This ICCV Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

RRNet: A Hybrid Detector for Object Detection in Drone-captured Images

Changrui Chen, Yu Zhang, Qingxuan Lv, Shuo Wei, Xiaorui Wang, Xin Sun*, Junyu Dong
Ocean University of China
{ccr, zy3s, lqx, weishuo, recyclerblacat}@stu.ouc.edu.cn, {sunxin, dongjunyu}@ouc.edu.cn

Abstract

Objects captured by UAVs and drones in city scenes usually come in various sizes and are extremely dense. Therefore, we propose a hybrid detector, called RRNet, for object detection in such challenging tasks. We mix up the anchor-free detectors with a re-regression module to construct the detector. The discard of prior anchors released our model from the difficult task on bounding-box size regression so that we achieved a better performance in multi-scale object detection in the dense scene. The anchor-free based detector firstly generates the coarse boxes. A re-regression module is then applied on the coarse predictions to produce accurate bounding boxes. In addition, we introduce an adaptive resampling augmentation strategy to logically augment the data. Our experiments demonstrate that RRNet significantly outperforms all the state-of-the-art detectors on VisDrone2018 dataset. We are runner-up to the ICCV VisDrone2019 Object Detection in Images Challenge [23], and we achieve the best AP50, AR10, and AR100. Source code will be published on our official website in due course.

1. Introduction

UAVs (Unmanned Aerial Vehicle) and drones have been widely adopted in both academia and real-world applications [18, 24, 25]. It therefore requires us to understand and analyse the image data captured by them. In the deep learning era, DNNs (deep neural networks) based object detectors [17, 16, 5, 9, 21] significantly boost the performance of object detection. However, there exists many significant differences between normal nature images and drone captured images; these differences make the object detection a challenging task. Firstly, the objects in such images come in various scales. As shown in Figure 1a, the far objects are extremely small, and the close objects are large. Moreover, there are numerous dense scenes in cities (*e.g.*, Figure 1b). The denseness causes a large amount of occlusion, making the object detection even more difficult .

(a) (b)

Figure 1. (a) Because of the special shooting angle, the object of the same category can come in extremely various size. (b) Dense scene example.

In general, the current deep learning based object detectors are divided into two categories. The first one is two-stage detectors [2, 17, 5]. They use a region proposal network to determine whether the prior anchors is an object or background. The prior anchors are several manually defined potential bounding boxes. Then, they use two head networks to classify the potential anchors into a set of categories and estimate the offset between the anchors and ground truth boxes. The other category is called one-stage detectors [14, 16]. Differing from the two-stage detectors, the one-stage detectors discard the region proposal network. They directly use two detectors to predict the categories and the offset of the prior anchors. The prior anchors of these two types of detectors are generated on the low-resolution image grid. Each prior anchor can be only assigned one object bounding box according to the IoU (intersection-over-union). However, with the drones captured image, the fixed-shape anchor can hardly handle the object of various scales. Recently, another type of detectors are proposed, *i.e.*, anchor-free detector. They reduce the bounding box prediction to the key point and size estimation. It poses a better way to detecte the objects with various scales. Nevertheless, the large difference in size (*e.g.*, from 10^1 to 10^3) makes regression difficult.

In this paper, we propose a hybrid detector called RRNet. Regardless of the various scales of objects, the cen-

*Prof.Sun is the corresponding author.

ter point of the objects always exists. Consequently, , we use two detectors to predict the center point and the width and height of each object instead of using the anchor box. Then, we transform these center points and sizes to coarse bounding boxes. Finally, we feed the deep feature maps and the coarse bounding boxes into a Re-Regression module. The Re-Regression module can adjust the coarse bounding boxes and generate the final accurate bounding boxes. Moreover, pieces of evidence [26] have shown that good data augmentation can even boost deep models to achieve state-of-the-art performance without changing the network architecture. Consequently, we propose a data augmentation strategy called adaptive resampling (AdaResampling). This strategy can logically augment objects on the image.

