VisDrone-DET2019: The Vision Meets Drone Object Detection in Image Challenge Results

Dawei Du[1], Pengfei Zhu[2], Longyin Wen[3], Xiao Bian[4], Haibin Ling[5], Qinghua Hu[1], Tao Peng[2], Jiayu Zheng[2], Xinyao Wang[3], Yue Zhang[3], Liefeng Bo[3], Hailin Shi[6], Rui Zhu[6], Aashish Kumar[22], Aijin Li[30], Almaz Zinollayev[32], Anuar Askergaliyev[32], Arne Schumann[33], Binjie Mao[20], Byeongwon Lee[15], Chang Liu[23], Changrui Chen[8], Chunhong Pan[20], Chunlei Huo[20], Da Yu[25], Dechun Cong[24], Dening Zeng[30], Dheeraj Reddy Pailla[13], Di Li[30], Dong Wang[23], Donghyeon Cho[9], Dongyu Zhang[31], Furui Bai[28], George Jose[22], Guangyu Gao[18], Guizhong Liu[14], Haitao Xiong[12], Hao Qi[14], Haoran Wang[30], Heqian Qiu[7], Hongliang Li[7], Huchuan Lu[23], Ildoo Kim[29], Jaekyum Kim[16], Jane Shen[28], Jihoon Lee[29], Jing Ge[18], Jingjing Xu[24], Jingkai Zhou[12], Jonas Meier[33], Jun Won Choi[16], Junhao Hu[11], Junyi Zhang[31], Junying Huang[31], Kaiqi Huang[20], Keyang Wang[17], Lars Sommer[33], Lei Jin[11], Lei Zhang[17], Lianghua Huang[20], Lin Sun[19], Lucas Steinmann[33], Meixia Jia[30], Nuo Xu[20], Pengyi Zhang[18], Qiang Chen[20], Qingxuan Lv[8], Qiong Liu[12], Qishang Cheng[7], Sai Saketh Chennamsetty[13], Shuhao Chen[23], Shuo Wei[8], Srinivas S S Kruthiventi[22], Sungeun Hong[9], Sungil Kang[9], Tong Wu[18], Tuo Feng[30], Varghese Alex Kollerathu[13], Wanqi Li[14], Wei Dai[21], Weida Qin[12], Weiyang Wang[21], Xiaorui Wang[8], Xiaoyu Chen[7], Xin Chen[23], Xin Sun[8], Xin Zhang[20], Xin Zhao[20], Xindi Zhang[26], Xinyu Zhang[23], Xuankun Chen[31], Xudong Wei[14], Xuzhang Zhang[27], Yanchao Li[28], Yifu Chen[10], Yu Heng Toh[28], Yu Zhang[8], Yu Zhu[20], Yunxin Zhong[18], Zexin Wang[30], Zhikang Wang[28], Zichen Song[7], Ziming Liu[18]

[1]University at Albany, SUNY, Albany, NY, USA
[2]Tianjin University, Tianjin, China
[3]JD Digits, Mountain View, CA, USA
[4]GE Global Research, Niskayuna, NY, USA
[5]Stony Brook University, New York, NY, USA
[6]JD AI research, Beijing, USA
[7]University of Electronic Science and Technology of China, Chengdu, China
[8]Ocean University of China, Qingdao, China
[9]SK T-Brain, Seoul, South Korea
[10]Harbin Institute of Technology, Shenzhen, China
[11]ShanghaiTech University, Shanghai, China
[12]South China University of Technology, Guangzhou, China
[13]Siemens Technology and Services Private Limited, Bengaluru, India
[14]Xi'an Jiaotong University, Xi'an, China
[15]SK Telecom, Seongnam-si, South Korea
[16]Hanyang University, Seoul, South Korea
[17]Chongqing University, Chongqing, China
[18]Beijing Institute of Technology, Beijing, China
[19]Samsung Inc., San Jose, CA, USA
[20]Institute of Automation, Chinese Academy of Sciences, Beijing, China
[21]Snowcloud.ai, Beijing, China
[22]Harman-Samsung, Bangalore, India
[23]Dalian University of Technology, Dalian, China
[24]Nanjing University of Posts and Telecommunications, Nanjing, China
[25]Harbin Institute of Technology, Harbin, China
[26]Queen Mary University of London, London, UK
[27]Huazhong University of Science and Technology, Wuhan, China
[28]Pensees Singapore Institute, Singapore
[29]Kakao Brain, Seongnam, South Korea
[30]Xidian University, Xi'an, China
[31]SUN YAT-SEN University, Guangzhou, China
[32]BTS Digital, Astana, Kazakhstan
[33]Fraunhofer IOSB, Karlsruhe, Germany

Abstract

Recently, automatic visual data understanding from drone platforms becomes highly demanding. To facilitate the study, the Vision Meets Drone Object Detection in Image Challenge is held the second time in conjunction with the 17*-th International Conference on Computer Vision (ICCV 2019), focuses on image object detection on drones. Results of* 33 *object detection algorithms are presented. For each participating detector, a short description is provided in the appendix. Our goal is to advance the state-of-the-art detection algorithms and provide a comprehensive evaluation platform for them. The evaluation protocol of the VisDrone-DET2019 Challenge and the comparison results of all the submitted detectors on the released dataset are publicly available at the website:* `http://www.aiskyeye.com/`*. The results demonstrate that there still remains a large room for improvement for object detection algorithms on drones.*

1. Introduction

Object detection is a basis of a wide range of many high-level computer vision applications, such as autonomous driving, face detection and recognition, and activity recognition. Although significant progress has been achieved in recent years, these algorithms usually focus on detection in general scenarios instead of drone-captured scenes. This is because the studies are seriously limited by the lack of public large-scale benchmarks or datasets.

To advance state-of-the-art detection algorithms in drone-based scenes, the first Vision Meets Drone Object Detection in Images Challenge (VisDrone-DET2018) [49] was held on September 8, 2018, in conjunction with the 15-th European Conference on Computer Vision (ECCV 2018) in Munich, Germany. Compared with the previous drone based datasets [29, 15, 9], a larger scale drone based object detection dataset [48] is proposed to evaluate detection algorithms in real scenarios. Then, there were 34 object detection methods submitted to this challenge, and we provided a comprehensive performance evaluation for them.

In this paper, researchers are encouraged to submit algorithms to detect objects of ten predefined categories (*e.g.*, pedestrian and car) in the VisDrone-DET2019 dataset. Specifically, there are 33 out 47 detection methods that performs better than the baseline state-of-the-arts. Derived from recently published top computer vision conferences or journals, we believe this challenge is useful to further promote the development of object detection algorithms on drone platforms. The experiments can be found at our website: `http://www.aiskyeye.com/`.

2. Related Work

2.1. Anchor based Detectors

The current state-of-the-art anchor based detectors can be divided into two categories: (1) the two-stage methods [33, 25, 13] with high accuracy, and (2) the one-stage methods [28, 32] with high efficiency.

Based on the previous works, Lin *et al.* [26] develop focal Loss to address the class imbalance issue in object detection by reshaping the standard cross entropy loss such that it down-weights the loss assigned to well-classified examples. To address the shortcoming in current two-stage methods, Li *et al.* [23] propose a new two-stage detector to make the head of network as light as possible, by using a thin feature map and a cheap R-CNN subnet (pooling and single fully-connected layer). To inherit the merits of both two-stage and one-stage methods, Zhang *et al.* [45] propose a single-shot detector formed by two inter-connected modules, *i.e.*, the anchor refinement module and the object detection module. Moreover, the Cascade R-CNN [2] is a multi-stage object detection architecture. That is, a sequence of detectors are trained with increasing IoU thresholds to be sequentially more selective against close false positives. Recently, Duan *et al.* [11] propose the channel-aware deconvolutional network to detect small objects, especially for drone based scenes. To keep the favourable performance independent to the network architecture, Zhu *et al.* [50] train detectors from scratch using BatchNorm with larger learning rate.

