VisDrone-DET2020: The Vision Meets DroneObject Detection in Image Challenge Results

by

Faizan Farooq Khan, Dheeraj Reddy Pailla, Sarvesh Mehta

Report No: IIIT/TR/2020/-1

Centre for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
September 2020

VisDrone-DET2020: The Vision Meets Drone Object Detection in Image Challenge Results

Dawei Du[1], Longyin Wen[2], Pengfei Zhu[3], Heng Fan[4], Qinghua Hu[3], Haibin Ling[4], Mubarak Shah[5], Junwen Pan[3], Apostolos Axenopoulos[34], Arne Schumann[31], Athanasios Psaltis[34], Ayush Jain[28], Bin Dong[36], Changlin Li[14], Chen Chen[14], Chengzhen Duan[7], Chongyang Zhang[33], Daniel Stadler[30], Dheeraj Reddy Pailla[22], Dong Yin[16], Faizan Khan[22], Fanman Meng[6], Guangyu Gao[15], Guosheng Zhang[17], Hansheng Chen[16], Hao Zhou[33], Haonian Xie[25], Heqian Qiu[6], Hongliang Li[6], Ioannis Athanasiadis[34], Jincai Cui[35], Jingkai Zhou[11], Jonghwan Ko[9], Joochan Lee[9], Jun Yu[25], Jungyeop Yoo[9], Lars Wilko Sommer[31], Lu Xiong[16], Michael Schleiss[21], Ming-Hsuan Yang[12], Mingyu Liu[6], Minjian Zhang[6], Murari Mandal[29], Petros Daras[34], Pratik Narang[28], Qiong Liu[11], Qiu Shi[20], Qizhang Lin[15], Rohit Ramaprasad[28], Sai Wang[36], Sarvesh Mehta[22], Shuai Li[13], Shuqin Huang[11], Sungtae Moon[8], Taijin Zhao[6], Ting Sun[24], Wei Guo[35], Wei Tian[16], Weida Qin[11], Weiping Yu[14], Wenxiang Lin[15], Xi Zhao[23], Xiaogang Jia[27], Xin He[32], Xingjie Zhao[24], Xuanxin Liu[18], Yan Ding[15], Yan Luo[33], Yang Xiao[13], Yi Wang[23], Yingjie Liu[26], Yongwoo Kim[10], Yu Sun[19], Yuehan Yao[36], Yuyao Huang[16], Zehui Gong[17], Zhenyu Xu[36], Zhipeng Luo[36], Zhiguo Cao[13], Zhiwei Wei[7], Zhongjie Fan[11], Zichen Song[6], Ziming Liu[15]

[1]Kitware, Inc., Clifton Park, NY, USA.
[2]JD Finance America Corporation, Mountain View, CA, USA.
[3]Tianjin University, Tianjin, China.
[4]Stony Brook University, New York, NY, USA.
[5]University of Central Florida, Orlando, FL, USA.
[6]University of Electronic Science and Technology of China, Chengdu, China
[7]Harbin Institute of Technology (Shenzhen), Shenzhen, China.
[8]Korea Aerospace Research Institute, Daejeon, South Korea.
[9]Sungkyunkwan University, Suwon, South Korea.
[10]Sangmyung University, Cheonan, South Korea.
[11]South China University of Technology, Guangzhou, China.
[12]University of California at Merced, Merced, CA, USA.
[13]Huazhong University of Science and Technology, Wuhan, China.
[14]University of North Carolina at Charlotte, Charlotte, NC, USA.
[15]Beijing Institute of Technology, Beijing, China.
[16]Tongji University, Shanghai, China.
[17]Guangdong University of Technology, Guangzhou, China.
[18]Beijing Forestry University, Beijing, China.
[19]Beihang University, Beijing, China.
[20]Beijing Vion Technology, Inc., Beijing, China.
[21]Fraunhofer FKIE, Wachtberg, Germany.
[22]International Institute of Information Technology, Hyderabad, Hyderabad, India.
[23]Xidian University, Xi'an, China.
[24]Xi'an Jiaotong University, Xi'an, China.

[25]Science and Technology of China, Hefei, China.
[26]North University of China, Taiyuan, China.
[27]Harbin Institute of Technology, Harbin, China.
[28]Birla Institute of Technology and Science, Pilani, Pilani, India.
[29]Malaviya National Institute of Technology Jaipur, Jaipur, India.
[30]Karlsruhe Institute of Technology, Karlsruhe, Germany.
[31]Fraunhofer IOSB, Karlsruhe, Germany.
[32]Wuhan University of Technology, Wuhan, China.
[33]Shanghai Jiao Tong University, Shanghai, China.
[34]Centre for Research and Technology Hellas, Thessaloniki, Greece.
[35]Chongqing University, Chongqing, China.
[36]DeepBlue Technology (Shanghai) Co., Ltd, Shanghai, China.

Abstract. The Vision Meets Drone Object Detection in Image Challenge (VisDrone-DET 2020) is the third annual object detector benchmarking activity. Compared with the previous VisDrone-DET 2018 and VisDrone-DET 2019 challenges, many submitted object detectors exceed the recent state-of-the-art detectors. Based on the selected 29 robust detection methods, we discuss the experimental results comprehensively, which shows the effectiveness of ensemble learning and data augmentation in drone captured object detection. The full challenge results are publicly available at the website http://aiskyeye.com/leaderboard/.

Keywords: drone, object detection, evaluation

1 Introduction

Object detection is a hot topic in computer vision community, which propels various industrial detection-based applications such as autonomous driving, anomaly detection, face detection and activity recognition. Although much progress has been made using deep learning based methods, it is still a difficult problem on real-world scenarios.

Despite of detection in general scenarios, our goal is to advance state-of-the-art object detection approaches on drone-captured scenes, which involves some unique challenging factors (*e.g.*, view point change, scale variation, occlusion and background clutter) in object detection. The studies are seriously limited by the lack of public drone based large-scale benchmarks. Following VisDrone-DET2018 Challenge [47] and VisDrone-DET2019 Challenge [8], we held the 3rd Vision Meets Drone Object Detection in Images Challenge (VisDrone-DET2020) on August 28, 2020, in conjunction with the 16-th European Conference on Computer Vision (ECCV 2020).

In this paper, we summarize 29 object detection algorithms submitted to this challenge, and provide a comprehensive performance evaluation for them. Theses algorithms are improved based state-of-the-art detectors that are recently published in top computer vision conferences or journals, *e.g.*, Cascade R-CNN [3],

CenterNet [45, 9], ATSS [42], YOLOv3 [28] and RetinaNet [21]. Specifically, there are 10 out of 29 detection methods that outperform the previous winners in VisDrone-DET2018 and VisDrone-DET2019. The complete experimental results can be found at our website http://www.aiskyeye.com/, which is useful to further promote the research on object detection on drone-captured scenes.

2 Related Work

With fast development of various effective detection framework, researchers focus on ensemble of complex models to improve the performance. Besides, it is crucial to apply dat augmentation strategies to train the deep model if lack of training data. In the following, we briefly review the current ensemble learning and data augmentation strategies in object detection field.

