VisDrone-VID2019: The Vision Meets Drone Object Detection in Video Challenge Results

Pengfei Zhu[1], Dawei Du[2], Longyin Wen[3], Xiao Bian[4], Haibin Ling[5], Qinghua Hu[1], Tao Peng[1], Jiayu Zheng[1], Xinyao Wang[3], Yue Zhang[3], Liefeng Bo[3], Hailin Shi[16], Rui Zhu[16], Bing Dong[9], Dheeraj Reddy Pailla[10], Feng Ni[9], Guangyu Gao[14], Guizhong Liu[17], Haitao Xiong[6], Jing Ge[14], Jingkai Zhou[6], Jinrong Hu[8], Lin Sun[11], Long Chen[8], Martin Lauer[7], Qiong Liu[6], Sai Saketh Chennamsetty[20], Ting Sun[17], Tong Wu[14], Varghese Alex Kollerathu[10], Wei Tian[7], Weida Qin[6], Xier Chen[15], Xingjie Zhao[17], Yanchao Lian[15], Yinan Wu[15], Ying Li[13], Yingping Li[15], Yiwen Wang[13], Yuduo Song[7], Yuehan Yao[9], Yunfeng Zhang[13], Zhaoliang Pi[15], Zhaotang Chen[8], Zhenyu Xu[9], Zhibin Xiao[12], Zhipeng Luo[9], Ziming Liu[14]

[1]Tianjin University, Tianjin, China.
[2]University at Albany, SUNY, Albany, NY, USA.
[3]JD Digits, Mountain View, CA, USA.
[4]GE Global Research, Niskayuna, NY, USA.
[5]Stony Brook University, New York, NY, USA.
[6]South China University of Technology, Guangzhou, China.
[7]Karlsruhe Institute of Technology, Karlsruhe, Germany.
[8]Sun Yat-sen University, Guangzhou, China.
[9]DeepBlue Technology (Shanghai) Co., Ltd, Beijing, China.
[10]Siemens Technology and Services Private Limited, Bengaluru, India.
[11]Samsung Inc., San Jose, CA, USA.
[12]Tsinghua University, Beijing, China.
[13]Northwestern Polytechnical University, Xi'an, China.
[14]Beijing Institute of Technology, Beijing, China.
[15]Xidian University, Xi'an, China.
[16]JD AI Research, Beijing, China.
[17]Xi'an Jiaotong University, Xi'an, China.

Abstract

Video object detection has drawn great attention recently. The Vision Meets Drone Object Detection in Video Challenge 2019 (VisDrone-VID2019) is held to advance the state-of-the-art in video object detection for videos captured by drones. Specifically, there are 13 *teams participating the challenge. We also report the results of* 6 *state-of-the-art detectors on the collected dataset. A short description is provided in the appendix for each participating detector. We present the analysis and discussion of the challenge results. Both the dataset and the challenge results are publicly available at the challenge website:* `http://www.aiskyeye.com/`.

1. Introduction

Although great progress has been achieved in detecting objects on static images, object detection in video has drawn increasing attention recently. However, it suffers from many challenging factors such as drastic appearance change, motion blur and occlusions when extending state-of-the-art object detectors from image.

To facilitate extending image object detectors for videos,

some benchmark datasets have been proposed, such as KITTI [11], ImageNet-VID [29] and UA-DETRAC [34, 23, 22]. Moreover, recent datasets [24, 28, 7] collected from drones, brings computer vision applications to drones more and more closely. Driven by these new datasets, the algorithms in video object detection are not usually optimal for dealing with video sequences generated by drones due to limited resources and new challenging factors (*e.g.*, camera change and motion blur) in drone platform. Therefore, a more general large-scale VisDrone-VDT2018 dataset [41] is proposed to further boost research on computer vision problems with drone platform.