2.2. Anchor-free Detectors

Although anchor based detectors have achieved much progress in object detection, it is still difficult to select optimal parameters of anchors. To guarantee high recall, more anchors are essential but introduce high computational-complexity. Moreover, different datasets correspond to different optimal anchors. To solve these issues, anchor-free detectors attract much research and have achieved significant advances with complex backbone networks recently.

Law and Deng [18] propose the CornerNet to detect an object bounding box as a pair of keypoints, the top-left corner and the bottom-right corner, using a single convolution neural network. To decrease the high processing cost, they further introduce CornerNet-Lite. It is a combination of two efficient variants of CornerNet: CornerNet-Saccade with an attention mechanism and CornerNet-Squeeze with a new compact backbone architecture [19]. Moreover, Duan *et al.* [10] detect the object as a triplet, rather than a pair, of keypoints, which improves both precision and recall. Zhou *et al.* [46] further model an object as the center point of its bounding box, and regress to all other object properties, such as size, 3D location, orientation, and even pose. On the other hand, Kong *et al.* [17] propose an accurate, flexible and completely anchor-free framework, which pre-

dicts category-sensitive semantic maps for the object existing possibility and category-agnostic bounding box for each position that potentially contains an object. Tian *et al.* [37] solve object detection in a per-pixel prediction fashion, analogue to semantic segmentation.

3. The VisDrone-DET2019 Challenge

Similar to the VisDrone-DET2018 Challenge [49], we mainly focus on human and vehicles in our daily life, and detect ten object categories of interest including *pedestrian*, *person*[1], *car*, *van*, *bus*, *truck*, *motor*, *bicycle*, *awning-tricycle*, and *tricycle*.

To obtain results on the VisDrone-DET2019 test-challenge set, the participators must generate the results in defined format and then upload to the evaluation server. If the results of the submitted method are above the performance of Cascade R-CNN [2], it will be automatically published in the ICCV 2019 workshop proceeding. Moreover, only the algorithms with detailed description (*e.g.*, speed, GPU and CPU information) have the the right of authorship.

3.1. The VisDrone-DET2019 Dataset

The VisDrone-DET2019 Dataset uses the same data in The VisDrone-DET2018 Dataset [49], namely $8,599$ images captured by drone platforms in different places at different heights. Moreover, more than $540k$ bounding boxes of targets are annotated with ten predefined categories. The dataset is divided into training, validation and testing subsets ($6,471$ for training, 548 for validation, $1,580$ for testing), which are collected from different locations but similar environments.

Furthermore, we use the evaluation protocol in MS COCO [27] to evaluate the results of detection algorithms, including AP, AP50, AP75, AR1, AR10, AR100 and AR500 metrics. Specifically, AP is computed by averaging over all 10 Intersection over Union (IoU) thresholds (*i.e.*, in the range $[0.50 : 0.95]$ with the uniform step size 0.05) of all categories, which is used as the primary metric for ranking. AP50 and AP75 are computed at the single IoU thresholds 0.5 and 0.75 over all categories. The AR1, AR10, AR100 and AR500 scores are the maximum recalls given $1, 10, 100$ and 500 detections per image respectively, averaged over all categories and IoU thresholds. Note that these criteria penalize missing detection of objects as well as duplicate detections (two detection results for the same object instance). Please refer to [27] for more details.

3.2. Submitted Detectors

There are 47 different object detection methods submitted to the VisDrone-DET2019 challenge, 33 of which performs better than the state-of-the-art object detector Cascade R-CNN [2]. Except Cascade R-CNN [2], The VisDrone team also gives the results of another 6 baseline methods, *i.e.*, CornerNet [18], Light-RCNN [23], DetNet59 [24], RefineDet [45], RetinaNet [26] and FPN [25]. For these baselines, the default parameters are used or set to reasonable values. Thus, there are 39 algorithms in total included in the report of VisDrone-DET2019 Challenge.

Nine submitted detectors improve the Cascade R-CNN [2], namely Airia-GA-Cascade (A.2), Cascade R-CNN+ (A.4), Cascade R-CNN++ (A.5), DCRCNN (A.13), DPN (A.14), DPNet-ensemble (A.15), MSCRDet (A.25), SAMFR-Cascade RCNN (A.29), and SGE-cascade R-CNN (A.30). Six detectors are based on CenterNet, including CenterNet (A.6), CenterNet-Hourglass (A.7), CN-DhVaSa (A.9), ConstraintNet (A.10), GravityNet (A.20) and RRNet (A.28). Five detectors are derived from RetinaNet [26], *i.e.*, DA-RetinaNet (A.11), EHR-RetinaNet (A.16), FS-Retinanet (A.19), MOD-RETINANET (A.24) and retinaplus (A.27). Three detectors employ FPN representation [25], ACM-OD (A.1), BetterFPN (A.3) and ODAC (A.26). Three detectors (*i.e.*, DBCL (A.12), HTC-drone (A.22) and S+D (A.31)) conduct segmentation of the objects to restrain the background noise. Four algorithms use ensemble of state-of-the-art detectors. Specifically, EnDet (A.17) combines YOLO and Faster R-CNN, while TSEN (A.33) use ensembles of 3 two-stage methods: Faster R-CNN, Guided Anchoring and Libra R-CNN. ERCNNs (A.18) is generated by Faster R-CNN and Cascade R-CNN with different backbones. Libra-HBR (A.23) consider SNIPER, Libra R-CNN, and cascade R-CNN. Different from FPN model, more multi-scale fusion strategies are proposed. CNAnet (A.8) use the multi-neighbor layers fusion modules to fuse the current layer with its multi-neighbor higher layers. HRDet+ (A.21) maintains high-resolution representation by connecting high-to-low convolutions in parallel. TridentNet (A.32) construct a parallel multi-branch architecture where each branch shares the same transformation parameters but with different receptive fields. More description can be found in Table 1.

3.3. Results and Analysis

The overall results of the submissions are presented in Table 2. We find that DPNet-ensemble (A.15) achieves the best performance among all submitted methods, *i.e.*, 29.62% AP score. It follows the idea of FPN [25] and improve Cascade-RCNN [2] with global context module (GC) [3] and deformable convolution (DC) [7] into the backbone network. RRNet (A.28) and ACM-OD (A.1) rank in the second place with more than 29% AP score. RRNet (A.28) is an anchor-free detector based on [46], where the re-regression module can predict the bias between the coarse bounding boxes and the ground-truth. ACM-OD

[1]If a human maintains standing pose or walking, we classify it as a *pedestrian*; otherwise, it is classified as a *person*.

Table 1. The descriptions of the submitted algorithms in the VisDrone-DET2019 Challenge. GPU/CPUs for training, implementation details, the tracking speed (in FPS), and references are reported.