2.1 Ensemble Learning

Ensemble learning contains several feature extractors from different backbones in parallel, which requires additional time cost to improve the accuracy. In [16], several detection models are ensembled to achieve the state-of-the-art performance on the 2016 COCO object detection challenge. Xu *et al.* [40] employ the Single Shot MultiBox Detector [23] as the backbone and combine ensemble learning with context modeling and multi-scale feature representation. Besides, Gao *et al.* [11] incorporate an ensemble of classification heads for both box predictor and region proposal predictor to reduce false positives of the mined bounding boxes. To reduce the computation cost, Chen and Shrivastava [6] develop the Group Ensemble Network that incorporates an ensemble of ConvNets in a single ConvNet by a shared-base and multi-head structure.

2.2 Data Augmentation

Inspired by image classification field, random cropping and multi-scale training are the most widely used among the data augmentation strategies for object detection. Some methods randomly erase or add objects to the image for improved accuracy [43, 10]. Despite of hand-tuned ranges, Zoph *et al.* [49] investigate the application of a learned data augmentation policy on object detection performance. In [1], a comprehensive experiment is conducted and shows that CutMix [41] and Mosaic data augmentation is effective in YOLOv4 detector. Notably, Mosaic is a new data augmentation method that mixes 4 training images with different contexts, which allows detection of objects outside their normal context.

3 The VisDrone-DET2020 Challenge

3.1 Dataset

As shown in Fig. 1, we use the same dataset as that in the previous two challenges [47, 8] for a fair comparison. Specifically, the challenge contains 6,471

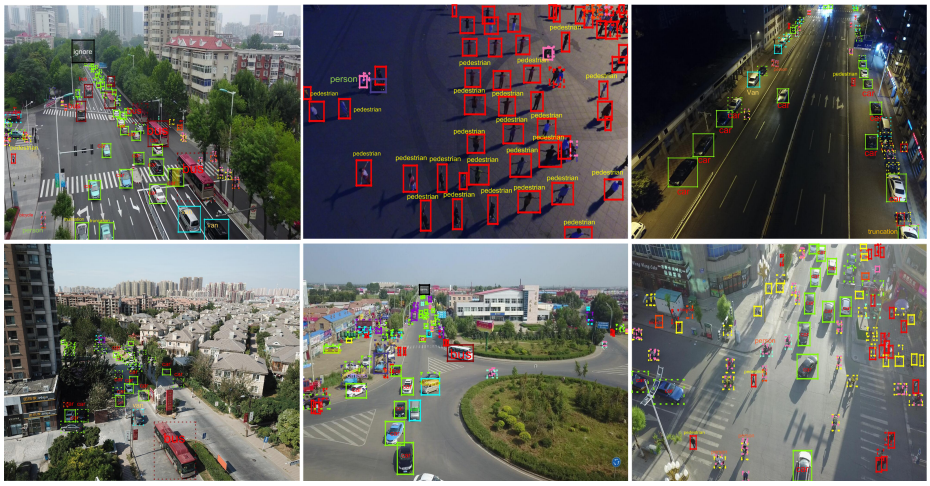

Fig. 1. Annotation exemplars in the VisDrone-DET2020 challenge. The dashed bounding box indicates occlusion of the object, and different colors indicate different categories of objects. Only some attributes are displayed for clarity.

images for training and 548 images for validation, and 3,190 images for testing. Among the testing set, we have 1,580 images in the **test-challenge** subset for workshop competition, and 1,610 images in the **test-dev** subset for public evaluation. Ten object categories are pre-defined, *i.e.*, *pedestrian*, *person*, *car*, *van*, *bus*, *truck*, *motor*, *bicycle*, *awning-tricycle*, and *tricycle*. Some rarely occurring special vehicles (*e.g.*, *machineshop truck*, *forklift truck*, and *tanker*) are ignored in evaluation.

The participators are required to submit the detection results of specific algorithm with detailed description to the evaluation server no more than 10 times. The best submission among ten times are used as the final result. We encourage the participants to use the provided training data, while also allow them to use additional training data. The use of external data must be indicated during submission. For a fair comparison, we rank the algorithms trained on external VisDrone test-dev set in the leaderboard individually. Notably, it is strictly forbidden to submit the same algorithm by different accounts. The teams that provide better performance than CornerNet [17] are offered co-authorship of this results paper.

For evaluation, we follow the MS COCO evaluation protocol [22] to rank the detection algorithms, *i.e.*, AP, AP50, AP75, AR1, AR10, AR100 and AR50 metrics. Specifically, AP is the primary metric and calculated by averaging over all Intersection over Union (IoU) thresholds in the range $[0.50, 0.95]$ with the uniform step size 0.05 of all 10 object categories. AP50 and AP75 are the average precision at the IoU threshold of 0.50 and 0.75 respectively. Besides, we compute average recalls given 1, 10, 100 and 500 detection per image over all object categories and IoU thresholds.

3.2 Submission

We received 85 submissions from all over the world in the VisDrone-DET2020 Challenge, where 29 teams from 31 different research institutes developed robust detectors better than the state-of-the-art detection method CornerNet [17]. The VisDrone committee also reports the results of another 3 detectors including Light-RCNN [19], FPN [20] and Cascade R-CNN [3].

Among all the submissions, several top methods use ensemble model to improve the accuracy, *i.e.*, DPNetV3 (A.1), SMPNet (A.2), and ECascade R-CNN (A.8). Eleven algorithms are based on Cascade R-CNN [3] with various effective modules, including DBNet (A.3), DroneEye2020 (A.4), CDNet (A.6), CascadeAdapt (A.7), HR-Cascade++ (A.9), Cascade R-CNN++ (A.21), DM-Net (A.16), CFPN (A.23), HRC (A.26), SSODD (A.28) and GabA-Cascade (A.29). Six methods are derived from anchor-free CenterNet [45, 9], *i.e.*, FPAFS-CenterNet (A.10), MSC-CenterNet (A.11), CenterNet+ (A.12), CN-FaDhSa (A.14), HRNet (A.15), and Center-ClusterNet (A.24). Three detectors combine ATSS [42] in their networks, namely TAUN (A.5), ASNet (A.13) and HR-ATSS (A.22). Besides, HRD-Net (A.17) is based on the High-Resolution Detection Network [25] which takes multiple resolution inputs using multi-depth backbones. PG-YOLO (A.18) is a variant of YOLOv3 [28] with a polymorphic module to learn the multi-scale and multi-shape object features and a group attention module to refine the combined features. EFPN (A.19) is based on the feature pyramid network to exploit the semantic information of small objects by multi-branched dilated bottleneck and attention and augmented bottom-up pathway [13]. CRENet (A.20) denotes the Cluster Region Estimation Network to search cluster regions containing dense objects, which makes the detector focus on these regions to reduce background interference. DOHR-RetinaNet (A.25) is modified from RetinaNet [21]. IterDet (A.27) proposes an alternative iterative scheme, where a new subset of objects is detected at each iteration.