Our experiment demonstrates that the proposed model model significantly outperforms the existing state-of-the-art detectors on the VisDrone2018 dataset [22]. In principle, our RRNet is a hybrid model of the anchor-free detector and the two-stage detector. We believe the re-regression module is critical for the good results. Our model is the **runner-up** to the *ICCV VisDrone2019 Object Detection in Images Challenge* [23]. Moreover, we achieve the best AP50, AR10, and AR100 results.

In summary, the main contributions of this paper are:

- We propose a novel hybrid object detector consists of a coarse detector and a re-regression module for detection in drones captured images.
- We propose an adaptive augmentation strategy called AdaResampling to logically augment the object.
- Our detector achieves the best results of AP50, AR10, and AR100 in the *ICCV VisDrone2019 Object Detection in Images Challenge* [23].

2. Related work

Data augmentation In order to eliminate the bias between the training dataset and the testing dataset. Deep models usually use many data augmentations, such as random cropping and random flipping, to avoid over-fitting. Zoph *et al*. [26] use automated machine learning (AutoML) to search the best augmentation strategies. They achieve state-of-the-art without changing any network architecture. Kisantal *et al*. [7] use the copy-pasting to boost the performance of small objects. They firstly use segmentation masks to crop small objects, and then randomly paste the cropped small objects in the image. However, we can not simply paste the cropped object randomly in the drone captured image. We noticed that there is an obvious position prior in the drones captured image. For example, car flies in the sky is impossible. So, we propose a novel adaptive data augmentation method called AdaResample.

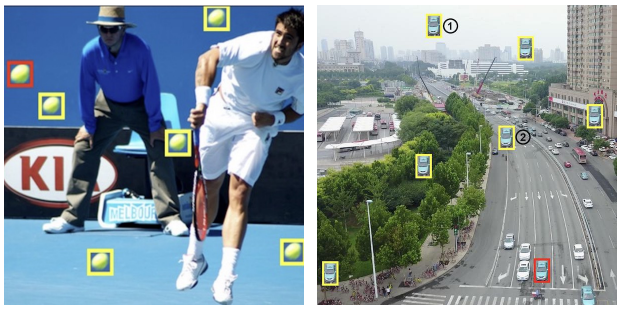
Anchor-based object detection. The anchor is widely adopted by most of the existing detectors. The two-stage detectors have long been the dominant method in the field of object detection. Faster RCNN [17] proposed the Region Proposal Network (RPN) to generate proposals. Then, the proposals are sent to the second stage to generate the final bounding boxes. Most of the other two-stage methods [11, 5] are a variant version of Faster RCNN. Besides, some multi-stage detectors have been proposed. Cascade RCNN [2] extends the Faster RCNN [17] to address the problems of over-fitting and quality mismatch. Compared to two-stage and multi-stage approaches, the single-stage methods have no proposal generation stage and predicts bounding boxes in one section. Although they do not generate proposals, single-stage methods still use anchor boxes. SSD [14] and YOLO [16] directly classify and regress the anchors to get the final bounding boxes. RetinaNet [12] introduces focal loss to address the class imbalance problem by reshaping the standard cross-entropy loss.

Anchor-free object detection. Recently, some detectors discard the prior anchors. They transform the object detection task to key point and size estimation. CornerNet [9] detects bounding box corners as key points and then matchs the upper left and lower right in the post-process, while ExtremeNet [21] detects the top-, left-, bottom-, right-most, and center points of all objects. CenterNet [21] simply extracts a single center point per object and predict the width and height of it. FoveaBox [8] predicts category-sensitive semantic maps for the object existing possibility, and produces a category-agnostic bounding box for each position that potentially contains an object.

3. AdaResampling

In this section, we introduce an adaptive augmentation called AdaResampling. Inspired by Kisantal *et al*. [7], the main idea of the proposed augmentation is to resample the confusing objects and paste them on the image many times.