In this paper, we present the "Vision Meets Drone Object Detection in Video" (VisDrone-VID2019) Challenge, organized in conjunction with the 17-th International Conference on Computer Vision (ICCV 2019) in Seoul, Korea. Based on the success of VisDrone-VDT2018 [41], the VisDrone-VID2019 Challenge continues to advance detection methods for various categories of objects from videos taken from drones. Specifically, 13 teams submit the detection results on the drone based dataset. We believe the submitted algorithms can facilitate boosting the research on video object detection with drones.

2. Related Work

In recent years, several video object detection algorithms are presented in the literature. We briefly discuss some prior work in video object detection field.

2.1. Feature Aggregation

One of the main-stream approaches [43, 36, 1, 33] is to enhance per-frame features through aggregating consecutive frames. Zhu *et al.* [43] perform video object detection using flow-guided feature aggregation to capture temporal coherence. Xiao and Lee [36] propose a new Spatial-Temporal Memory module to serve as the recurrent computation unit to model long-term temporal appearance and motion dynamics. In [1], deformable convolutions are used to learn spatially sample features from consecutive frames. MANet [33] can jointly calibrates the features of objects on both pixel-level and instance-level in a unified framework, while the pixel-level calibration is flexible in modeling detailed motion while the instance-level calibration captures more global motion cues.

2.2. Object Association

Another kind of typical solutions [12, 14, 9, 10] focuses on association between object proposals in sequential frames. Han *et al.* [12] use high-scoring object detections from nearby frames to boost scores of weaker detections within the same clip for better detection performance. Kang *et al.* [14] use a novel tubelet proposal network to generate spatio-temporal proposals, and a Long Short-term Memory (LSTM) network to incorporate temporal information from tubelet proposals. In [9], a new network is proposed to jointly perform detection and tracking of objects. Galteri *et al.* [10] connect detectors and object proposal generating functions to exploit the ordered and continuous nature of video sequences in a closed-loop.

3. The VisDrone-VID2019 Challenge

The goal of object detection in videos is to locate various categories of object instances in the videos instead of a static image. Specifically, 10 object categories of interest include *pedestrian*, *person*[1], *car*, *van*, *bus*, *truck*, *motor*, *bicycle*, *awning-tricycle*, and *tricycle*. The detector is required to run with fixed parameters on all experiments. Notably, the algorithms with detailed description (*e.g.*, speed, GPU and CPU information) will be published in the ICCV 2019 workshop proceeding with authorship.

3.1. The VisDrone-VID2019 Dataset

The VisDrone-VID2019 Dataset uses the same dataset in the VisDrone-VDT2018 Challenge, as shown in Figure 1. That is, it includes 79 sequences with $33,366$ frames in total, including three non-overlapping subsets, `training` set (56 video clips with $24,198$ frames), `validation` set (7 video clips with $2,846$ frames), and `testing` set (16 video clips with $6,322$ frames). These sequences are captured from different cities under various weather and lighting conditions. The annotations for the `training` and `validation` subsets are made available to users, but the annotations of the `testing` set are reserved to avoid (over)fitting of algorithms. The video sequences of the three subsets are captured at different locations, but share similar environments and attributes.

For quntitative evaluation, we use the AP, AP_{50}, AP_{75}, AR_1, AR_{10}, AR_{100} and AR_{500} metrics, similar to that in MS COCO [20] and the ILSVRC 2015 challenge [29]. Specifically, AP is the average score 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 object categories. AP_{50} and AP_{75} are the score at the single IoU thresholds 0.5 and 0.75 over all object categories, respectively. AR_1, AR_{10}, AR_{100}, and AR_{500} correspond to the maximum recalls with 1, 10, 100 and 500 detections per frame, averaged over all categories and IoU thresholds. Note that The submitted algorithms are ranked based on the AP score.

3.2. Submitted Detectors

We have received 13 detectors in the VisDrone-VID2019 Challenge, which are summarized in Table 1. Four detection methods employ multi-scale representation, including

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

Figure 1. Some annotated example frames of video object detection. The bounding boxes and the corresponding attributes of objects are shown for each sequence.