3.3 Overall Evaluation

The overall results of the submissions are presented in Table 1 and Table 2. Compared with the winner detectors HAL-Retina-Net in the VisDrone-DET2018 Challenge [47] and DPNet-ensemble in the VisDrone-DET2019 Challenge [8], there are top 10 methods in the VisDrone-DET2020 Challenge achieving better mAP score more than 32.0. By using test-dev dataset in the training phase, the top performer DPNetV3 (A.1) in Table 1 performs slightly better than the top performer DroneEye2020 (A.4), *i.e.*, 37.37 vs. 34.57.

As discussed above, ensemble of several networks is effective to improve the accuracy of object detection. In Table 1, DPNetV3 (A.1) ensembles a few powerful backbones such as HRNet-W40 [33], Res2Net [12], Balanced Feature Pyramid Network [26], and Cascade R-CNN paradigm [3]. SMPNet (A.2) ranks the second place with the mAP score of 35.98, which also uses different combinations of multiple models (*i.e.*, Cascade-RCNN [3], HRNet [33], and ATSS [42]) to fuse the detection results.

Table 1. Object detection results in the VisDrone-DET2020 Challenge (model trained with the `test-dev` subset).

Method	AP[%]	AP50[%]	AP75[%]	AR1[%]	AR10[%]	AR100[%]	AR500[%]
DPNetV3 (A.1)	**37.37**	**62.05**	**39.10**	**0.85**	**7.96**	**42.03**	**53.78**
SMPNet (A.2)	**35.98**	**59.53**	**37.41**	0.29	2.01	8.46	**53.33**
DBNet (A.3)	**35.73**	**59.63**	**36.92**	0.37	2.78	12.70	**52.57**
ECascade R-CNN (A.8)	34.09	56.77	35.30	**1.06**	**7.73**	**35.31**	49.57
FPAFS-CenterNet (A.10)	32.34	56.46	32.39	**1.20**	**9.45**	**38.55**	51.61
DOHR-RetinaNet (A.25)	21.68	44.59	18.73	0.55	5.64	28.89	39.48
SSODD (A.28)	19.65	34.75	19.50	0.44	3.91	27.63	27.63

On the other hand, the use of the Cascade-RCNN [3] framework has become wide-spread recently (*e.g.*, from (A.4) to (A.9)) due to its high performance and easy extensibility. Compared with the baseline Cascade-RCNN [3] method with the mAP score of 16.09, the submitted varaints largely improve the performance by combining several effective modules. In Table 1, DBNet (A.3) improves Cascade R-CNN [3] by adding global context block [4], DCN [7] and double heads [39]. In Table 2, DroneEye2020 (A.4) is mainly based on Cascade R-CNN [3] with Recursive Feature Pyramid and Switchable Atrous Convolution [27], achieving comparable performance with 34.57 mAP. TAUN (A.5) uses mean teacher [36] to train the cascade DetectoRS model [3, 27], which performs similarly as DroneEye2020 (A.4). CDNet (A.6) and CascadeAdapt (A.7) combine Cascade-RCNN [3] with deformable convolutions, and then improve the detection accuracy using several data augmentation strategies such as sub-image splitting and mosaic [1].

3.4 Category based Evaluation

For comprehensive evaluation, we also report the detection results of each object category in Table 3 and Table 4. Compared with the results in the VisDrone-DET2019 Challenge [8], the top performers achieve much better accuracy in several categories, *e.g.*, *bus* and *awning-tricycle*. This is attributed to two reasons. First, ensemble learning takes full advantage of various backbones to deal with different poses and scales of objects especially in *bus*. Second, augmentation strategies can help training some categories (*e.g.*, *awning-tricycle*) lacking of training data.

In Table 3, It can be observed that the top 2 performers obtain the best results on almost all 10 categories. That is, using test-dev set in the training phase generally improves the performance. Notably, CascadeAdapt (A.7) achieve the best mAP score on *awning-tricycle*, which adopts several augmentation methods such as mosaic in YOLOv4 [1]. Besides, Cascade R-CNN variants take top three places in term of each category.

4 Conclusions

In this paper, we present the results of the VisDrone-DET2020 Challenge. It is the third annual object detector benchmarking activity, following the very

Table 2. Object detection results in the VisDrone-DET2020 Challenge (model trained without the `test-dev` subset). $*$ indicates that the detection algorithm is submitted by the committee.

Method	AP[%]	AP50[%]	AP75[%]	AR1[%]	AR10[%]	AR100[%]	AR500[%]
DroneEye2020 (A.4)	**34.57**	**58.21**	**35.74**	0.28	1.92	6.93	**52.37**
TAUN (A.5)	**34.54**	**59.42**	**34.97**	0.14	0.72	12.81	49.80
CDNet (A.6)	**34.19**	57.52	**35.13**	0.80	8.12	**39.39**	**52.62**
CascadeAdapt (A.7)	34.16	**58.42**	34.50	0.84	**8.17**	**39.96**	47.86
HR-Cascade++ (A.9)	32.47	55.06	33.34	0.94	7.81	**37.93**	50.65
MSC-CenterNet (A.11)	31.13	54.13	31.41	0.27	1.85	6.12	50.48
CenterNet+ (A.12)	30.94	52.82	31.13	0.27	1.84	5.67	**50.93**
ASNet (A.13)	29.57	52.25	29.37	0.25	1.69	6.46	46.01
CN-FaDhSa (A.14)	28.52	49.50	28.86	0.26	1.76	6.32	48.06
HRNet (A.15)	27.39	49.90	26.71	0.80	7.67	33.67	46.16
DMNet (A.16)	27.33	48.44	27.31	0.65	7.15	32.91	37.06
HRD-Net (A.17)	26.93	45.45	27.77	0.27	2.58	35.38	35.38
PG-YOLO (A.18)	26.05	49.63	24.15	**1.45**	**9.20**	33.65	42.63
EFPN (A.19)	25.27	48.18	23.37	**1.45**	**9.21**	32.91	40.65
CRENet (A.20)	25.16	44.38	24.57	0.27	2.44	21.21	36.44
Cascade R-CNN++ (A.21)	24.66	43.53	24.71	0.25	1.70	7.97	40.42
HR-ATSS (A.22)	24.23	41.84	24.43	0.27	2.15	34.97	34.97
CFPN (A.23)	22.85	42.33	21.88	0.81	7.08	29.65	39.55
Center-ClusterNet (A.24)	22.72	41.45	22.13	1.01	7.75	28.56	33.85
HRC (A.26)	21.23	43.56	18.39	0.18	1.16	4.88	37.25
IterDet (A.27)	20.42	36.73	20.25	0.21	1.34	8.86	33.04
GabA-Cascade (A.29)	18.85	33.60	18.66	**1.09**	7.68	26.25	33.03
CornerNet* [17]	17.41	34.12	15.78	0.39	3.32	24.37	26.11
Light-RCNN* [19]	16.53	32.78	15.13	0.35	3.16	23.09	25.07
FPN* [20]	16.51	32.20	14.91	0.33	3.03	20.72	24.93
Cascade R-CNN* [3]	16.09	31.91	15.01	0.28	2.79	21.37	28.43

Table 3. Object detection results of each object category (model trained with the `test-dev` subset).