Figure 2a is an image sampled from the COCO dataset [13]. Randomly pasting the cropped object in this type of image will not break the logicality of the image. However, as shown in Figure 2b, the simple copy-pasting augmentation may generate a very ridiculous image. We noticed that there are two mismatches. The first one is **background mismatch**. For example, the car marked by ① is flying in the sky. The background mismatch may lead the model to generate more false-positive bounding boxes. The reason is that The classifier relies on not only the object feature but also the context features. The classifier can learn the background prior knowledge to assist itself in classification. The second one is **scale mismatch**. If we copy a large object to a far background, the object (*e.g*., ② in Figure 2b) will be extremely bigger than the neighbor objects. In general, the



(a) (b)

Figure 2. The object surrounded by the red box is the original object. The objects surrounded by the yellow boxes is the resampled objects (a) Object resampling example on COCO dataset. (b) Object resampling example on VisDrone2018 dataset.

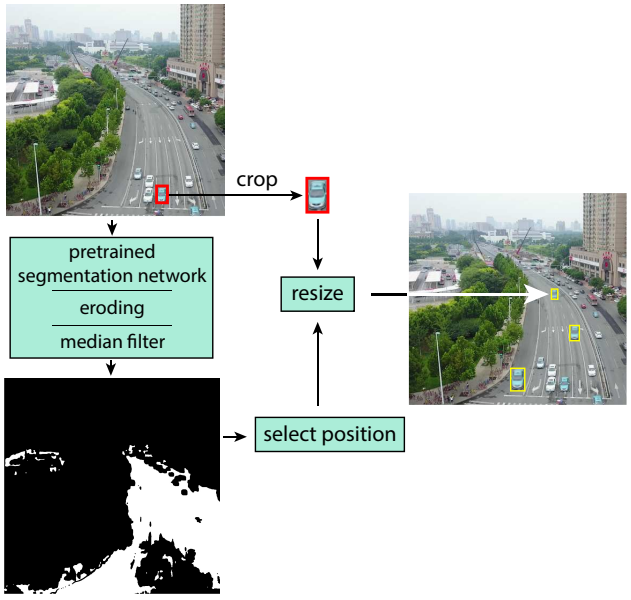
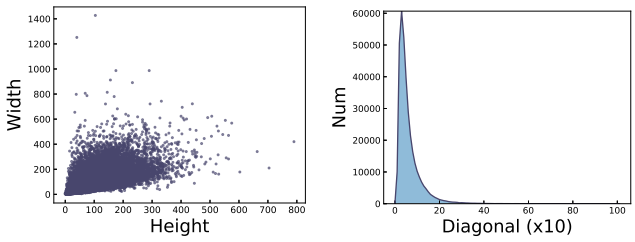


Figure 3. The pipeline of AdaResampling.

neighbor objects can provide useful knowledge to the size regression of the current object. The scale mismatch will mess up this knowledge.

In order to eliminate these two mismatches, we propose an adaptive augmentation strategy called AdaResampling. Figure 3 presents the pipeline of the AdaResampling. At the beginning, we feed the drone captured image into a pretrained semantic segmentation network to get the prior road map. Because of the discrepancy between the drones captured image and the dataset used for segmentation network training, the segmentation network might produce a noisy result. We do not require a high recall value, but a high precision of the road. Therefore, we use the eroding algorithm and a 3×3 median filter to remove the fake road area as possible as we can. Then, we sample a valid position according to the road map to place the augmented object. After that,



(a) (b)

Figure 4. The statistic data of all the bounding box in VisDrone2018 *train* set. (a) The height and width. (b) The number of different diagonal size.