DM2Det (A.6), FRCFPN (A.8), HRDet (A.10) and Sniper+ (A.12). Four detectors are variants of Cascade R-CNN [2], *i.e.*, DBAI-Det (A.4), DetKITSY (A.5), Libra-HBR (A.11) and VCL-CRCNN (A.13). Three detectors are based on anchor-free Cornet-Net [15], AFSRNet (A.1), CN-DhVaSa (A.2) and CornerNet-lite-FS (A.3). Besides, EODST++ (A.7) and FT (A.9) consider temporal coherency for more robustness. We present a brief description of the submitted algorithms in Appendix A. We also conduct 6 state-of-the-art detectors, including 4 image object detectors (*i.e.*, FPN [18], CornerNet [15], CenterNet [39] and Faster R-CNN [27]) and 2 video object detectors (*i.e.*, FGFA [43] and D&T [9]). In summary, we evaluate 19 detectors in total in this challenge.

3.3. Results and Analysis

The results of the submitted algorithms are presented in Table 2. DBAI-Det (A.4) achieves the best AP score of all submissions, *i.e.*, 29.22. This is attributed to combination of many recently proposed powerful networks including DCNv2 [42], FPN [18] and Cascade R-CNN [2]. AFSRNet (A.1) ranks the second place, closely followed by HRDet+ (A.10). AFSRNet (A.1) takes full use of the advantages of both anchor based RetinaNet [19] and anchor-free FSAF [40], which improves the detection performance significantly. Besides, to reduce computational complexity, only the P2,P4,P6 feature maps of FPN [18] are used for classification and localization. HRDet+ (A.10) augments the high-resolution representation by aggregating the upsampled representations from all the parallel convolutions, leading to more discriminative representations. VCL-CRCNN (A.13) and CN-DhVaSa (A.2)) rank the 4-th and 5-th place respectively. The former is based on Cascade R-CNN [2] and the latter is based on CenterNet [39] with fine-tuned parameters. However, all the video object detection methods run in the speed of less than 10 fps because of high computational complexity.

Compared with the submissions in VisDrone-VDT2018 Challenge, the top 5 algorithms (*i.e.*, DBAI-Det (A.4), AFSRNet (A.1), HRDet+ (A.10), VCL-CRCNN (A.13) and

Table 1. The descriptions of the submitted video object detection algorithms in the VisDrone-VID2019 Challenge. GPUs and CPUs for training, implementation details, the running speed (in FPS), external training datasets and the references on the video object detection task are reported.

Method	GPU	CPU	Code	Speed	Datasets	Reference
AFSRNet (A.1)	GTX-1080Ti	Intel i7-5930K@3.50GHz	Python	4	×	RetinaNet [19]
CN-DhVaSa (A.2)	Tesla P100	Xeon Silver 4110	Python	4	COCO	CenterNet [39]
CornerNet-lite-FS (A.3)	RTX 2080TI		Python		×	CornerNet-Lite [16]
DBAI-Det (A.4)	Tesla V100	Intel Platinum 8160@2.10GHz	Python	1	COCO	Cascade R-CNN [2]
DetKITSY (A.5)	GTX Titan X	Intel i7-6700	C++	1.5	COCO	Cascade R-CNN [2]
DM2Det (A.6)	GTX Titan X		Python		×	M2Det [38]
EODST++ (A.7)	GTX Titan X	Xeon E5-2630v3	Python,C++	1	ImageNet	SSD [21]
FRFPN (A.8)	GTX 1080Ti		Python		×	Faster R-CNN [27]
FT (A.9)	GTX 1080Ti	E5-2620	Python	8	COCO	Faster R-CNN [27]
HRDet+ (A.10)	RTX 2080Ti	Intel E5-2650v4	Python	5	×	HRNet [31]
Libra-HBR (A.11)	GTX 1080Ti				×	SNIPER [30]
Sniper+ (A.12)	GTX 1080Ti	Intel E5-1620v4	Python	3.3	VOC+COCO	SNIPER [30]
VCL-CRCNN (A.13)	GTX 1080Ti	Intel E5-2640	Python	6.7	COCO	Cascade R-CNN [2]

Table 2. Video object detection results on the VisDrone-VID2019 `testing` set. $*$ indicates that the algorithm is submitted by the VisDrone Team. The best three performers are highlighted by the red, green and blue fonts.