Method	ped.	person	bicycle	car	van	truck	tricycle	awn.	bus	motor
DPNetV3 (A.1)	**38.03**	**22.10**	**18.68**	**57.14**	**44.67**	**40.33**	**38.21**	**28.06**	**54.10**	**32.39**
SMPNet (A.2)	**36.75**	**20.08**	**15.78**	**57.35**	**44.98**	**40.61**	**34.67**	**26.70**	**53.33**	**29.53**
DBNet (A.3)	**35.73**	**20.58**	**15.75**	**55.24**	**43.92**	**39.87**	**36.16**	**28.01**	**53.57**	28.43
ECascade R-CNN (A.8)	34.66	18.94	12.64	55.07	42.82	38.14	32.74	24.74	52.06	**29.11**
FPAFS-CenterNet (A.10)	31.55	13.77	14.68	55.04	42.30	37.55	29.48	24.23	48.24	26.57
DOHR-RetinaNet (A.25)	23.31	10.66	6.35	44.74	29.10	26.64	17.99	13.00	27.70	17.38
SSODD (A.28)	19.39	6.97	2.77	42.77	24.70	20.79	15.93	12.38	35.57	15.18

successful VisDrone-DET2018 and VisDrone-DET2019 challenges. Evaluated on the same dataset, many submitted object detection methods set a new state-of-the-art. Specifically, the top performer is DPNetV3 (A.1) using test-dev set as training data, with the overall mAP score of 37.37. Without test-dev set, the top three performers are DroneEye2020 (A.4), TAUN (A.5), and CDNet (A.6), with the overall mAP score of more than 35.00. The experimental results indicate that ensemble learning of a few powerful detectors can largely boost the detection performance. Besides, Cascade R-CNN and ATSS are other popular detection frameworks. It is worth mentioning that the best detector DPNetV3 (A.1) improves the mAP score by over 6% than before, which shows the development of object detection in the past year. However, the best mAP score is still less than 40% and far from satisfactory in real applications. Meanwhile, the computational complexity of the submitted algorithms is another issue on the

Table 4. Object detection results of each object category (model trained without the `test-dev` subset). $*$ indicates the detection algorithms submitted by the VisDrone Team.

Method	ped.	person	bicycle	car	van	truck	tricycle	awn.	bus	motor
DroneEye2020 (A.4)	**35.70**	**18.27**	**14.02**	**56.51**	**42.91**	37.61	**35.41**	**25.91**	**50.37**	**28.95**
TAUN (A.5)	**34.98**	**19.05**	**17.23**	54.62	41.71	**38.67**	**35.10**	**26.53**	48.49	**29.06**
CDNet (A.6)	**35.64**	**19.15**	**13.84**	**55.77**	42.12	**38.22**	32.97	25.42	**49.49**	**29.28**
CascadeAdapt (A.7)	31.61	15.63	12.83	53.59	**43.17**	**38.29**	**34.80**	**32.09**	**51.49**	28.13
HR-Cascade++ (A.9)	32.58	17.31	11.05	54.71	**42.37**	35.27	32.68	24.09	46.48	28.20
MSC-CenterNet (A.11)	33.70	15.23	12.07	55.19	40.47	34.08	29.24	21.63	42.23	27.45
CenterNet+ (A.12)	32.56	16.15	12.14	**55.35**	38.79	33.71	30.35	22.59	41.12	26.69
ASNet (A.13)	28.34	12.32	10.18	51.38	38.99	33.03	28.94	22.51	47.49	22.56
CN-FaDhSa (A.14)	30.52	12.89	9.85	52.52	38.14	32.91	25.85	22.00	39.89	20.61
HRNet (A.15)	27.61	13.35	11.79	50.78	36.46	29.89	24.42	21.03	35.16	23.36
DMNet (A.16)	29.06	13.30	9.59	50.27	30.54	32.04	27.15	17.60	39.68	24.03
HRD-Net (A.17)	26.83	12.13	8.05	48.97	32.73	31.38	26.16	18.39	42.10	22.57
PG-YOLO (A.18)	26.82	13.83	9.90	46.61	32.92	26.49	22.87	19.33	41.45	20.32
EFPN (A.19)	25.57	12.45	8.72	46.06	32.29	25.72	22.16	18.99	41.58	19.15
CRENet (A.20)	27.57	11.50	6.61	46.11	32.73	28.66	23.84	17.77	37.19	19.57
Cascade R-CNN++ (A.21)	24.81	8.83	4.79	48.96	35.45	29.01	24.36	19.95	33.46	17.02
HR-ATSS (A.22)	27.23	11.12	5.75	48.76	31.84	21.27	22.70	18.20	35.95	19.48
CFPN (A.23)	23.22	9.70	4.78	45.37	30.83	23.85	19.37	16.51	38.35	16.54
Center-ClusterNet (A.24)	24.02	9.14	5.70	48.43	26.43	27.14	19.36	14.90	33.30	18.76
HRC (A.26)	17.26	8.50	4.85	40.53	29.82	24.22	18.70	16.29	37.22	14.95
IterDet (A.27)	17.33	6.76	3.61	41.21	30.27	23.76	19.04	15.05	35.68	11.53
GabA-Cascade (A.29)	17.43	6.07	3.21	39.21	27.22	20.26	15.72	13.20	33.84	12.31
CornerNet* [17]	20.43	6.55	4.56	40.94	20.23	20.54	14.03	9.25	24.39	12.10
Light-RCNN* [19]	17.02	4.83	5.73	32.29	22.12	18.39	16.63	11.91	29.02	11.93
FPN* [20]	15.69	5.02	4.93	38.47	20.82	18.82	15.03	10.84	26.72	12.83
Cascade R-CNN* [3]	16.28	6.16	4.18	37.29	20.38	17.11	14.48	12.37	24.31	14.85

drone platform with limited resource. We hope we can provide a platform to advance state-of-the-art object detection methods on the drone-captured scenarios [46].

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61876127 and Grant 61732011, in part by Natural Science Foundation of Tianjin under Grant 17JCZDJC30800.

A Submitted Detectors

In this appendix, we provide a short summary of all algorithms that were considered in the VisDrone-DET2020 Challenge.