the cropped object is resized by a transform function. The ratio of height to width is constant. The scaled height can be calculated by a simple linear function:

$$\begin{aligned} a &= \frac{1}{k}\sum_{i=0}^{k}\frac{h_i^{(l)} - h_i^{(s)}}{y_i^{(l)} - y_i^{(s)}} \\ b &= \frac{1}{k}\sum_{i=0}^{k} h_i^{(s)} - a y_i^{(s)} \\ h_{scaled} &= ay + b, \end{aligned} \tag{1}$$

where $h^{(l)}$, $h^{(s)}$, $y^{(l)}$, and $y^{(s)}$ are the height and the y coordinate of the largest and smallest object. We only use the largest and the smallest k *pedestrian* to calculate a. y is the y coordinate of the selected valid position. Finally, the scaled object can be placed in the selected position. We define a dense coefficient d to control the number of resampled object. The number of the resampled objects n can be calculated by:

$$n = max(d \times N_r, 5), \tag{2}$$

where N_r is the numbers of the prior road pixels.

The right part of Figure 3 is the training image augmented by our AdaResampling. We can see that the car can only be placed on the road and the scale of the augmented object is suitable.

4. Re-Regression Net

We collect some statistic data of the VisDrone2018 datasets. The results are reported in Figure 4. Figure 4a is the height and width of all the bounding boxes. The object size varies from 10^1 to 10^3. It is hard to define a proper set of prior anchors to cover this large gap. Besides, Figure 4b is the diagonal length of all the bounding boxes. Most of the objects are smaller than 50×50 pixels. We believe that the key point based detectors are more suitable for small object detection. Consequently, we propose the RRNet.

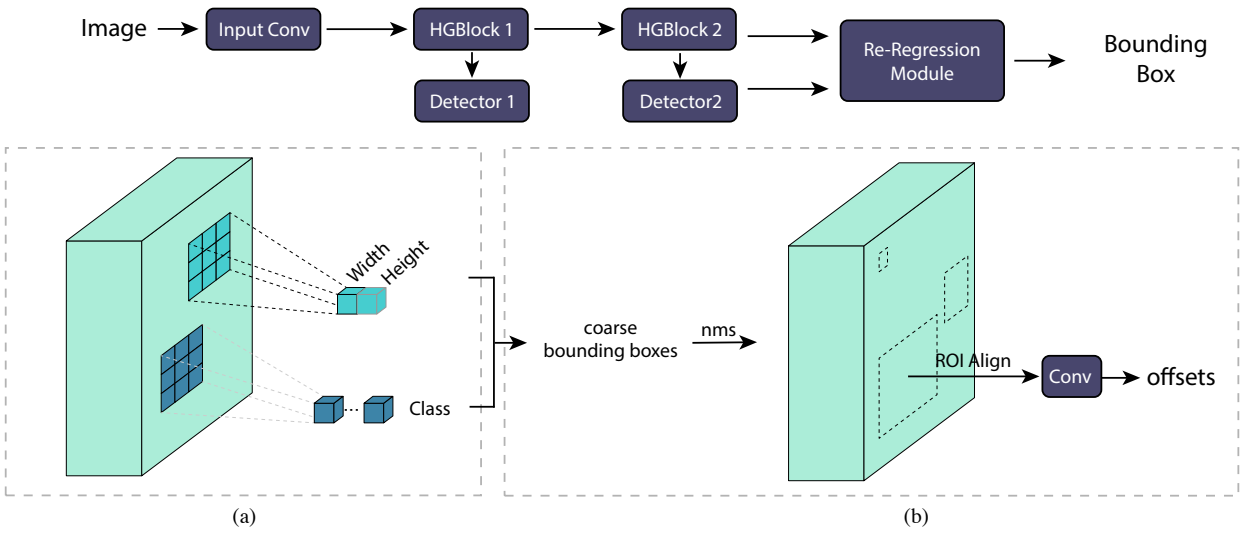


Figure 5. The architecture of the RRNet. (a) The detector in the RRNet. We only present the prediction for one feature pixel. We perform the same prediction on all the feature pixel. (b) The Re-Regression Module.

The top part of Figure 5 is the architecture of the RRNet. We firstly feed the image into some convolutional blocks to get the initial feature maps. After that, two HourGlass blocks (HGBlock) [15] extracts robust feature maps with more semantic information. We feed these features into two independent detectors. The heatmap detector produces a category-sensitive probability heatmap for the object center points. Besides, another detector will give the size estimation for all the center points.