Method	AP[%]	AP_{50}[%]	AP_{75}[%]	AR_1[%]	AR_{10}[%]	AR_{100}[%]	AR_{500}[%]
AFSRNet (A.1)	**24.77**	**52.52**	**19.38**	**12.33**	**33.14**	**45.14**	**45.69**
CN-DhVaSa (A.2)	21.58	48.09	16.76	**12.04**	29.60	39.63	40.42
CornerNet-lite-FS (A.3)	12.65	27.23	10.08	7.50	19.69	24.07	24.07
DBAI-Det (A.4)	**29.22**	**58.00**	**25.34**	**14.30**	**35.58**	**50.75**	**53.67**
DetKITSY (A.5)	20.43	46.33	14.82	8.64	25.80	33.40	33.40
DM2Det (A.6)	13.52	30.57	9.99	8.21	19.68	23.85	23.85
EODST++ (A.7)	18.73	44.38	12.68	9.67	22.84	27.62	27.62
FRFPN (A.8)	16.50	40.15	11.39	9.72	22.55	28.40	28.40
FT (A.9)	9.15	25.36	4.30	5.91	12.14	13.80	13.80
HRDet+ (A.10)	**23.03**	**51.79**	16.83	4.75	20.49	38.99	40.37
Libra-HBR (A.11)	18.29	44.92	11.64	10.69	26.68	35.83	36.57
Sniper+ (A.12)	18.16	38.56	14.79	9.98	27.18	38.21	39.08
VCL-CRCNN (A.13)	21.61	43.88	**18.32**	10.42	25.94	33.45	33.45
CFE-SSDv2 [37]	21.57	44.75	17.95	11.85	**30.46**	**41.89**	**44.82**
FGFA* [43]	18.33	39.71	14.39	10.09	26.25	34.49	34.89
D&T* [9]	17.04	35.37	14.11	10.47	25.76	31.86	32.03
FPN* [18]	16.72	39.12	11.80	5.56	20.48	28.42	28.42
CornerNet* [15]	16.49	35.79	12.89	9.47	24.07	30.68	30.68
CenterNet* [39]	15.75	34.53	12.10	8.90	22.80	29.20	29.20
Faster R-CNN* [27]	14.46	31.8	11.20	8.55	21.31	26.77	26.77

CN-DhVaSa (A.2)) in this year perform better than the winner of VisDrone-VDT2018 Challenge CFE-SSDv2 [37]. Apart from powerful backbone network, the top video detection methods benefit from multi-scale representation and multi-stage proposals. On the other hand, there are 4 more methods (*i.e.*, DetKITSY (A.5), EODST++ (A.7), Libra-HBR (A.11) and Sniper+ (A.12)) that outperforms or are on par with the best baseline method FGFA [43]. It is worth mentioning that EODST++ (A.7) uses the single object trackers to capture temporal information. Besides, two video detectors performs better than the remain three baseline image detectors. This is because the temporal coherency information helps reduce false positives and false negatives of detections.

3.4. Performance Analysis by Categories

Moreover, as shown in Figure 2, we report the performance of each detector in terms of 10 categories. The AP scores on 5 categories (*i.e.*, *pedestrian*, *car*, *van*, *truck* and *bus*) are obviously better than the rest 5 categories (*i.e.*, *person*, *bicycle*, *tricycle*, *awning-tricycle* and *motor*). We can conclude that the detectors perform not well when several objects appear at the same time, *e.g.*, the person riding a bicycle.