A.1 Drone Pyramid Network V3 (DPNetV3)

Heqian Qiu, Zichen Song, Minjian Zhang, Mingyu Liu, Taijin Zhao, Fanman Meng, Hongliang Li
hqqiu@std.uestc.edu.cn, szc.uestc@gmail.com, jamiezhang722@outlook.com, myl8562@163.com, zhtjww@std.uestc.edu.cn, fmmeng@uestc.edu.cn,

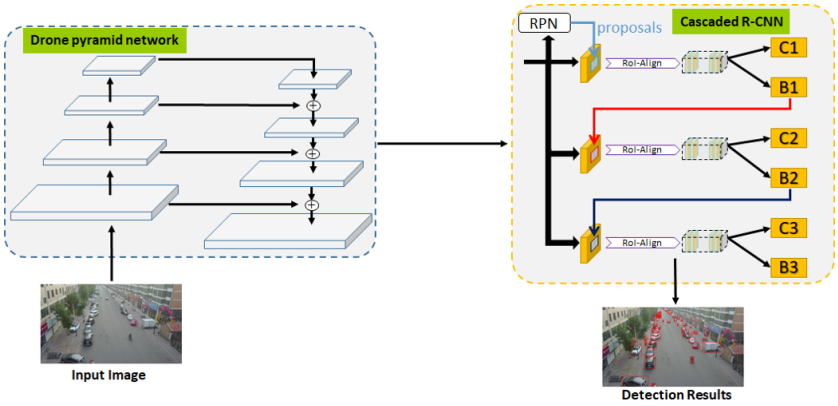

Fig. 2. The framework of DPNetv3.

hlli@uestc.edu.cn

DPNetV3 is an ensemble model for object detection, see Fig. 2. First, it adopts HRNet-W40 [33] pre-trained on ImageNet dataset as our backbone network, which starts from a high-resolution subnetwork as the first stage, gradually adds high-to-low resolution subnetworks one by one to form more stages, and connects the mutli-resolution subnetworks in parallel. In addition, we also use Res2Net [12] as our backbone networks. To make features more robust for complex scenes, we introduce Balanced Feature Pyramid Network [26] with CARAFE (Content-Aware ReAssembly of FEatures) [37] and Deformable Convolution [7] into these backbone networks. Furthermore, we use Cascade R-CNN paradigm [3] to progressively refine detection boxes for accurate object localization. We ensemble them using weighted box fusion method.

A.2 Using Split, Mosaic and Paster Modules for Detecting Aerial Images (SMPNet)

Chengzhen Duan, Zhiwei Wei
{18S151541, 19S051024}@stu.hit.edu.cn

In order to improve the accuracy of aerial image detection, we propose adaptive split method, mosaic data enhancement method and resampling enhancement method. The adaptive split method adjusts the split absolute area according to the average target size in the split, so that the detector can focus on the narrower target scale range that is conducive to detection. After that, we calculate the scaling factor required by the target in split, and then scale the split proportionally. Then we cut the four splits and splice them into mosaics [1]. In order to alleviate the problem of class imbalance, we use panoramic segmentation to build the target pool, and then paste the appropriate target from the target pool

to the training sample. Different from the previous method of pasting the whole GT box [5], we only paste the target. We use multi-model to infer and fuse the detection results, including Cascade-RCNN [3]+HRNet [33], ATSS [42]+HRNet [33], and ATSS [42]+Res2Net [12]+general focal loss [18].

A.3 DeepBlueNet (DBNet)

Zhipeng Luo, Sai Wang, Zhenyu Xu, Yuehan Yao, Bin Dong
{luozp, wangs, xuzy, yaoyh, dongb}@deepblueai.com

DBNet adopts Cascade_x101_64-4d [3] as the pipeline and adda global context block [4] to improve the ability of the extractor which could get more information from global feature. Meanwhile, we use DCN [7] to reduce the effect of feature misalignment, and adaptive part localization for objects with different shapes. As for R-CNN part, we use Double-Head RCNN [39]. Thus object classification is enhanced by adding classification task in conv-head, as it is complementary to the classification in fc-head. That is, bounding box regression provides auxiliary supervision for fc-head. We ensemble multi-scale testing results as our final result.

A.4 Cascade R-CNN on Drone-captured Scenarios (DroneEye2020)

Sungtae Moon, Joochan Lee, Jungyeop Yoo, Jonghwan Ko, Yongwoo Kim
stmoon@kari.re.kr, {maincold2, soso030, jhko}@skku.edu, yongwoo.kim@smu.ac.kr

DroneEye2020 is improved on Cascade R-CNN [3]. We divide original training images by 2×2 and horizontally flip all patches. If a divided patch has no objects, we exclude the patch when training. We use Cascade R-CNN [3] with ResNet-50 backbone, which is pretrained by COCO dataset. Notably, we use Recursive Feature Pyramid (RFP) for neck, and additionally use Switchable Atrous Convolution (SAC) for better performance [27].

A.5 Tricks are All yoU Need (TAUN)

Jingkai Zhou, Weida Qin, Zhongjie Fan, Shuqin Huang, Qiong Liu, Ming-Hsuan Yang
fs.jingkaizhou@gmail.com

TAUN is based on the cascade DetectoRS [3, 27]. The backbone is HRNet-40 [38], and the neck is the original HRFPN neck. ATSS [42] is used as the assigner in the RPN [29]. We use multi-scale image crop (not SAIC [44] for saving time and memory) to augment training and testing data, use mean teacher [36] to train the model. When model testing, we use ratio, outside, and scale filters to post filter outline bounding boxes. The threshold of those filters are counted based on the training dataset.

A.6 Cascade RCNN with DCN (CDNet)

Shuai Li, Yang Xiao, Zhiguo Cao
{*shuai_li1997, yang_xiao, zgcao*}*@hust.edu.cn*

CDNet is based on Cascade RCNN [3] with ResNeXt101. Moreover, we add deformable convolutional network [7] for better performance. To reduce GPU memory and consider small objects, we split the train images into sub-image with size 416×416 and train the network between 416 and 416×2 size. In testing phase, we use Soft-NMS and multi-scale testing to achieve better accuracy.

A.7 Cascade network with test time self-supervised adaptation (CascadeAdapt)

Weiping Yu, Chen Chen
2220170717@bit.edu.cn, chen.chen@uncc.edu

CascadeAdapt is improved from Cascade R-CNN [3] with ResNet backbone and Deformable Convolutions. We adopt several augmentation methods such as mosaic in YOLOv4 [1]. We use double heads instead of traditional box head to output the detection results. Although We use weighted box fusion to ensemble several models and obtain prediction on the test-challenge set. After that, we obtain pseudo labels of the test-challenge set by setting the threshold, and then finetune the model by the pseudo labels for 2 epoch to remove a large amount of false detections. The performance can be further improved by using other tricks such as GN, test time augmentation, label smooth and GIOU loss.

A.8 Enhanced Cascade R-CNN for Drone (ECascade R-CNN)

Wenxiang Lin, Yan Ding, Qizhang Lin
{*eutenacity, dingyan, 3120190071*}*@bit.edu.cn*

ECascade R-CNN is an ensembling method based on the work in [32]. We additionally train a detector with the backbone of HRNetv2 [34], ResNet50 [14] and ResNet101 [14]. For the convenience, we call the additional detectors as HRDet, Res50Det and Res101Det and call the first detector as RXDet. Finally, we train two detectors (*i.e.*, RXDet and HRDet) with the trick for class variance and four more detectors (RXDet, HRDet, Res50Det and Res101Det) without that. The final result is the output of the ensemble of all six detectors.