Method	GPU	CPU	Code	Speed	Reference
DPNet ensemble (A.15)	TITAN Xp	Xeon E5-2620v4	C++,P	6	Cascade R-CNN [2]
RRNet (A.28)	RTX 2080ti	Xeon E5-2620v4	P	1.5	CenterNet [46]
ACM-OD (A.1)	Tesla V100	Xeon 6150	P	0.6	FPN [25]
S+D (A.31)	RTX 2080Ti	i7-7800X	P	10	DeepLab [43]+Cascade RCNN [2]
BetterFPN (A.3)	Tesla M40	Xeon E5-2680v4@2.40GHz	P	4.2	FPN [25]
HRDet+ (A.21)	RTX 2080Ti	E5-2650v4	P	5	HRNet [36]
CN-DhVaSa (A.9)	Tesla P-100	Xeon Silver 4110	p	0.3	CenterNet [46]
SGE-Cascade R-CNN (A.30)	GTX 1080Ti	E5-1620	P	4.3	Cascade R-CNN [2]
EHR-RetinaNet (A.16)	Tesla V-100	Xeon E5-2698v4@2.20GHz	P	0.5	RetinaNet [26]
CNAnet (A.8)	TITAN Xp	E5-1620	P	15	Cascade R-CNN [2]
FS-RetinaNet (A.19)	GTX 1080Ti	Intel i7-5930K	C++,P	4	RetinaNet [26]
CenterNet (A.6)	GTX 1080Ti	ES-2603	P	50	CenterNet [46]
Airia-GA-Cascade (A.2)	Tesla V100	Gold 6130	P	6.1	Cascade R-CNN [2]
MSCRDet (A.25)	GTX 1080	i7-7700	P	1.3	FPN [25]
DPN (A.14)	TITAN Xp	Xeon E5-2680v4@2.40GHz	P	3	FPN [25]
HTC-drone (A.22)	GeForce 1060	Intel i7-7700K@4.20GHz	P	1.7	HTC [4]
TridentNet (A.32)	RTX 2080Ti	Intel E5-2620v4@2.10GHz	C++,P	0.2	TridentNet [22]
CenterNet-Hourglass (A.7)	TITAN Xp	E5-2650v4	P	7.8	CenterNet [46]
ERCNNs (A.18)	Tesla V100	Intel XEON@2.20GHz	P	2	Cascade R-CNN [2]
SAMFR-Cascade RCNN (A.29)	Tesla V100	Gold 6130	P	7	Cascade R-CNN [2]
EnDet (A.17)	GTX 1080Ti	Xeon E5-2683v3@2.00GHz	P	2.3	YOLOv3 [32]+Faster R-CNN [33]
DCRCNN (A.13)	GTX 1080Ti	Xeon Gold 6126	P	3	Cacade-RCNN [2]
Cascade R-CNN+ (A.4)	Tesla P40	E5-2650v4	C++,P	5.2	Cascade R-CNN [2]
ODAC (A.26)	TITAN Xp	Xeon E5-2678v3@2.50GHz	P	2.5	Faster R-CNN [33]

(A.1) employs the framework of FPN [25] and active learning scheme to operate jointly with object augmentation. Among the 7 baseline methods provided by the VisDrone Team, CornerNet [18] achieves the best performance, while RetinaNet [26] performs the worst.

We also report the detection results of each object category in Table 3. We observe that all the best results of different kinds of objects are produced by the detectors with top 6 AP scores. However, they do not achieve good detection results in terms of *person* and *awning-tricycle*. This is because *person* does not maintain standing pose and usually involves other kinds of objects such as *tricycle* and *bicycle*, while *awning-tricycle* lacks of training data.

3.4. Discussion

In summary, the best detector DPNet-ensemble (A.15) achieves less than 30% AP score, which is still far from satisfactory in real applications. As shown in Figure 1, we report top 10 detection methods in both VisDrone-DET2018 and VisDrone-DET2019 challenges. We can conclude that DPNet-ensemble (A.15) is slightly inferior than the winner of VisDrone-DET2018 Challenge HAL-Retina-Net in terms of the AP metric. However, 33 detectors perform better than all the baseline methods. That means that the average performance in this year is better than that in the previous year.

Moreover, given the detection results of DPNet-ensemble (A.15) in Figure 2, we discuss some critical issues in object detection on drone platforms.

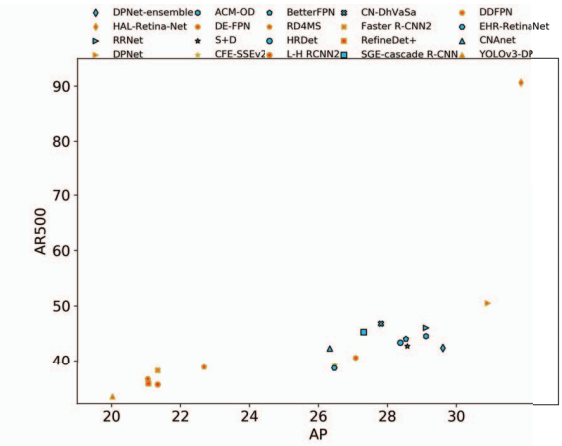

Figure 1. Comparison between the top 10 performer of VisDrone-DET2019 (blue) and VisDrone-DET2018 (red).

Small object detection. Objects are usually very small in drone based scenes. As shown in Figure 2, DPNet-ensemble (A.15) performs well in large scale objects (*e.g.*, cars) but introduce many false positives of detections with small scale (see the third row of the figure). This is because there are designed small anchors in the Cascade-RCNN [2] framework for small object detection. To achieve better performance, it is necessary to extract more contextual semantic information for discriminative representation of small objects.

Occlusion. Occlusion is another critical issue that limits the detection performance, especially in drone based scenes

Table 2. Object detection results on the VisDrone-DET2019 `testing` set. The submitted algorithms are ranked based on the AP score. $*$ indicates that the detection algorithm is submitted by the committee.

Method	AP[%]	AP50[%]	AP75[%]	AR1[%]	AR10[%]	AR100[%]	AR500[%]
DPNet-ensemble (A.15)	**29.62**	54.00	**28.70**	0.58	3.69	17.10	42.37
RRNet (A.28)	**29.13**	**55.82**	27.23	**1.02**	**8.50**	**35.19**	**46.05**
ACM-OD (A.1)	**29.13**	**54.07**	**27.38**	0.32	1.48	9.46	44.53
S+D (A.31)	**28.59**	50.97	**28.29**	0.50	3.38	15.95	42.72
BetterFPN (A.3)	28.55	53.63	26.68	0.86	7.56	**33.81**	44.02
HRDet+ (A.21)	28.39	**54.53**	26.06	0.11	0.94	12.95	43.34
CN-DhVaSa (A.9)	27.83	50.73	26.77	0.00	0.18	7.78	**46.81**
SGE-cascade R-CNN (A.30)	27.33	49.56	26.55	0.48	3.19	11.01	**45.23**
EHR-RetinaNet (A.16)	26.46	48.34	25.38	0.87	7.87	32.06	38.42
CNAnet (A.8)	26.35	47.98	25.45	0.94	7.69	32.98	42.28
FS-Retinanet (A.19)	26.31	50.52	24.07	0.43	3.01	10.23	42.88
CenterNet (A.6)	26.03	48.69	24.29	0.97	7.91	**33.40**	43.14
Airia-GA-Cascade (A.2)	25.99	45.41	26.18	0.47	3.06	10.94	41.30
GravityNet (A.20)	25.66	47.96	23.94	**1.04**	**7.99**	33.10	42.79
Libra-HBR (A.23)	25.57	48.32	24.02	0.83	7.32	33.16	38.53
MSCRDet (A.25)	25.13	46.02	24.25	0.47	3.25	14.91	38.53
DPN (A.14)	25.09	50.61	21.83	0.89	7.79	31.44	39.62
TSEN (A.33)	23.83	48.27	20.49	0.14	0.81	9.99	37.64
HTC-drone (A.22)	22.61	45.16	19.94	0.42	2.84	17.10	35.27
TridentNet (A.32)	22.51	43.29	20.50	**1.17**	**8.30**	28.98	39.84
CenterNet-Hourglass (A.7)	22.36	41.76	20.87	0.43	2.96	11.15	40.57
retinaplus (A.27)	20.57	40.57	18.09	0.77	7.08	26.75	31.25
ERCNNs (A.18)	20.45	41.20	17.84	0.93	7.57	27.61	34.29
SAMFR-Cascade RCNN (A.29)	20.18	40.03	18.42	0.46	3.49	21.60	30.82
Cascade R-CNN++ (A.5)	18.33	33.50	17.72	0.93	7.48	26.06	26.06
EnDet (A.17)	17.81	37.27	14.95	0.31	2.49	24.47	29.06
DCRCNN (A.13)	17.79	42.03	12.26	0.34	2.44	12.58	29.25
Cascade R-CNN+ (A.4)	17.67	34.89	15.83	0.91	6.59	24.21	27.06
ODAC (A.26)	17.42	40.55	12.44	0.30	1.94	11.77	27.96
DA-RetinaNet (A.11)	17.05	35.93	14.32	0.70	6.29	24.81	31.77
MOD-RETINANET (A.24)	16.96	33.77	14.90	0.69	6.03	24.27	32.47
DBCL (A.12)	16.78	31.08	16.02	0.73	6.96	22.99	22.99
ConstraintNet (A.10)	16.09	30.72	14.84	0.44	3.97	21.23	24.12
CornerNet* [18]	17.41	34.12	15.78	0.39	3.32	24.37	26.11
Light-RCNN* [23]	16.53	32.78	15.13	0.35	3.16	23.09	25.07
FPN* [25]	16.51	32.20	14.91	0.33	3.03	20.72	24.93
Cascade R-CNN* [2]	16.09	31.91	15.01	0.28	2.79	21.37	28.43
DetNet59* [24]	15.26	29.23	14.34	0.26	2.57	20.87	22.28
RefineDet* [45]	14.90	28.76	14.08	0.24	2.41	18.13	25.69
RetinaNet* [26]	11.81	21.37	11.62	0.21	1.21	5.31	19.29

where objects are often occluded by other objects or background obstacle. As shown in the second row of Figure 2, DPNet-ensemble (A.15) tends to generate duplicate detections when occlusion occurs. It is essential to handle occlusions by context or semantic information.