DBAI-Det (A.4) performs the best in every object category except *motor*. AFSRNet (A.1) performs the best in *motor* and ranks the second place in terms of *person*, *bicycle*, *car*, *tricycle*, *van* and *bicycle*. VCL-CRCNN (A.13)

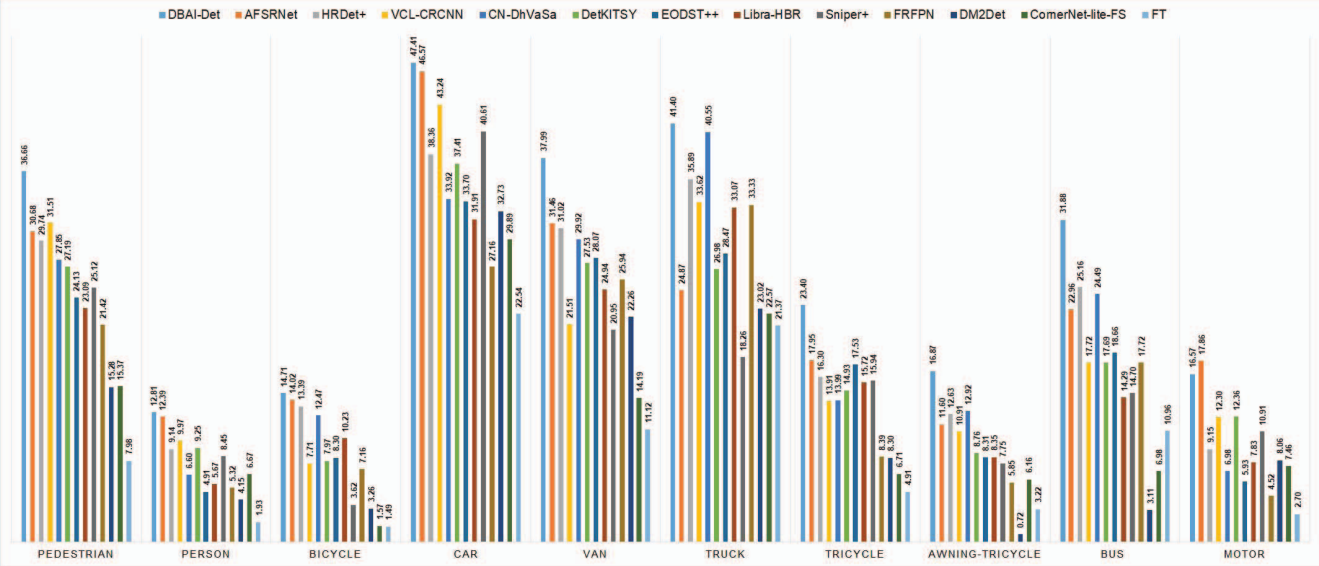

Figure 2. The average precision scores of the submitted detectors in each object category.

ranks the second place in terms of *pedestrian*. HRDet+ (A.10) ranks the second place in terms of *bus*. CN-DhVaSa (A.2) ranks the second place in terms of *truck* and *awning-tricycle*.

4. Conclusion

This paper concludes the VisDrone-VID2019 Challenge and its results. The dataset is the same as that in the VisDrone-VDT2018 Challenge with 10 object categories. The experiments of VisDrone-VID2019 show that the top performing detector in terms of the AP score is DBAI-Det (A.4), which merges several effective networks from recent published top conferences. Following DBAI-Det (A.4), AFSRNet (A.1) and HRDet+ (A.10) achieve similar promising performance, which demonstrates the effectiveness of multi-scale representation. However, the complexity of the current video detection methods is high, which is limited in real-time practical applications.