A.9 Cascade R-CNN with High-Resolution Network and enhanced feature pyramid (HR-Cascade++)

Hansheng Chen, Lu Xiong, Dong Yin, Yuyao Huang, Wei Tian
{*1552083, xiong_lu, tjyd, huangyuyao*}*@tongji.edu.cn, tian-w@hotmail.com*

HR-Cascade++ is based on the multi-stage detection architecture Cascade R-CNN [3], and is tuned specifically for dense small object detection. As the drone images include many small objects, we seek to obtain high resolution feature and improve the bounding box spatial accuracy. We adopt HRNet-W40 [38] as the backbone, which maintains high-resolution representations (1/4 of the original size) through the whole process. The feature pyramid network is enhanced with an additionally upsampled high resolution level (1/2 of the original image). This enables smaller and denser anchor generation, without resizing the original image. We adopt Quality Focal Loss [18] for R-CNN classification. Multi-scale and flip augmentations are applied in both training and testing. Photometric distortion is also used for training image augmentation. To reduce GPU memory consumption, the training images are cropped after resizing. If a ground truth bounding box is truncated during cropping, it is marked as ignore region when truncation ratio is greater than 50%. Soft-NMS is used for post-processing.

A.10 CenterNet with Feature Pyramid and Adaptive Feature Selection (FPAFS-CenterNet)

Zehui Gong
zehuigong@foxmail.com

FPAFS-CenterNet is based on CenterNet [45], because of its simplicity and high efficiency for object detection. CenterNet presents a new representation for detecting objects, in terms of their center locations. Other object properties, *e.g.*, object size, are regressed directly using the image features from the center locations. To achieve better performance, we have applied some useful modifications to CenterNet, with regard to the data augmentation, backbone network, feature fusion neck, and also, the detection loss. In order to extract more powerful features from input image, we employ CBNet [24] as our backbone network. we use BiFPN [35] as a feature fusion neck to enhance the information flow across the highly semantic and spatially finer features. We use GIOU loss [30], which is irrelevant to the object size.

A.11 CenterNet with multi-scale cropping (MSC-CenterNet)

Xuanxin Liu, Yu Sun
liuxuanxin@bjfu.edu.cn, sunyv@buaa.edu.cn

MSC-CenterNet employs CenterNet [45] with Hourglass-104 as the base network and does not use the pre-trained models. Considering the larger scale range for the intra-class and inter-class objects, the model is trained with multi-scale cropping. The input resolution is 1024×1024 and the input region is cropped from the image with the scale randomly choose from $(0.9, 1.1, 1.3, 1.5, 1.8)$.

A.12 CenterNet (CenterNet+)

Qiu Shi
qiushi_0425@foxmail.com

CenterNet+ is CenterNet [45] with the hourglass feature extractor where there are three hourglass blocks. In order to improve the performance of our detector for small samples, we change the stride from 7 to 2 in each hourglass block. Besides, we adopt multi-scale training and multi-scale test to improve the detector performance.

A.13 Aerial Surveillance Network (ASNet)

Michael Schleiss
michael.schleiss@fkie.fraunhofer.de

ASNet adopts the ATSS detector [42] based on the implementation from mmdetection v2.2. Standard settings are used if not stated otherwise. Our backbone is Res2Net [12] with 152 layers pretrained on ImageNet. In the neck we replace FPN with Carafe [37]. The neck has 384 output channels instead of 256 in the original implementation. Focal loss is replaced by generalized focal loss [30]. We use multi-scale training with sizes $[600, 800, 1000, 1200]$ for the shorter side. We train on a single gpu with batch size 4 for 12 epochs with a step wise learning schedule. Learning rate starts with 0.01 and is divided by 10 after epoch 8 and 11 respectively. We use no TTA and apply a scale of 1200 for the shorter side during testing.

A.14 CenterNet-Hourglass-104 (CN-FaDhSa)

Faizan Khan, Dheeraj Reddy Pailla, Sarvesh Mehta
{faizan.farooq, dheerajreddy.p}@students.iiit.ac.in, sarvesh.mehta@research.iiit.ac.in

CN-FaDhSa is modified from CenterNet [45]. Instead of using the default resolution of 512×512 during training, we train the model at the resolution of 1024×1024 and test at various scales of 2048×2048.

A.15 High-Resolution Net (HRNet)

Guosheng Zhang, Zehui Gong
249200734@qq.com
HRNet is similar to CenterNet [9]. However, we detect the object just as a single center point instead of triplets, and regress to object size $s = (h, w)$ for each object. In addition, we use the High-Resolution Net (HRNet) followed by FPN as the backbone, which is able to maintain high-resolution representations through the whole process.

A.16 Density map guided object detection (DMNet)

Changlin Li
cli33@uncc.edu

We use density crop+uniform crop to train baseline model and conduct fusion detection to obtain final detection. The baseline is Cascade R-CNN [3].

A.17 High-resolution Detection Network (HRD-Net)

Ziming Liu, Guangyu Gao
liuziming.email@gmail.com, guangyugao@bit.edu.cn

To keep the benefits of high-resolution images without bringing up new problems, we propose the High-Resolution Detection Network (HRDNet) [25]. HRDNet takes multiple resolution inputs using multi-depth backbones. To fully take advantage of multiple features, we propose Multi-Depth Image Pyramid Network (MD-IPN) and Multi-Scale Feature Pyramid Network (MS-FPN) in HRDNet. MD-IPN maintains multiple position information using multiple depth backbones. Specifically, high-resolution input will be fed into a shallow network to reserve more positional information and reducing the computational cost while low-resolution input will be fed into a deep network to extract more semantics. By extracting various features from high to low resolutions, the MD-IPN is able to improve the performance of small object detection as well as maintaining the performance of middle and large objects. MS-FPN is proposed to align and fuse multi-scale feature groups generated by MD-IPN to reduce the information imbalance between these multi-scale multi-level features.

A.18 A Slimmer Network with Polymorphic and Group Attention Modules for More Efficient Object Detection in Aerial Images (PG-YOLO)

Wei Guo, Jincai Cui
{*gwfemma, jinkaicui*}*@cqu.edu.cn*

PG-YOLO is a YOLOv3 [28] based slimmer network for more efficient object detection in aerial images. Firstly, a polymorphic module (PM) is designed for simultaneously learning the multi-scale and multi-shape object features, so as to better detect the hugely different objects in aerial images. Then, a group attention module (GAM) is designed for better utilizing the diversiform concatenation features in the network. By designing multiple detection headers with adaptive anchors and above-mentioned two modules, the final one-stage network called PG-YOLO is obtained for realizing the higher detection accuracy.

A.19 Extended Feature Pyramid Network with Adaptive Scale Training Strategy and Anchors for Object Detection in Aerial Images (EFPN)

Wei Guo, Jincai Cui
{gwfemma, jinkaicui}@cqu.edu.cn

EFPN comes from the work in [13]. To enhance the semantic information of small objects in deep layers of the network, the extended feature pyramid network is proposed. Specifically, we use the multi-branched dilated bottleneck module in the lateral connections and an attention pathway to improve the detection accuracy for small objects. for better locating the objects. Besides, an adaptive scale training strategy is developed to enable the network to deal with multi-scale object detection, where adaptive anchors are achieved by a novel clustering method.