Data augmentation. The VisDrone data has imbalanced categories of objects. As presented in Table 3, every detection method achieves inferior performance in *awning-tricycle* and *bicycle* than that in the *car* and *pedestrian*. For example, DPNet-ensemble (A.15) produces 51.53% and 32.31% APs on the *car* and *pedestrian* classes, while only produces 18.41% and 12.86% APs on the *awning-tricycle* and *bicycle*. To deal with this issue, the detection methods can adjust the weights of different object classes in the loss function or perform data augmentation for the category with small data.

4. Conclusion

This paper reviews the VisDrone-DET2019 Challenge and its results. A set of 47 detectors have been evaluated on the released dataset, 33 of which perform better than the strong baseline Cascade-RCNN [2] detector. The top three

Table 3. The AP scores on the VisDrone-DET2019 `testing` set of each object category. $*$ indicates the detection algorithms submitted by the VisDrone Team. The top three results are highlighted in **red**, **green** and **blue** fonts.

Method	ped.	person	bicycle	car	van	truck	tricycle	awn.	bus	motor
DPNet-ensemble (A.15)	**32.31**	**15.97**	**12.86**	51.53	**39.80**	30.66	**30.66**	18.41	38.45	**28.03**
RRNet (A.28)	30.44	14.85	**13.72**	51.43	36.14	**35.22**	28.02	19.00	**44.20**	**25.85**
ACM-OD (A.1)	30.75	**15.50**	10.26	**52.69**	**38.93**	**33.19**	26.96	**21.88**	41.39	**24.91**
S+D (A.31)	31.01	14.54	9.27	**52.51**	**40.36**	31.90	25.77	**21.78**	39.91	22.31
BetterFPN (A.3)	30.23	**16.45**	10.01	51.45	38.85	31.57	26.73	17.79	41.75	24.83
HRDet+ (A.21)	28.60	14.58	**11.71**	49.46	37.13	**35.20**	**28.85**	**21.93**	**43.30**	23.55
CN-DhVaSa (A.9)	**31.50**	13.00	9.08	51.93	38.33	31.15	24.24	21.07	40.94	20.36
SGE-cascade R-CNN (A.30)	29.00	13.51	8.44	51.82	38.00	29.83	25.49	20.67	39.15	22.04
EHR-RetinaNet (A.16)	**30.82**	13.54	8.10	50.00	29.00	30.64	25.39	16.42	41.03	23.52
CNAnet (A.8)	26.19	12.22	6.45	**52.15**	38.29	30.43	22.94	19.55	**42.13**	21.22
FS-Retinanet (A.19)	28.44	14.72	7.20	49.38	36.18	27.97	23.06	18.12	38.97	22.08
CenterNet (A.6)	27.99	11.61	9.02	51.03	36.52	27.88	20.09	19.88	37.71	20.96
Airia-GA-Cascade (A.2)	26.22	12.45	8.67	50.54	38.32	30.62	28.08	19.84	35.97	15.50
GravityNet (A.20)	27.76	11.72	7.97	50.72	36.38	28.04	19.61	18.50	35.23	21.28
Libra-HBR (A.23)	28.53	13.38	6.95	49.58	33.80	25.19	22.86	18.76	37.75	21.24
MSCRDet (A.25)	26.68	11.11	7.28	49.35	34.60	29.20	21.12	19.20	40.44	18.67
DPN (A.14)	23.94	12.80	10.03	43.89	32.43	29.02	**28.45**	20.30	42.56	21.90
TSEN (A.33)	24.36	12.36	8.72	45.08	33.03	28.09	23.35	18.25	36.52	15.08
HTC-drone (A.22)	21.80	11.16	6.23	41.23	32.43	25.52	25.60	18.71	37.17	18.19
TridentNet (A.32)	22.92	9.01	5.24	46.15	30.66	26.70	20.30	16.04	38.93	17.92
CenterNet-Hourglass (A.7)	25.65	9.17	4.54	48.92	30.31	24.64	16.83	14.98	30.46	16.09
retinaplus (A.27)	24.11	9.07	3.49	45.86	24.25	21.30	17.48	12.14	30.12	17.10
ERCNNs (A.18)	18.31	8.28	6.96	41.46	29.00	24.65	18.54	15.43	38.72	15.54
SAMFR-Cascade RCNN (A.29)	23.67	9.75	4.41	41.74	24.22	22.18	16.26	15.01	27.34	15.92
Cascade RCNN+ (A.4)	17.75	5.08	3.54	42.01	26.50	22.58	15.96	12.71	33.28	12.31
EnDet (A.17)	17.19	7.45	2.73	40.16	26.31	18.08	15.42	14.19	31.98	11.50
DCRCNN (A.13)	15.21	8.88	7.06	32.51	25.94	20.40	19.15	15.72	35.65	11.79
Cascade R-CNN++ (A.5)	20.96	7.53	3.05	41.92	21.11	15.34	13.78	10.04	22.38	15.88
ODAC (A.26)	14.24	7.84	4.76	32.37	25.55	21.56	18.27	16.30	37.56	12.83
DA-RetinaNet (A.11)	19.63	7.22	2.76	40.47	22.15	16.95	11.53	9.47	23.74	13.19
MOD-RETINANET (A.24)	17.76	6.79	2.78	40.20	25.87	16.24	12.82	10.08	25.64	11.44
DBCL (A.12)	16.44	5.76	2.45	39.02	22.58	19.86	15.25	10.77	30.66	12.18
ConstraintNet (A.10)	17.49	6.81	2.59	39.17	25.17	14.41	11.02	8.64	18.11	11.84
CornerNet* [18]	20.43	6.55	4.56	40.94	20.23	20.54	14.03	9.25	24.39	12.10
Light-RCNN* [23]	17.02	4.83	5.73	32.29	22.12	18.39	16.63	11.91	29.02	11.93
DetNet59* [24]	15.26	4.07	3.13	36.12	17.29	20.87	13.52	10.45	26.01	10.92
RefineDet* [45]	14.90	3.67	2.02	30.14	16.33	18.13	9.03	10.25	21.93	8.38
RetinaNet* [26]	9.91	2.92	1.32	28.99	17.82	11.35	10.93	8.02	22.21	7.03
FPN* [25]	15.69	5.02	4.93	38.47	20.82	18.82	15.03	10.84	26.72	12.83
Cascade R-CNN* [2]	16.28	6.16	4.18	37.29	20.38	17.11	14.48	12.37	24.31	14.85

detectors are DPNet-ensemble (A.15), RRNet (A.28) and ACM-OD (A.1), achieving 29.62%, 29.13%, and 29.13% APs, respectively. The state-of-the-art detection framework in this challenge can be concluded as "Cascade-RCNN [2]+FPN [25]+Attention Modules". However, it is still far from satisfactory in real applications. We hope our workshop challenge can provide a community-based common platform for evaluation of detection algorithms on drones.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants 61502332, 61876127 and 61732011, Natural Science Foundation of Tianjin Under Grants 17JCZDJC30800, Key Scientific and Technological Support Projects of Tianjin Key R&D Program 18YFZCGX00390 and 18YFZCGX00680 and JD Digits.

Figure 2. The detection results of the winner of the VisDrone-DET2019 Challenge DPNet-ensemble (A.15). The ground-truth and estimated bounding boxes of objects are shown in red and green colors, respectively. Only the categories of some objects are shown for clarity.