The focus of video object detection is on how we exploit temporal context across consecutive frames to improve object detection. One solution is to embed state-of-the-art single object trackers such as ECO [6] and SiamRPN++ [17] into image detectors. It can expand the detections with high confidence to recall the false negative objects efficiently (see EODST++ (A.7)). Another solution is to design end-to-end network based on several consecutive frames. For example, FT (A.9) takes three consecutive frames as the input of the network to extract time saliency features by three-dimensional convolution. Despite the performance improvement of video object detection on desktop GPUs, our future work will focus on advancing detectors at limited computational overhead.

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.

A. Submitted Detector

In the appendix, we summarize 13 video detection algorithms submitted in the VisDrone2019-VID Challenge, which are ordered alphabetically.

A.1. Augmented Feature Selected RetinaNet (AFSRNet)

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

AFSRNet is improved from RetinaNet [19], using the ResNeXt as backbone [13]. 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) [18]. 2) We add feature selected anchor-free head (FSAF) [40] 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 in 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.2. 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 [39]. 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.3. CornerNet in light version (CornerNet-lite-FS)

Hongwei Xv, Meng Zhang, Zihe Dong, Lijun Du and Xin Sun
coolbenn@foxmail.com, sunxin@ouc.edu.cn

CornerNet-lite-FS is improved from CornerNet-Lite [16]. Specifically, we model temporal appearance and enrich feature representation respectively in a video object detection framework. When we use CornerNet-Lite, we found that there lies a wrong big bounding box covering an area if the top-left and the bottom-right are the same class. We conquer this problem by removing the bbox if it covers the area more than 4 times of the average of the sum of top-left bbox and bottom-right bounding box. We add the features of adjacent frames to the current frame, which is also one of the common techniques in video object detection. We selected all boxes with confidence greater than 0.5 and average them to serve as their feature representations. Then all the boxes (classified as pedestrian) will calculate the cosine-similarity. If the similarity is less than 0.5, we filter it out.

A.4. DeepBlueAI-Detector (DBAI-Det)

Zhipeng Luo, Feng Ni, Yuehan Yao, Bing Dong and Zhenyu Xu
{luozp,nif,yaoyh,dongb,xuzy}@deepblueai.com

DBAI-Det is improved from Cascade-RCNN [2] with the ResNeXt101 backbone. We use FPN [18] based multi-scale feature maps to exploit robust representation of the object. Besides, DCNv2 [5] and GCNet [3] are used for better performance. The proposed model is implemented using the mmdetection toolbox[2].

A.5. KIT's detector for drone based scenes (DetKITSY)

Wei Tian, Jinrong Hu, Yuduo Song, Zhaotang Chen, Long Chen and Martin Lauer
{wei.tian, martin.lauer}@kit.edu, utppm@student.kit.edu, hujr3@mail2.sysu.edu.cn, 761042366@qq.com, chenl46@mail.sysu.edu.cn

DetKITSY is based on the Cascade R-CNN [2]. There are two stages in this approach. In the first stage, Region Proposal Network (RPN) predicts the anchors close to the objects and regresses the bounding box offsets, which is similar to Faster R-CNN [27]. RPN outputs a series of bounding boxes as proposals and feeds them into the second stage. The second stage consists of three serially connected bounding box heads, in which the output of previous head will be used as proposals for the next one. In the training, an anchor is considered as a positive example only when the its overlapping (by Intersection over Union (IoU)) with the ground-truth is bigger than a predefined threshold. For above three heads, we respectively set the thresholds as 0.5, 0.6, 0.7. Several modifications are applied to the original Cascade R-CNN to adapt to the VisDrone dataset. First, to fit the big variance of bounding box aspect ratio, we add more anchors with different aspect ratios in the RPN. Second, photo metric distortion and random cropping are used as data augmentation in training. Third, lower IoU threshold is used in non-maximum-suppression (NMS) in the post-processing. The reason is that, according to our observation, the objects with valid annotation seldom overlap, while the overlapping objects are usually in the "ignored" region. Last, multi-scale training and testing are used to improve the precision.