A.20 Cluster Region Estimation Network (CRENet)

Yi Wang, Xi Zhao
{wangyi0102, xizhao_1}@stu.xidian.edu.cn

Aerial images are increasingly used for critical tasks, such as traffic monitoring, pedestrian tracking, and infrastructure inspection. However, aerial images have the following main challenges: 1) small objects with non-uniform distribution; 2) the large difference in object size. In this paper, we propose a new network architecture, Cluster Region Estimation Network (CRENet), to solve these challenges. CRENet uses a clustering algorithm to search cluster regions containing dense objects, which makes the detector focus on these regions to reduce background interference and improve detection efficiency. However, not every cluster region can bring precision gain, so each cluster region is calculated a difficulty score, mining the difficult cluster region and eliminating the simple cluster region to speed up the detection. Finally, a Gaussian scaling function is used to scale the difficult cluster region to reduce the difference of object size.

A.21 Cascade R-CNN for drone-captured scenes (Cascade R-CNN++)

Ting Sun, Xingjie Zhao
sunting9999@stu.xjtu.edu.cn, 1243273854@qq.com

We use Cascade R-CNN [3] as the baseline with four improvements: 1) We use Group normalization instead of Batch normalization; 2) We use online hard example mining to select positive and negative samples; 3) We use multi-scale testing; 4) We use two stronger backbones to train models and integrate them. Besides, we use ResNeXt as backbone to training, and Soft-Non maximum suppression instead of Non maximum suppression. At the same time, we use online

hard example mining to select positive and negative samples in Region proposal networks.

A.22 HRNet Based ATSS for Object Detection (HR-ATSS)

Jun Yu, Haonian Xie
harryjun@ustc.edu.cn, xie233@mail.ustc.edu.cn

HR-ATSS is based on Adaptive Training Sample Selection (ATSS) [42], which can automatically select positive and negative samples according to statistical characteristics of object. We employ HRNet [33] as backbone to improve small object performance, where HRFPN [33] is adopted as the feature pyramid. Specifically, we adopt HRNet-W32 as backbone, HRFPN as the feature pyramid, and ATSS detection head to regress and classify objects. We adopt Synchronized BN instead of BN.

A.23 Concat Feature Pyramid Networks (CFPN)

Yingjie Liu
1497510582@qq.com
CFPN is improved from FPN [20] and Cascade R-CNN [3], which uses concatenation for lateral connections rather than the addition in FPN. Meanwhile, in the fast R-CNN stage, cascade architecture named Cascade R-CNN is utilized to refine the bounding box regression. ResNet-152 is used as the pre-trained backbone. In the training stage, we apply a manual adjustment on the learning rate to optimize detection performance. In the testing stage, we use Soft-NMS for better recall on the dense objects.

A.24 CenterNet+HRNet (Center-ClusterNet)

Xiaogang Jia
18846827115@163.com

The Center-ClusterNet detector is based on CenterNet [45] and HRNet [33]. We use MobileNetV3 [15] as the backbone to predict all centers of the objects. Then K-Means is used as a post-processing method to generate clusters. Both original images and cropped images are processed by the detector. Then the predicted bounding boxes are merged by standard NMS.

A.25 Deep Optimized High Resolution RetinaNet (DOHR-RetinaNet)

Ayush Jain, Rohit Ramaprasad, Murari Mandal, Pratik Narang
{f20170093, f20180224}@pilani.bits-pilani.ac.in, 2015rcp9525@mnit.ac.in, pratik.narang@pilani.bits-pilani.ac.in

DOHR-RetinaNet is based on RetinaNet [21], using ResNet101 as the backbone. FPN is used for semantically strong feature extraction. Our backbone is pretrained on the ImageNet dataset. We use optimized anchors of 5 ratios and 5 scales using the optimization algorithm in [48]. All our input images are resized such that the minimum side is $1728px$ and the maximum side is $3072px$.

A.26 High Resolution Cascade R-CNN (HRC)

Daniel Stadler, Arne Schumann, Lars Wilko Sommer
daniel.stadler@kit.edu, {arne.schumann, lars.sommer}@iosb.fraunhofer.de

HRC is based on Cascade R-CNN [3] with FPN [20] and HRNetV2p-W32 [33] as backbone. We train four detectors with different anchor scales to account for varying object scales on randomly sampled image crops (608x608 pixels) of the VisDrone DET train and val set and use the SSD [23] data augmentation pipeline to enhance feature representation learning. For each class, the detector with best anchor scale is utilized. During test time, we follow a multi-scale strategy and, additionally, apply horizontal flipping. The resulting detections are combined via Soft-NMS [2]. To account for the large number of objects per frame, we increase the number of proposals and the maximum number of detections per image.

A.27 Iterative Scheme for Object Detection in Crowded Environments (IterDet)

Xin He
2962575697@whut.edu.cn

IterDet is an alternative iterative scheme, where a new subset of objects is detected at each iteration. Detected boxes from the previous iterations are passed to the network at the following iterations to ensure that the same object would not be detected twice. This iterative scheme can be applied to both one-stage and two-stage object detectors with just minor modifications of the training and inference procedures.

A.28 Small-scale Object Detection for Drone Data (SSODD)

Yan Luo, Chongyang Zhang, Hao Zhou
{luoyan_bb, sunny_zhang, zhouhao_0039}@sjtu.edu.cn

SSODD is based on Cascade R-CNN [3] using ResNeXt-101 64x4d as backbone and FPN [20] as feature extractor. We also use deformable convolution to enhance our feature extractor. Our pretrained model is based on COCO dataset. Other techniques like multi-scale training and soft-nms are involved in our method. To detect small-scale objects, we crop each image into four parts,

which are evenly distributed on the row image. The cropped four regions are fed into the framework with the multi-scale training technique, in which the scale is $(960, 720)$ and $(960, 640)$.

A.29 Cascade R-CNN enhanced by Gabor-based anchoring (GabA-Cascade)

Ioannis Athanasiadis, Athanasios Psaltis, Apostolos Axenopoulos, Petros Daras
{*athaioan, at.psaltis, axenop, daras*}*@iti.gr*

GabA-Cascade is build upon Cascade R-CNN [3] which is enhanced by considering an additional set of anchors targeted explicitly at small objects. Inspired by [31], a set of simplified Gabor wavelets (SGWs) is applied on the input image resulting in an edge-enhanced version of the latter. Thereafter, the maximally stable extremal regions (MSERs) algorithm is applied on the edge-enhanced image extracting regions possible of containing an object, called edge anchors. As a next step, we aim at integrating the edge anchors into the Region Proposal Network (RPN). Due to edge anchors being of varying scale and having continuous center coordinates, some modifications are required so as to be compatible with the RPN training procedure. In order for the Feature Pyramid Network (FPN) feature maps to remain scale specific and have the bounding box regressor referring to identical shaped anchors, the edge anchors are refined to match the closest available shape and size hyper parameters configuration. The issue of edge anchor centers not being aligned with the pixel grid is addressed through rounding their centers. Furthermore, additional feature maps, dedicated to the edge anchors, are introduced with the purpose of minimizing the previously mentioned refinement. These feature maps correspond to different scales relevant to small objects and are identical to the feature map of the first FPN pyramid level. After the modifications described above the RPN is able to evaluate regions given both edge and regular anchors as input. Finally the rest of the object detection pipeline follows the cascade architecture as described in [3], deploying classifiers of increasing quality.