A. Submitted Detectors

In the appendix, we summarize the top 33 detectors submitted in the VisDrone2019-DET Challenge, which are ordered alphabetically.

A.1. Augmented Chip Mining for Object Detection (ACM-OD)

Sungeun Hong, Sungil Kang and Donghyeon Cho
{csehong,sung1.kang,cdh12242}@sktbrain.com

ACM-OD is based on Faster R-CNN with FPN [25] in which backbone is ResNet-101 pre-trained on MS COCO [27]. The key distinction of the proposed method is augmented chip mining, which actively utilizes hard examples for training. As the first step, to efficiently localize small objects, we train our model from chip areas [35], *i.e.*, sub-region in which ground truth samples are located densely. We then perform patch-level augmentation to reduce class imbalance issue and high false positive rate. As a result, patch-level augmented object instances, as well as whole images, are used to construct chips consisting of hard examples; and we train our model again by using them. ACM-OD progressively evolves as the training, augmentation, and inference processes are performed iteratively. In addition to active learning with data augmentation scheme, change of scales and aspect radios considering VisDrone dataset improves the accuracy. Masking "ignored regions" or "others" objects during training also helped to improve performance. Furthermore, the post-processing steps including box voting [12] are helpful for accurate localization. Our final resultsare based on the ensemble of the slightly different models. The detector is trained on VisDrone2019 train/val set, DOTA train/val set [42, 8] and MS COCO [27]. More details can be found in the 2019 ICCV workshop paper titled "Patch-level Augmentation for Object Detection in Aerial Images."

A.2. Guided Anchor based Cascade R-CNN (Airia-GA-Cascade)

Yu Zhu and Qiang Chen
zhuyu@airia.cn, qiang.chen@nlpr.ia.ac.cn

Airia-GA-Cascade is Guided Anchor based Cascade R-CNN [2]. It is a multi-stage extension of the popular two-stage R-CNN object detection framework. The goal is to obtain high quality object detection, which can effectively reject close false positives. It consists of a sequence of detectors trained end-to-end with increasing IoU thresholds, to be sequentially more selective against close false positives. The output of a previous stage detector is forwarded to a later stage detector, and the detection results will be improved stage by stage. Our network is based on FPN detector with ResNeXt-101-64 $\times$ 4d backbone and Guided Anchor [38], while adding deformable

convolution [51] in backbone.

A.3. FPN-based Faster R-CNN with Better Training and Testing for Drones (BetterFPN)

Junhao Hu and Lei Jin
{hujh,jinlei}@shanghaitech.edu.cn

BetterFPN uses FPN [25] with ResNet-50 as the backbone of our Faster R-CNN. We use Mask R-CNN [13] pre-trained model (on COCO [27]). We use random cropping and image pyramid in training besides to horizontal flipping. Also, we enlarge every patch by 4 times in training, as well as in testing. We use a better crowd region handling in training. We lower the IoU threshold for matching to crowd region (label ignore) when determine an anchor or proposal as negative sample. Also, we do not optimize the score or bounding box regression when an anchor or proposal matching to crowd region.

A.4. Modified Cascade R-CNN (Cascade R-CNN+)

Jonas Meier, Lars Sommer, Lucas Steinmann and Arne Schumann
{jonas.meier,lars.sommer,lucas.steinmann, arne.schumann}@iosb.fraunhofer.de

Cascade R-CNN+ is based on Cascade R-CNN [2], which consists of a sequence of detectors trained with increasing IoU thresholds. Thus, the detector becomes sequentially more selective against close false positive detections. Feature Pyramid Network (FPN) [25] is used as base detector to account for different object scales. As backbone architecture, we employ SEResNeXt-50 [16]. Furthermore, we employ focal loss as loss function [26]. The anchor box priors are halved compared to the default settings of Cascade R-CNN to account for smaller objects. For testing, the tiled images are scaled by a factor of about 1.25 yielding in tiles of size 736×736 pixels.

A.5. Deformable Cascade R-CNN (Cascade R-CNN++)

Haocheng Han and Jiaqi Fan
hhchyer@gmail.com, garyfan@connect.hku.hk

Cascade R-CNN++ is based on Cascade R-CNN [2], using ResNeXt-101 $64 \times 4d$ as backbone and FPN [25] as feature extractor. We use deformable convolution to enhance our feature extractor. We do not use external data except pre-trained model on COCO dataset. Other techniques like multi-scale training and soft-nms are involved in our method. We use VisDrone train set and fine tune on COCO pre-trained model. The model is trained with four K40 GPUs and mmdetection [5] framework.

A.6. Objects as Points (CenterNet)

Yanchao Li, Zhikang Wang, Yu Heng Toh, Furui Bai and Jane Shen
yanchao.li@u.nus.edu, zkwang00@gmail.com, tohyuheng-@hotmail.com, frbai@stu.xidian.edu.cn, jane.shen@pensees.ai

CenterNet [46] is end-to-end differentiable, simpler, faster, and more accurate than corresponding bounding box based detectors. Firstly, two stack hourglass networks, which are pre-trained models for human pose estimation, are adopted for generating the heat maps. Then, three branches convolutional layers are adopted for generating the center points, offsets and height-width for the proposals. Focal loss and L1 label smoothing loss are used for objects classification and regression respectively. Specifically, we set the image size as 1024×1024 while training and input the original images without resizing for testing.

A.7. Objects as Points (CenterNet-Hourglass)

Da Yu, Lianghua Huang, Xin Zhao and Kaiqi Huang
yuda@hit.edu.cn, huanglianghua2017@ia.ac.cn, {xzhao,kaiqi.huang}@nlpr.ia.ac.cn

CenterNet-Hourglass is based on CenterNet [46]. Due to the large number of objects in each image in the VisDrone dataset, we increase the maximum of instances that per image can be detected.

A.8. Convolution Neighbor Aggregation Detector for multi-scale detection (CNAnet)

Keyang Wang and Lei Zhang
{wangkeyang,leizhang}@cqu.edu.cn

CNAnet uses the Convolution Neighbors Aggregation Detector for multi-scale detection. Specifically, we use the Multi-neighbor layers fusion modules to fuse the current layer with its Multi-neighbor higher layers, we call this process as backward augmentation. Then we forward propagate the enhanced feature to high-level layers, we call this process as forward augmentation. In order to improve the accuracy with multi-scale instances, we use the multi-scale test for the final inference. But we use the single network.

A.9. CenterNet-Hourglass-104 (CN-DhVaSa)

Dheeraj Reddy Pailla, Varghese Alex Kollerathu and Sai Saketh Chennamsetty
dheerajreddy.p@students.iiit.ac.in, varghese.kollerathu@siemens.com, sai.chennamsetty@siemens.com

CN-DhVaSa is derived from the original CenterNet [46]. During the training phase, images are resized to 1024×1024 and the batch size was set to 8. During inference, the multi-scale strategy is used to increase the performance. An image with dimension of 2048×2048 is resized based on different scales factors, *i.e.*, $0.5, 0.75, 1, 1.25, 1.5$. After that, a confidence threshold of 0.25 is used to weed out the false detections.

A.10. Constraint Keypoint Triplets for Object Detection (ConstraintNet)

Dening Zeng, Di Li
dnzeng@stu.xidian.edu.cn

ConstraintNet is buit upon a one-stage keypoint-based detector named CenterNet. Our approach detects each object by restricting their width and height, which improves both precision and recall. First, we design a customized modules named constraint corner pooling, which convolution kernel is depending on constrain boundary of each object. Constraint corner pooling play the roles of extracting features around the target rather than the whole image. Second, the prediction box which boundary is greater than constrain boundary will be abandoned. Last, observing that the corner localization accuracy is gradually refined during multi-stage, we adopt a coarse-to-fine supervision strategy in accordance [20]. Overview of ConstrainNet. A covolutional backbone network applies three corners prediction modules to output heatmaps, embeddings and offsets, respectively. Similar to CenterNet, a triplet of corners and the similar embeddings used to detect a potential bounding box. Then the constrain boundary is used to determine the final bounding boxes.