A.6. DroneEye based on M2Det (DM2Det)

SungTae Moon, Dongoo Lee, Yongwoo Kim and SungHyun Moon
{stmoon,ldg810,ywkim85}@kari.re.kr,
mosuhy@gmail.com

[2] https://github.com/open-mmlab/mmdetection

DM2Det is implemented based on M2Det network [38], which focuses on the detection of small object with general size in the VisDrone dataset. To check the small objects, the image is split into 4 pieces without image reduction. Then each split image is processed by M2Det and merged again using non-maximum-suppression. For the suitable detection of drone images, the way for image augmentation is fine-tuned.

A.7. Efficient Object Detector with the support of Multi-model Fusion and Spatial-Temporal information (EODST++)

Zhaoliang Pi, Yingping Li, Xier Chen, Yanchao Lian and Yinan Wu
{zlpi,ypli_3,xechen,yclian,wuyn}@stu.xidian.edu.cn

EODST++ consists of detection, tracking and false positive analysis modules. For detection, we train the models of SSD [21] and FCOS [32] to take advantage of their ability to detect targets with different scales. We combine the results of the two models at the decision level, based on the scores and categories of the targets they detected. For tracking, we use ECO [6] and SiamRPN++ [17] to conduct single object tracking from the objects with high score in chosen frame. The two trackers can recall the false negative objects efficiently. The objects maybe disappear across many continuous frames, so we will conduct the tracking process again based on the new recalled objects after the trajectory is confirmed to be correct, which could avoid the offset caused by long-term tracking and recall more lost objects. In terms of false positive analysis, we conduct box refinement and false positive suppression by inference according to temporal and contextual information of videos. First, based on the relationship of contextual regions, we use the features of different contextual regions to validate each other (bicycle and people, motor and people). Second, we evaluate the regional distribution of objects in each video and the range of object size of each categories, and remove the singular boxes with low scores.

A.8. Faster RCNN less FPN (FRFPN)

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

FRFPN is derived from still image detector Faster R-CNN [27] with ResNet101 backbone. Since the training set is limited in terms of environment variation, we train on the detector on patches cropped from original image with the size of 896×896. Two patch sampling strategies are used: one is to select patch center uniformly from grid that spreads over image evenly; while another one is to select x and y coordinates of the center with independent probability. Note that we crop patches on the fly during training. We also distort images in brightness, contrast, saturation and hue, and the SSD style [21] cropping strategy is used as well. In addition, we use the data augmentation policies learned by [44].

A.9. Faster R-CNN with temporal information (FT)

Yunfeng Zhang, Yiwen Wang and Ying Li
{zhangyunfeng,wangyiwen94}@mail.nwpu.edu.cn,
lybyp@nwpu.edu.cn

FT is based on Faster R-CNN [27] pre-trained on the MS COCO dataset [20]. We propose the network structure based on three-dimensional convolution to extract video timing information. Specifically, the network takes three consecutive frames as the input of the network structure to extract features, fuses the time saliency features obtained with the features extracted by Faster R-CNN and predicts the location of the object. In the model training, the two networks can be trained in parallel, which speeds up the convergence speed.

A.10. 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 [31]. It 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.11. Hybrid model based on Improved SNIPER, Libra R-CNN and Cascade R-CNN (Libra-HBR)

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

Libra-HBR is an ensemble of improved SNIPER [30], Libra R-CNN [25] 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 [5], attention mechanism [3], weight standardization [26] and group normalization [35]. 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.12. Improved SNIPER: Efficient Multi-Scale Training (Sniper+)

Xingjie Zhao, Ting Sun and Guizhong Liu
{zhaoxingjie,sunting9999}@stu.xjtu.edu.cn,
liugz@xjtu.edu.cn

Sniper+ is implemented by SNIPER [30], which can train on high resolution images for instance level visual recognition tasks. Our implementation based on Faster R-CNN[3] with a ResNet-101 backbone. We use VisDrone-VID2019, VOC0712 [8], and COCO2014 [20] datasets for training.