4.1. Coarse detector

As shown in Figure 5, the coarse detector consists of one size estimation block and one category prediction block. The size estimation part is used to directly predict the height and width of each object. The category prediction network operates very similar to a semantic segmentation network. We predict the category-sensitive center point for each pixel and finally apply a sigmoid active function to get the independent probability for each category.

4.2. Re-Regression

We transform the heatmap and the size prediction to the coarse bounding boxes. Finally, the re-regress module is applied to these coarse detection boxes to generate the refined bounding boxes.

The Re-Regression module allows our model to refine the coarse bounding boxes. We feed the feature maps produced by HGBlock 2 and the coarse bounding boxes into the Re-Regression module. The Re-Regression module is similar to the Faster-RCNN head but excludes the classification network. We firstly use the NMS algorithm to filter the duplicate bounding boxes. After that, we use the ROI-align to align the features and use two convolutional layers to predict the offset value. Finally, we apply the offset value to the coarse bounding box to get the final prediction.

5. Experiments

We use the VisDrone2018 dataset [22] to evaluate our model. We report the mAP, AP50, AP75, and AR1$\sim$500.

5.1. Data augmentation

Similar to most of the deep neural networks, we also apply the horizontal flipping and random cropping as our basic augmentation. The crop size for the training phase is 512×512. We use the proposed AdaResampling to augment the people, pedestrian, bicycle, tricycle, awning-tricycle, and motor. The dense coefficient d is set to 0.00005. The pretrained segmentation network in our AdaResampling is a Deeplabv3 [3] pretrained on Cityscapes dataset [4].

5.2. Network details

Table 2 presents the detail settings of our RRNet. The input convolution and the HGBlocks are following the official setting of the HourGlass network [15]. Before re-regressing the coarse bounding boxes, we first select the top 1500 bounding boxes according to their classification confidence. Then, we use the Non-Maximum Suppression (NMS) with

Methods	mAP	AP50	AP75	AR1	AR10	AR100	AR500
RetinaNet [12]	11.81	21.37	11.62	0.21	1.21	5.31	19.29
RefineDet [20]	14.90	28.76	14.08	0.24	2.41	18.13	25.69
DetNet [19]	15.26	29.23	14.34	0.26	2.57	20.87	22.28
Cascade RCNN [2]	16.09	16.09	15.01	0.28	2.79	21.37	28.43
CornerNet [9]	17.41	34.12	15.78	0.39	3.32	24.37	26.11
FPN [11]	16.51	32.20	14.91	0.33	3.03	20.72	24.93
Light-RCNN [10]	16.53	32.78	15.13	0.35	3.16	23.09	25.07
ACM-OD†	29.13	54.07	27.38	0.32	1.48	9.46	44.53
DPNet-ensemble†	**29.62**	54.00	**28.70**	0.58	3.69	17.10	42.37
RRNet	29.13	**55.82**	27.23	**1.02**	**8.50**	**35.19**	**46.05**

Category	ped	people	bicycle	car	van	truck	tricycle	awn	bus	motor
mAP	30.442	14.851	13.724	51.427	36.143	35.224	28.019	18.999	44.204	25.854

Table 1. The performances on VisDrone2018 *test* subset. † is the champion and the third place in the *ICCV VisDrone2019 Object Detection in Images Challenge* (reported in the pre-released version of the leaderboard). They may change their method's names and performances in the final version of their papers.

Module Name	Details
Input Conv	[15]
HGBlock1	[15]
HGBlock2	[15]
Detector	size: conv(3×3) class: conv(3×3), ReLU, conv(1×1) , Sigmoid
RR Module	nms: IoU threshold=0.7 topk: 1500 ROI Align: 3×3 conv: Bottleneck [6], conv(1×1)

Table 2. The detail settings of the RRNet. The Input Conv and HGBlocks are following setting of HourGlass Net [15].

