem Schermerhorn Award (2020).



Serge Belongie received a B.S. (with honor) in EE from Caltech in 1995 and a Ph.D. in EECS from Berkeley in 2000. While at Berkeley, his research was supported by an NSF Graduate Research Fellowship. From 2001-2013 he was a professor in the Department of Computer Science and Engineering at University of California, San Diego. He is currently a professor at Cornell Tech and the Department of Computer Science at Cornell University. His research interests include Computer Vision, Machine Learning, Crowdsourcing and Human-in-the-Loop Computing. He is also a co-founder of several companies including Digital Persona, Anchovi Labs and Orpix. He is a recipient of the NSF CAREER Award, the Alfred P. Sloan Research Fellowship, the MIT Technology Review "Innovators Under 35" Award and the Helmholtz Prize for fundamental contributions in Computer Vision.



Jiebo Luo joined the University of Rochester in Fall 2011 after over fifteen prolific years at Kodak Research Laboratories, where he was a Senior Principal Scientist leading research and advanced development. He has been involved in numerous technical conferences, including serving as the program co-chair of ACM Multimedia 2010, IEEE CVPR 2012 and IEEE ICIP 2017. He has served on the editorial boards of the IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Multimedia, IEEE Transactions on Circuits and Systems for Video Technology, ACM Transactions on Intelligent Systems and Technology, Pattern Recognition, Machine Vision and Applications, and Journal of Electronic Imaging. Dr. Luo will serve as the Editor-in-Chief of the IEEE Transactions on Multimedia for the 2020-2022 term. Dr. Luo is a Fellow of SPIE, IAPR, IEEE, ACM, and AAAI. In addition, he is a Board Member of the Greater Rochester Data Science Industry Consortium.



Mihai Datcu received the M.S. and Ph.D. degrees in electronics and telecommunications from the University "Politehnica" of Bucharest UPB, Bucharest, Romania, in 1978 and 1986, and the title "Habilitation a diriger des recherches" from Université Louis Pasteur, Strasbourg, France. He holds a Professorship in electronics and telecommunications with UPB since 1981. Since 1993, he has been a Scientist with the German Aerospace Center (DLR), Oberpfaffenhofen, Germany. He is

currently developing algorithms for model-based information retrieval from high-complexity signals, methods for scene understanding from SAR and interferometric SAR data, and he is engaged in research in information theoretical aspects and semantic representations in advanced communication systems. His research interests are in Bayesian inference, information and complexity theory, stochastic processes, model-based scene understanding, image information mining, for applications in information retrieval and understanding of high-resolution SAR and optical observations.



Marcello Pelillo is a Full Professor of Computer Science at Ca' Foscari University, Venice, where he leads the Computer Vision and Pattern Recognition Lab. He has been the Director of the European Centre for Living Technology (ECLT) and has held visiting research/teaching positions in several institutions including Yale University (USA), University College London (UK), McGill University (Canada), University of Vienna (Austria), York University (UK), NICTA (Australia), Wuhan University (China), Huazhong University

of Science and Technology (China), and South China University of Technology (China). He is also an external affiliate of the Computer Science Department at Drexel University (USA). His research interests are in the areas of computer vision, machine learning and pattern recognition where he has published more than 200 technical papers in refereed journals, handbooks, and conference proceedings. He has been General Chair for ICCV 2017, Program Chair for ICPR 2020, and has been Track or Area Chair for several conferences in his area. He is the Specialty Chief Editor of Frontiers in Computer Vision and serves, or has served, on the Editorial Boards of several journals, including IEEE Transactions on Pattern Analysis and Machine Intelligence, Pattern Recognition, IET Computer Vision, and Brain Informatics. He also serves on the Advisory Board of Springer's International Journal of Machine Learning and Cybernetics. Prof. Pelillo has been elected Fellow of the IEEE and Fellow of the IAPR, and is an IEEE SMC Distinguished Lecturer.



Liangpei Zhang received the B.S. degree in physics from Hunan Normal University, Changsha, China, in 1982, the M.S. degree in optics from the Xian Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xian, China, in 1988, and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 1998. He is currently a Chang-Jiang Scholar Chair Professor with Wuhan University, appointed by the Ministry of Education of China. He has authored

or coauthored over 500 research papers and five books. He holds 15 patents. His research interests include hyper spectral remote sensing, high resolution remote sensing, image processing, and artificial intelligence. Dr. Zhang was a recipient of the 2010 Best Paper Boeing Award and the 2013 Best Paper ERDAS Award from the American Society of Photogrammetry and Remote Sensing. He serves as a Co-Chair for the series SPIE Conferences on Multispectral Image Processing and Pattern Recognition, the Conference on Asia Remote Sensing, and many other conferences. He serves as an Associate Editor for the IEEE Transactions on Geoscience and Remote Sensing. He is a fellow of IEEE.