A.11. RetinaNet with Convolutional Block Attention Module (DA-RetinaNet)

Jingjing Xu and Dechun Cong
{1017010628,1017010643}@njupt.edu.cn

DA-RetinaNet is based on the Focal Loss for Dense Object Detection [26]. The main changes we made are concluded as follows: (1) We use more scales of smaller anchors to detect low-resolution objects; (2) A RetinaNet with ResNet-101 pre-trained weights on ImageNet as the backbone is used, specifically, the features from Conv2_x are also used to detect objects. (3) To further improve the detection accuracy, we add the dual attention mechanism. That is, we use additional channel attention module and spatial attention module from Convolutional Block Attention Module (CBAM) [39] to learn channel attention and spatial attention. However, different from the original CBAM, our method uses both attention modules in parallel, and the output feature vectors are merged by using element-wise summation.

A.12. Detection based on coarse-to-fine labeling (DBCL)

Wei Dai and Weiyang Wang
daiwei@co-mall.com, weiyang.wang@snowcloud.ai

DBCL is based on Segmentation Is All You Need [40], which uses weakly supervised multimodal annotation segmentation (WSMA-Seg) to achieve an accurate and robust object detection without NMS. In WSMA-Seg, multimodal annotations are proposed to achieve an instance-aware segmentation using weakly supervised bounding boxes; we also develop a run-data-based following algorithm to trace contours of objects. In addition, we propose a multi-scale pooling segmentation (MSP-Seg) as the underlying segmentation model of WSMA-Seg to achieve a more accurate segmentation and to enhance the detection accuracy of WSMA-Seg.

A.13. Deformable Cascade-RCNN (DCRCNN)

Almaz Zinollayev and Anuar Askergaliyev
{almaz.zinollayev,anuar.askergaliyev}@btsdigital.kz

DCRCNN adopts Cacade-RCNN [2] as the baseline, and builds additional blocks on top of it. We add deformable convolutional neural networks, as well as attention mechanisms to our backbone. We trained using SyncBN for batch normalizaiton. We use random cropping and color jittering augmentation for our training. For inference we ensebmle best checkpoints of trained network. We also use TTA and Soft-NMS for inference.

A.14. Double Pyramid Network (DPN)

Nuo Xu, Xin Zhang, Binjie Mao, Chunlei Huo and Chunhong Pan
{nuo.xu,xin.zhang2018,binjie.mao,clhuo,
chpan}@nlpr.ia.ac.cn

DPN is a double pyramid network model based on Cascade R-CNN [2], which consists of image pyramid and feature pyramid. Because of the downsampling operation in the network, the information loss of most small targets is serious. Image pyramids are used to generate input images of different scales, and normalize each object scale to a fixed range to reduce the missed detection rate of minimal and maximal objects. Feature pyramids are used to generate feature maps of different scales. Multi-scale feature fusion technology enhances the features of feature maps at each level. The obtained feature pyramid is used as the learned image feature to detect the objects in Cascade R-CNN. Multi stage boundary box regression and classification with Cascade R-CNN make the detection results more accurate.

A.15. Drone Pyramid Networks-ensemble (DPNet ensemble)

Hongliang Li, Qishang Cheng, Heqian Qiu, Zichen Song and Xiaoyu Chen
{hlli,cqs,hqqiu,}@std.uestc.edu.cn, szc.uestc@gmail.com, xychen9459@gmail.com

DPNet-ensemble trains only two object detectors based on Cascade-RCNN [2] by mmdetection [5] deep learning framework. The design of our detectors follows the idea of FPN [25], whose feature extractors are ResNet-50 and ResNet-101 [14] which are pre-trained on ImageNet. In order to make full use of the ability of feature extraction, we introduce the global context module (GC) [3] and deformable convolution (DC) [7] into the backbone network. To make the most of the data, we train Cascade-RCNN with FPN using multiple scales ($1000, 800, 600$ for the short edge) to naturally handle objects of various sizes. We use nms to select predicted boxes. We changed RoIPooling to RoIAlign [13] to do feature quantification. In the training phase, we use multi-scale training and the balance strategy used in Libra R-CNN [30]. In the inference phase, we use Multi-scale testing.

A.16. Enhanced High Resolution RetinaNet (EHR-RetinaNet)

Jaekyum Kim, Byeongwon Lee and Jun Won Choi
jkkim@spa.hanyang.ac.kr, bwon.lee@sk.com, junwchoi@hanyang.ac.kr

RetinaNet+ is improved from the RetinaNet [26]. We use the most powerful SE-ResNeXt-101 [16] as the backbone network and change the anchor boxes to detect the tiny objects. Furthermore, we use the many data augmentation strategies including distortion, sample crop, mirror and expansion. The training input size is 1728×3072 and the test input size 2160×3840.

A.17. Ensemble deep object detector based on graph clique for VisDrone2019 Challenge (EnDet)

Pengyi Zhang and Yunxin Zhong
zhangpybit@gmail.com, bityunxinz@gmail.com

EnDet is an ensemble of two yolov3-based [32] networks and two faster R-CNN [33] based networks. First, we modify yolov3 network with spatial pyramid pooling (spp) module. Specifically, we add one spp module to the first yolo branch network of yolov3 to implement yolov3-spp1 and one spp module to each of the three yolo branch networks to get yolov3-spp3. Second, we add residual attention module to feature pyramid network (FPN) (called ra-fpn) in faster R-CNN. Finally, we built EnDet by combining yolov3-spp1, yolov3-spp3, faster R-CNN with resnet101-fpn backbone and modify faster R-CNN with resnet101-ra-fpn backbone through an ensemble method [44] based on graph clique.

A.18. Ensemble of RCNNs (ERCNNs)

Jihoon Lee
jihoon.lee@kakaobrain.com

ERCNNs is generated by ensemble of the following object detection models: Cascade R-CNN with ResNeXt-101, Faster R-CNN with ResNeXt-101, Faster R-CNN with ResNet-50 and deformable convolution, Faster R-CNN with resNet-50 and spatial attention mechanism, Faster R-CNN with ResNet-50, deformable convolution and spatial attention mechanism.

A.19. Feature Selected RetinaNet (FS-Retinanet)

Ziming Liu, Jing Ge, Tong Wu, Lin Sun and Guangyu Gao
liuziming.email@gmail.com,
{398817430,547636024}@qq.com,
lin1.sun@samsung.com, guangyugao@bit.edu.cn

FS-Retinanet is improved from RetinaNet [26], using the ResNeXt as backbone [14]. There are several differences compared with the original RetinaNet. 1) To reduce GPU memory, we only use P2,P4,P6 of Feature Pyramid Network (FPN) [25]. 2) We add feature selected anchor-free head (FSAF) [47] into RetinaNet, which improves the performance significantly. Thus there are one anchor head and one anchor free head in our model. Next, we will describe some details of the proposed detection pipeline. Most importantly, we perform several data augmentations before model training. Firstly, each original Images is cropped into 4 patches, while each patch is rescaled to 1920×1080, and we also propose an online algorithm to obtain sub-images. Secondly, the Generative Adversarial Network is used to transform the image of the day into the night, which reduces the unbalance of day and night samples. After that, the overall model is composed of 4 parts, including the ResNet backbone, the FPN network, and the FSAF module as well as the retina head. Finally, we train the model with an end-to-end way and test on multi-scales data to obtain better results. In addition, we also fuse multi-models to improve performance.

A.20. Gravitational Centroid Points based Network (GravityNet)

Toh Yu Heng and Harry Nguyen
tohyuheng@hotmail.com, harry.nguyen@glasgow.ac.uk

GravityNet is derived from CenterNet [46] which uses the

center of mass of each object to produce key points for detection. Our algorithm includes the occlusion details of bounding boxes from the VisDrone2019 dataset during training. Category-specific heat maps are produced by generating gravitational centroid points to capture necessary details of objects for better performance.