0.7 IoU threshold to filter the duplicated bounding boxes. The ROI Align size is set to 3.

5.3. Training details

In our experiments, we adopt Adam as our optimizer. Each mini-batch has 4 images per GPU, we train our model on 4 GPUs for 100k iterations, with a learning rate of 2.5e-4 which is decreased by 10 at the 60k and 80k iteration. The loss function for classification is the focal loss. The smooth L1 is used for regression. The overall training objective is:

$$L_{overall} = L_{cls}^{(d)} + \alpha L_{size}^{(d)} + L_{off}^{(d)} + L_{size}^{(r)}, \qquad (3)$$

where $L^{(d)}$ is the loss function for the coarse detectors, $L_{(r)}$ is for the Re-Regression module. $L_{(d)}$ and α is following the setting of CenterNet [21]. Similar to Faster RCNN [17], $L_{size}^{(d)}$ operates on the offset vector:

$$\begin{aligned} \delta_x &= (g_x - b_x)/b_w, \qquad \delta_y = (g_y - b_y)/b_h, \\ \delta_w &= log(g_w/b_w), \qquad \delta_h = log(g_h/b_h) \end{aligned} \qquad (4)$$

5.4. Inference details

At inference time, we discard the first detectors and perform the coarse box prediction only on the second detectors. The re-regression module is then applied to the highest scoring 1500 coarse detection boxes followed by the soft non-maximum suppression [1].

5.5. Performance

We show the comparison results of RRNet to the state of the art detectors in Table 1. RRNet outperforms all the state-of-the-art baseline models. We also cite the performance of DPNet-ensemble and ACM-OD, which is the first and the third place of the challenge. Our RRNet gets the highest AP50 and the best AR. Notably, all the AR of our RRNet are significantly higher than others. These results suggest one conclusion. Our network can detect more hard examples. Figure 6 is the example visualization of our model.

There are also some interesting results in Table 1. The point-based detectors (*e.g.*, CornerNet [9], RRNet) performances better than all the anchor-based detectors.

6. Ablation study

In this section, we perform a thorough ablation study on the VisDrone2018 *val* subset to analyze our RRNet.