A.13. VCL's Cascade R-CNN (VCL-CRCNN)

Zhibin Xiao
xzb18@mails.tsinghua.edu.cn

VCL-CRCNN is based on the PyTorch implementation of Cascade R-CNN [2] by [4]. We use the model pre-trained on MS COCO dataset [20] and fine-tuned the model on VisDrone-VID train set. Besides, we choose ResNeXt-101 as the backbone network.

References

[1] G. Bertasius, L. Torresani, and J. Shi. Object detection in video with spatiotemporal sampling networks. In *ECCV*, pages 342–357, 2018.

[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.

[3] https://github.com/MahyarNajibi/SNIPER

[4] 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.

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

[6] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. ECO: efficient convolution operators for tracking. In *CVPR*, pages 6931–6939, 2017.

[7] 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.

[8] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015.

[9] C. Feichtenhofer, A. Pinz, and A. Zisserman. Detect to track and track to detect. In *ICCV*, pages 3057–3065, 2017.

[10] L. Galteri, L. Seidenari, M. Bertini, and A. D. Bimbo. Spatio-temporal closed-loop object detection. *TIP*, 26(3):1253–1263, 2017.

[11] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, pages 3354–3361, 2012.

[12] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang. Seq-nms for video object detection. *CoRR*, abs/1602.08465, 2016.

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

[14] K. Kang, H. Li, T. Xiao, W. Ouyang, J. Yan, X. Liu, and X. Wang. Object detection in videos with tubelet proposal networks. In *CVPR*, pages 889–897, 2017.

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

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

[17] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. *CoRR*, abs/1812.11703, 2018.

[18] 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.

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

[20] 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.

[21] 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.

[22] S. Lyu, M. Chang, D. Du, W. Li, Y. Wei, M. D. Coco, P. Carcagnì, and et al. UA-DETRAC 2018: Report of AVSS2018 & IWT4S challenge on advanced traffic monitoring. In *AVSS*, pages 1–6, 2018.

[23] S. Lyu, M. Chang, D. Du, L. Wen, H. Qi, Y. Li, Y. Wei, L. Ke, T. Hu, M. D. Coco, P. Carcagnì, and et al. UA-DETRAC 2017: Report of AVSS2017 & IWT4S challenge on advanced traffic monitoring. In *AVSS*, pages 1–7, 2017.

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

[25] 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.

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

[27] 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.

[28] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *ECCV*, pages 549–565, 2016.

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

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

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

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

[33] S. Wang, Y. Zhou, J. Yan, and Z. Deng. Fully motion-aware network for video object detection. In *ECCV*, pages 557–573, 2018.

[34] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu. DETRAC: A new benchmark and protocol for multi-object tracking. *CoRR*, abs/1511.04136, 2015.

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

[36] F. Xiao and Y. J. Lee. Video object detection with an aligned spatial-temporal memory. In *ECCV*, pages 494–510, 2018.

[37] Q. Zhao, T. Sheng, Y. Wang, F. Ni, and L. Cai. Cfenet: An accurate and efficient single-shot object detector for autonomous driving. *CoRR*, abs/1806.09790, 2018.

[38] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling. M2det: A single-shot object detector based on multi-level feature pyramid network. In *AAAI*, pages 9259–9266, 2019.

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

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

[41] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, H. Wu, Q. Nie, H. Cheng, C. Liu, X. Liu, W. Ma, L. Wang, and et al. Visdrone-vdt2018: The vision meets drone video detection and tracking challenge results. In *ECCVW*, pages 496–518, 2018.

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

[43] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei. Flow-guided feature aggregation for video object detection. In *ICCV*, pages 408–417, 2017.

[44] B. Zoph, E. D. Cubuk, G. Ghiasi, T. Lin, J. Shlens, and Q. V. Le. Learning data augmentation strategies for object detection. *CoRR*, abs/1906.11172, 2019.