A.21. Improved high resolution detector (HRDet+)

Jingkai Zhou, Weida Qin, Qiong Liu and Haitao Xiong
{201510105876,201530061442}@mail.scut.edu.cn,
liuqiong@scut.edu.cn, 201821038528@mail.scut.edu.cn

HRDet+ is improved from HRNet [36]. The model maintains high-resolution representations through the whole process by connecting high-to-low resolution convolutions in parallel and produces strong high-resolution representations by repeatedly conducting fusions across parallel convolutions. The code and models have been publicly available at `https://github.com/HRNet`. Beyond this, we modify HRNet by introducing a guided attention neck and propose a harmonized online hard example mining strategy to sample data. At last, HRDet+ is trained on multi-scale data, and the model assemble is also adopted.

A.22. Hybrid Task Cascade for Drone Object Detection (HTC-drone)

Xindi Zhang
xindi.zhang@qmul.ac.uk

HTC-drone is improved from the Hybrid Task Cascade (HTC) model [4]. The changes are: (1) Each training images are cropped into four small parts. (2) One model is trained based on pedestrian, person and car class, and another model is trained based on other classes. They are combined at testing time. (3) The NMS is replaced by soft-NMS. We use ResNet50 as backbone with COCO pre-trained model. (4) The testing set is cropped, and the final result is the combination of different cropped images with NMS.

A.23. Hybrid model based on Improved SNIPER, Libra R-CNN and Cascade R-CNN (Libra-HBR)

Chunfang Deng, Shuting He, Qinghong Zeng, Zhizhao Duan and Bolun Zhang
{dengcf,shuting_he,zqhzju,21825106}@zju.edu.cn,
zh98ang@163.com

Libra-HBR is an ensemble of improved SNIPER [35], Libra R-CNN [30] and Cascade R-CNN [2]. It is proved to generalize very well in various weather and light conditions in real-world drone images, especially for small objects. SNIPER presents an algorithm for performing efficient multi-scale training in instance level visual recognition tasks. We replace Faster-RCNN detection framework in SNIPER with deformable ResNet-101 FPN structure, which introduce additional context in object detection and improve accuracy in small objects. We use the max-out operation for classification, to kill false positive proposals brought by dense small anchors. On the other hand, we apply Cascade R-CNN to solve IoU threshold selection problem. We use ResNext-101 as the backbone network and use Libra R-CNN to get the better performance. Moreover, we add deformable convolutional network [7], attention mechanism [3], weight standardization [31] and group normalization [41]. In the above mentioned models, we use balanced-data-augmentation, and adapt the anchor size during training time. To further boost the performance, We add bag of tricks during testing steps, including Soft-NMS, multi-scale detection, flip detection and crop detection. Finally, we use bounding box voting to integrate above two novel models to obtain higher performance.

A.24. Modified Retinanet (MOD-RETINANET)

Aashish Kumar, George Jose and Srinivas S S Kruthiventi
{aashish.kumar,george.jose,srinivas.sai}@harman.com

MOD-RETINANET is a modified version of RetinaNet [26] with the ResNet-50 backbone. We have adapted the algorithm for small objects which is required for drone Images. We add additional TAPs to the FPN [25]. We use a modified version of RetinaNet with additional TAPs to detect small objects. We trained multiple models by varying learning rate, batch-size, prior-box (size & stride) and also varying the complexity of the backbone as well as the regression and classification sub-models.

A.25. Multi-Scale Object Detector Based on Cascade R-CNN (MSCRDet)

Xin Chen, Chang Liu, Shuhao Chen, Xinyu Zhang, Dong Wang, Huchuan Lu
{chenxin3131,lcqctk0914,shuhaochn,
zhangxy71102}@mail.dlut.edu.cn,
{wdice, lhchuan}@dlut.edu.cn

MSCRDet uses Cascade R-CNN [2] with three stages as the basic structure. We add FPN [25] to deal with the various object scales, especially for small object detection. We use P3 tp P6 feature maps for RPN. In addition to this, taking account into the fact that the scene taken by drone has lots of small objects and dense object distribution, we replace the RoIPooling [33] with RoIAlign [13], for RoIPooling [33] performs coarse spatial quantization for feature extraction while RoIAlign [13]

preserves more accurate spatial location information. We use Soft-NMS [1] rather than NMS for better recall in the scene of dense object distribution. Moreover, we adopt ResNeXt-101($64 \times 4d$) as the strong backbone.

A.26. Object Detection in Aerial Images Using Adaptive Cropping (ODAC)

Junyi Zhang, Junying Huang, Xuankun Chen and Dongyu Zhang
{zhangjy329,huangjy229,chenxk3}@mail2.sysu.edu.cn,
zhangdy27@mail.sysu.edu.cn

ODAC is a simple and effective framework based on Faster RCNN [33] and FPN [25]. First, based on the prior knowledge of the preliminary trained object detection model, we propose an adaptive cropping method based on an difficult region estimation network to enhance the detection of the difficult targets, which allows the detection model to fully exploit its performance. Second, we use the well-trained adaptive region estimation network to generate more diverse and representative images, which is effective in enhancing the training data. In addition, in order to alleviate the imbalance problem during training, we adopted the IoU-balanced sampling method [30] and the balanced L1 loss [30] as the box regression loss in our framework. At the time of the test, we selected the top 5 difficult regions with the highest scores predicted as the final difficulty regions and tested them again, then merged the results.

A.27. Retinanet Plus (retinaplus)

Zikai Zhang and Peng Wang
andychang@mail.nwpu.edu.cn, peng.wang@nwpu.edu.cn

The retinaplus detector is based on RetinaNet [26]. For small target detection, it is improved by data enhancement, multi-scale training and multi-scale testing.

A.28. Re-RegressionNet (RRNet)

Changrui Chen, Yu Zhang, Qingxuan Lv, Xiaorui Wang, Shuo Wei and Xin Sun
{ccr,zhangyu9520,lqx,recyclerblacat,
weishuo}@stu.ouc.edu.cn,sunxin@ouc.edu.cn

RRNet is inspired by [46], which uses a convolutional neural network to predict the center point and the size of the object of each class. We use these points and size predictions to generate the coarse bounding boxes. After that, we send these coarse bounding boxes to a Re-Regression Module (RR), which consists of a ROIAlign module [13] and some convolution layers. The RR Module predicts the bias between the coarse bounding boxes and the GT. It only refines the position of these coarse bounding boxes. Finally, we use the soft-nms to generate the final bounding boxes.

A.29. Spatial Attention for Multi-scale Feature Refinement Based on Cascade RCNN (SAMFR-Cascade RCNN)

Haoran Wang, Zexin Wang, Meixia Jia, Aijin Li and Tuo Feng
wanghaoran@stu.xidian.edu.cn,
zexinwang2016@gmail.com,
{mxjia,aijinli,fengt}@stu.xidian.edu.cn

SAMFR-Cascade RCNN uses Cascade R-CNN [2] as the base network to continuously optimize the prediction results by cascading several detection networks. Different from normal cascade, several detection networks of Cascade R-CNN are trained on positive and negative samples determined by different IOU thresholds. On the basis of this model, we add the deformable convolution (DCN) layer, which can actively learn the object area under the guidance of ground truth, so as to change the sampling location of convolution filter to achieve more accurate position of the target, and obtain more representative characteristics. For high-level features, we point that the neural network gets more and more information of the central parts of the object, but the edge information of the object has a great influence on object locating. For this problem, we design a novel Spatial-Refinement Module based on the attention mechanism to repair the edge details in the multi-scale features. Meanwhile, downsampling serves well in classification task, which is, however, not necessarily beneficial for object detection because localization may suffer from the absence of the global location information, and the proposed RFEB tries to address the problem. In the end, we adopt contour detection algorithm and multi-model fusion for post processing.

A.30. cascade R-CNN with SGE backbone (SGE-cascade R-CNN)

Xudong Wei, Hao Qi, Wanqi Li and Guizhong Liu
{wxd6994,qihao456,wanqili,liugz}@stu.xjtu.edu.cn

SGE-cascade R-CNN is improved from cascade R-CNN [2]. What we change is that we use SGE [21] block and ResNet50 [5] as our backbone. SGE makes each set of features robust and well-distributed over the space, and models a spatial enhance mechanism inside each feature group, by scaling the feature vectors over all the locations with an attention mask.