References

1. Bochkovskiy, A., Wang, C., Liao, H.M.: Yolov4: Optimal speed and accuracy of object detection. CoRR **abs/2004.10934** (2020)
2. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms - improving object detection with one line of code. In: ICCV. pp. 5562–5570 (2017)
3. Cai, Z., Vasconcelos, N.: Cascade R-CNN: delving into high quality object detection. In: CVPR. pp. 6154–6162 (2018)
4. Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H.: Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In: ICCVW. pp. 1971–1980 (2019)
5. Chen, C., Zhang, Y., Lv, Q., Wei, S., Wang, X., Sun, X., Dong, J.: Rrnet: A hybrid detector for object detection in drone-captured images. In: ICCV. pp. 100–108 (2019)

6. Chen, H., Shrivastava, A.: Group ensemble: Learning an ensemble of convnets in a single convnet. CoRR **abs/2007.00649** (2020)
7. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: ICCV. pp. 764–773 (2017)
8. Du, D., Zhang, Y., Wang, Z., Wang, Z., Song, Z., Liu, Z., Bo, L., Shi, H., Zhu, R., *et al.*: Visdrone-det2019: The vision meets drone object detection in image challenge results. In: ICCVW. pp. 213–226 (2019)
9. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: Centernet: Keypoint triplets for object detection. In: ICCV. pp. 6568–6577 (2019)
10. Dwibedi, D., Misra, I., Hebert, M.: Cut, paste and learn: Surprisingly easy synthesis for instance detection. In: ICCV. pp. 1310–1319 (2017)
11. Gao, J., Wang, J., Dai, S., Li, L., Nevatia, R.: NOTE-RCNN: noise tolerant ensemble RCNN for semi-supervised object detection. In: ICCV. pp. 9507–9516 (2019)
12. Gao, S., Cheng, M., Zhao, K., Zhang, X., Yang, M., Torr, P.H.S.: Res2net: A new multi-scale backbone architecture. TPAMI (2019)
13. Guo, W., Li, W., Gong, W., Cui, J.: Extended feature pyramid network with adaptive scale training strategy and anchors for object detection in aerial images. Remote. Sens. **12**(5), 784 (2020)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
15. Howard, A., Pang, R., Adam, H., Le, Q.V., Sandler, M., Chen, B., Wang, W., Chen, L., Tan, M., Chu, G., Vasudevan, V., Zhu, Y.: Searching for mobilenetv3. In: ICCV. pp. 1314–1324 (2019)
16. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K.: Speed/accuracy trade-offs for modern convolutional object detectors. In: CVPR. pp. 3296–3297 (2017)
17. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: ECCV. pp. 765–781 (2018)
18. Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., Yang, J.: Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. CoRR **abs/2006.04388** (2020)
19. Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., Sun, J.: Light-head R-CNN: in defense of two-stage object detector. CoRR **abs/1711.07264** (2017)
20. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR. pp. 936–944 (2017)
21. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV. pp. 2999–3007 (2017)
22. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: ECCV. vol. 8693, pp. 740–755 (2014)
23. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., Berg, A.C.: SSD: single shot multibox detector. In: ECCV. vol. 9905, pp. 21–37 (2016)
24. Liu, Y., Wang, Y., Wang, S., Liang, T., Zhao, Q., Tang, Z., Ling, H.: Cbnet: A novel composite backbone network architecture for object detection. In: AAAI. pp. 11653–11660 (2020)
25. Liu, Z., Gao, G., Sun, L., Fang, Z.: Hrdnet: High-resolution detection network for small objects. CoRR **abs/2006.07607** (2020)
26. Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., Lin, D.: Libra R-CNN: towards balanced learning for object detection. In: CVPR. pp. 821–830 (2019)
27. Qiao, S., Chen, L., Yuille, A.L.: Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. CoRR **abs/2006.02334** (2020)

28. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. CoRR **abs/1804.02767** (2018)
29. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) NeurIPS. pp. 91–99 (2015)
30. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I.D., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: CVPR. pp. 658–666 (2019)
31. Shao, F., Wang, X., Meng, F., Zhu, J., Wang, D., Dai, J.: Improved faster R-CNN traffic sign detection based on a second region of interest and highly possible regions proposal network. Sensors **19**(10), 2288 (2019)
32. Solovyev, R., Wang, W.: Weighted boxes fusion: ensembling boxes for object detection models. CoRR **abs/1910.13302** (2019)
33. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: CVPR. pp. 5693–5703 (2019)
34. Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W., Wang, J.: High-resolution representations for labeling pixels and regions. CoRR **abs/1904.04514** (2019)
35. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: CVPR. pp. 10778–10787 (2020)
36. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: NeurIPS. pp. 1195–1204 (2017)
37. Wang, J., Chen, K., Xu, R., Liu, Z., Loy, C.C., Lin, D.: CARAFE: content-aware reassembly of features. In: ICCV. pp. 3007–3016 (2019)
38. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., Xiao, B.: Deep high-resolution representation learning for visual recognition. TPAMI (2020)
39. Wu, Y., Chen, Y., Yuan, L., Liu, Z., Wang, L., Li, H., Fu, Y.: Rethinking classification and localization in R-CNN. CoRR **abs/1904.06493** (2019)
40. Xu, J., Wang, W., Wang, H., Guo, J.: Multi-model ensemble with rich spatial information for object detection. PR **99** (2020)
41. Yun, S., Han, D., Chun, S., Oh, S.J., Yoo, Y., Choe, J.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: ICCV. pp. 6022–6031 (2019)
42. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: CVPR. pp. 9756–9765 (2020)
43. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: AAAI. pp. 13001–13008 (2020)
44. Zhou, J., Vong, C., Liu, Q., Wang, Z.: Scale adaptive image cropping for UAV object detection. Neurocomputing **366**, 305–313 (2019)
45. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. CoRR **abs/1904.07850** (2019)
46. Zhu, P., Wen, L., Du, D., Bian, X., Hu, Q., Ling, H.: Vision meets drones: Past, present and future. CoRR **abs/2001.06303** (2020)
47. Zhu, P., Wen, L., Du, D., Bian, X., Ling, H., Hu, Q., Nie, Q., Cheng, H., Liu, C., Liu, X., Ma, W., Wu, H., Wang, L., *et al.*: Visdrone-det2018: The vision meets drone object detection in image challenge results. In: ECCVW. vol. 11133, pp. 437–468 (2018)

48. Zlocha, M., Dou, Q., Glocker, B.: Improving retinanet for CT lesion detection with dense masks from weak RECIST labels. In: MICCAI. vol. 11769, pp. 402–410 (2019)
49. Zoph, B., Cubuk, E.D., Ghiasi, G., Lin, T., Shlens, J., Le, Q.V.: Learning data augmentation strategies for object detection. CoRR **abs/1906.11172** (2019)