A.31. Segmentation + Detection (S+D)

Yifu Chen
chenyifu@stu.hit.edu.cn

S+D consists of two steps, segmentation and detection. In the segmentation step, the model outputs low accuracy heat map (smaller size than original images), which is used to generate several regions that might exist objects. In the detection step, common detection model is used to detect objects in the regions cropped from original images. After these two main steps, some post-processings are needed to merge boxes from regions to the full image. In the first step, ASPP module in DeepLab [43] was used to predict the class agnostic heatmap of the image. Pixel level segmentation heat map is far beyond the need of low accuracy region generation. Several modifications are made in ASPP including reducing branches in ASPP module and out channels of the last convolution layer. MobileNet v2 [34] is used as the model backbone because this step very low demand for model capacity and speed is also important in region generation. Hierarchical clustering and image morphology method are respectively used to deal with the small regions and big regions. In the second step, Cascade RCNN [2] with HRNet [36] as backbone from Open MMLab Detection Toolbox was used as the detection model and no modifications were made. After detection per region, NMS is also needed because there is overlapping in regions.

A.32. Scale-Aware Trident Networks for Object Detection (TridentNet)

Xuzhang Zhang
shantan@hust.edu.cn

TridentNet [22] aims to generate scale-specific feature maps with a uniform representational power. We use a parallel multi-branch architecture in which each branch shares the same transformation parameters but with different receptive fields. Then, we use a scale-aware training scheme to specialize each branch by sampling object instances of proper scales for training. Other techniques like multi-scale training and soft-nms are involved in our method. The model is trained with one 2080Ti GPU and SimpleDet [6] framework.

A.33. TwoStage ENsembles (TSEN)

Zhifan Zhu and Zechao Li
{zhifanzhu,zechao.li}@njust.edu.cn

TSEN uses ensembles of 3 two-stage methods: vanilla Faster R-CNN [33], Guided Anchoring [38] and Libra-RCNN [30]. Note that all the backbones are ResNeXt101 with deformable convolution from C2 to C5. We replace classification cross-entropy loss with bootstrap loss in Libra-RCNN, which increases AP about 1% in validation. For vanilla Faster R-CNN with FPN [25], we add anchor with aspect ratio 0.8 to each FPN level. We train the model on random cropped patch with 640×640 pixels, and the image is scaled to 1024×1024 during training. During inference, we use multi-crop patches on test images ($640, 768, 896, 1024$ pixels), and merge the result with NMS (0.33 threshold).

References

[1] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-nms - improving object detection with one line of code. In *ICCV*, pages 5562–5570, 2017.

[2] Z. Cai and N. Vasconcelos. Cascade R-CNN: delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018.

[3] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. *CoRR*, abs/1904.11492, 2019.

[4] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. Hybrid task cascade for instance segmentation. In *CVPR*, 2019.

[5] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. Mmdetection: Open mmlab detection toolbox and benchmark. *CoRR*, abs/1906.07155, 2019.

[6] 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. *CoRR*, abs/1903.05831, 2019.

[7] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017.

[8] J. Ding, N. Xue, Y. Long, G. Xia, and Q. Lu. Learning roi transformer for detecting oriented objects in aerial images. *CoRR*, abs/1812.00155, 2018.

[9] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, and Q. Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *ECCV*, pages 375–391, 2018.

[10] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. Centernet: Keypoint triplets for object detection. *CoRR*, abs/1904.08189, 2019.

[11] K. Duan, D. Du, H. Qi, and Q. Huang. Detecting small objects using a channel-aware deconvolutional network. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[12] S. Gidaris and N. Komodakis. Object detection via a multi-region and semantic segmentation-aware CNN model. In *ICCV*, pages 1134–1142, 2015.

[13] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[15] M. Hsieh, Y. Lin, and W. H. Hsu. Drone-based object counting by spatially regularized regional proposal network. In *ICCV*, 2017.

[16] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018.

[17] T. Kong, F. Sun, H. Liu, Y. Jiang, and J. Shi. Foveabox: Beyond anchor-based object detector. *CoRR*, abs/1904.03797, 2019.

[18] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 765–781, 2018.

[19] H. Law, Y. Teng, O. Russakovsky, and J. Deng. Cornernet-lite: Efficient keypoint based object detection. *CoRR*, abs/1904.08900, 2019.

[20] W. Li, Z. Wang, B. Yin, Q. Peng, Y. Du, T. Xiao, G. Yu, H. Lu, Y. Wei, and J. Sun. Rethinking on multi-stage networks for human pose estimation. *CoRR*, abs/1901.00148, 2019.

[21] X. Li, X. Hu, and J. Yang. Spatial group-wise enhance: Improving semantic feature learning in convolutional networks. *CoRR*, abs/1905.09646, 2019.

[22] Y. Li, Y. Chen, N. Wang, and Z. Zhang. Scale-aware trident networks for object detection. *CoRR*, abs/1901.01892, 2019.

[23] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Light-head R-CNN: in defense of two-stage object detector. *CoRR*, abs/1711.07264, 2017.

[24] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Detnet: A backbone network for object detection. *CoRR*, abs/1804.06215, 2018.

[25] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, 2017.

[26] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, pages 2999–3007, 2017.

[27] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014.

[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. In *ECCV*, pages 21–37, 2016.

[29] M. Mueller, N. Smith, and B. Ghanem. A benchmark and simulator for UAV tracking. In *ECCV*, pages 445–461, 2016.

[30] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin. Libra R-CNN: towards balanced learning for object detection. In *CVPR*, 2019.

[31] S. Qiao, H. Wang, C. Liu, W. Shen, and A. L. Yuille. Weight standardization. *CoRR*, abs/1903.10520, 2019.

[32] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[33] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015.

[34] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018.

[35] B. Singh, M. Najibi, and L. S. Davis. SNIPER: efficient multi-scale training. In *NeurIPS*, pages 9333–9343, 2018.

[36] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[37] Z. Tian, C. Shen, H. Chen, and T. He. FCOS: fully convolutional one-stage object detection. *CoRR*, abs/1904.01355, 2019.

[38] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin. Region proposal by guided anchoring. In *CVPR*, 2019.

[39] S. Woo, J. Park, J. Lee, and I. S. Kweon. CBAM: convolutional block attention module. In *ECCV*, pages 3–19, 2018.

[40] Y. Wu, Z. Cheng, Z. Xu, and W. Wang. Segmentation is all you need. *CoRR*, abs/1904.13300, 2019.

[41] Y. Wu and K. He. Group normalization. In *ECCV*, pages 3–19, 2018.

[42] G. Xia, X. Bai, J. Ding, Z. Zhu, S. J. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang. DOTA: A large-scale dataset for object detection in aerial images. In *CVPR*, pages 3974–3983, 2018.

[43] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. Learning a discriminative feature network for semantic segmentation. In *CVPR*, pages 1857–1866, 2018.

[44] P. Zhang, X. Li, and Y. Zhong. Ensemble mask-aided r-cnn. In *ISBI*, pages 6154–6162, 2018.

[45] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. Single-shot refinement neural network for object detection. In *CVPR*, pages 4203–4212, 2018.

[46] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. *CoRR*, abs/1904.07850, 2019.

[47] C. Zhu, Y. He, and M. Savvides. Feature selective anchor-free module for single-shot object detection. *CoRR*, abs/1903.00621, 2019.

[48] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu. Vision meets drones: A challenge. *CoRR*, abs/1804.07437, 2018.

[49] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, and et al. Visdrone-det2018: The vision meets drone object detection in image challenge results. In *ECCVW*, pages 437–468, 2018.

[50] R. Zhu, S. Zhang, X. Wang, L. Wen, H. Shi, L. Bo, and T. Mei. Scratchdet: Exploring to train single-shot object detectors from scratch. In *CVPR*, 2019.

[51] X. Zhu, H. Hu, S. Lin, and J. Dai. Deformable convnets v2: More deformable, better results. *CoRR*, abs/1811.11168, 2018.