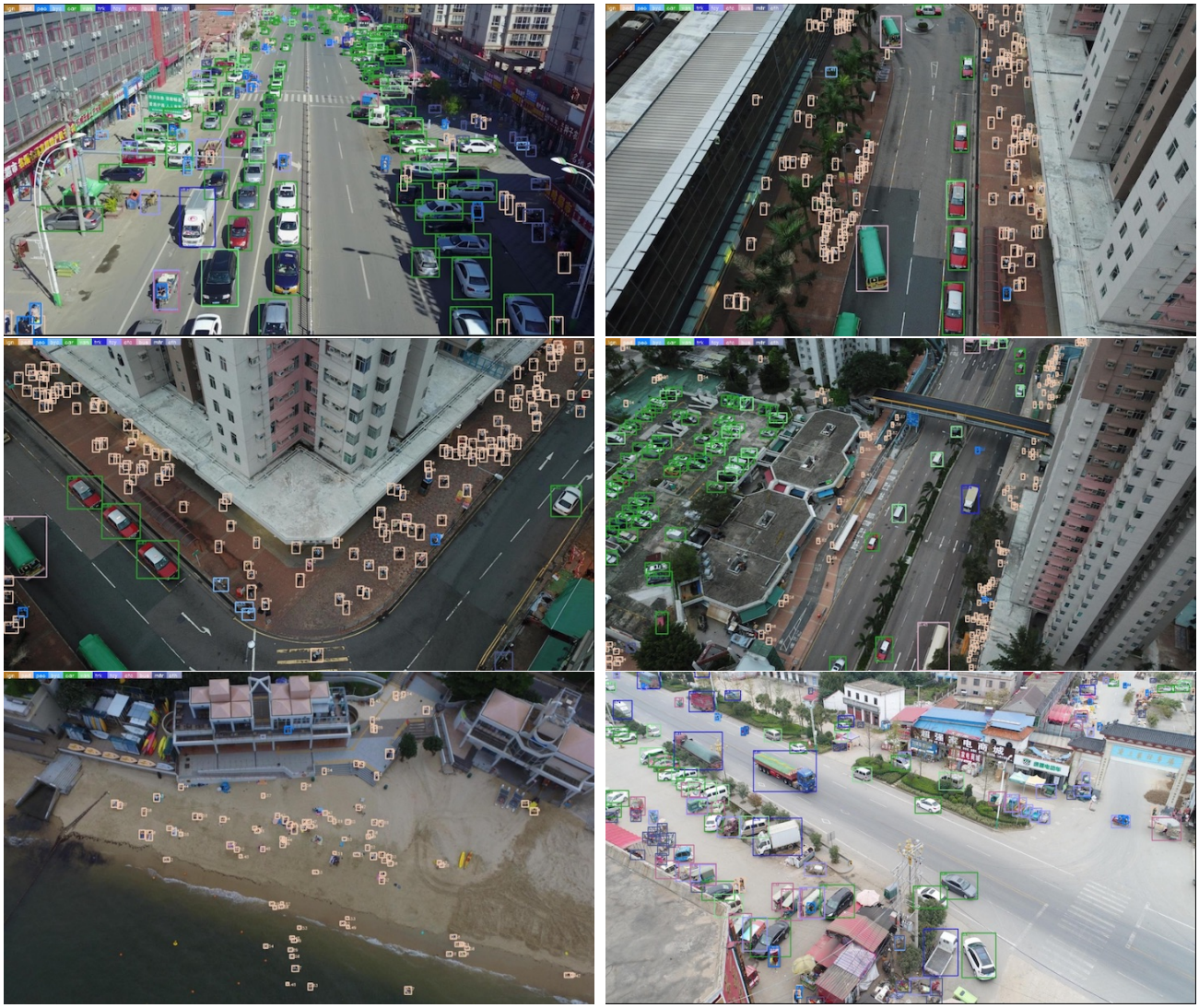


Figure 6. The visual result of our RRNet. Our network is very good at detecting small objects.

Module Name	AP50	AP75
Baseline	0.5974	0.3133
+ Random Resampling	0.6103	0.3232
+ AdaResampling	**0.6108**	**0.3288**

Table 3. The comparison between the proposed AdaResampling and the random resampling. Keeping the logical position prior is better for the data augmentation.

6.1. AdaResample *vs.* Randomly Resample

We propose the AdaResampling to keep the position prior in the drone captured images. Table 3 is the comparison between the AdaResampling and the random resampling. AdaResampling can significantly boost the mAP. The result demonstrates that breaking this prior is fatal for the detector training. Figure 7 is the visualization comparison of these two resampling strategy. The model trained with the random resampling generates more false-positive example.

6.2. Re-Regression Module

An evaluation of the proposed Re-Regression module is shown in Table 4. RR module improves mAP by $\sim$ 0.8 points. Ap75 are significantly boosted from 0.2958 to 0.3133.

In addition, we also evaluate mAP if we just use RR module for training, but abandon it during the testing phase.

Figure 7. The results visualization of the random resampling and the AdaResampling. The left column is the results of random resampling. The right column is for the AdaResampling. The read circle are the false-positive examples.

Setting	mAP	AP75
w/o RR Module	0.3214	0.2958
Train w/ RR, Test w/o RR Module	0.3241	0.3032
Train w/ RR, Test w/ RR Module	**0.3292**	**0.3133**

Table 4. The performance with or without the Re-Regression Module. Our RR Module improves mAP by $\sim$ 0.8 points. Notably, AP75 significantly increases from 0.2958 to 0.3133.

We achieve 0.3241 mAP, which is higher than the baseline. It illustrates that the gradient generated by the RR module is profitable for the backbone and the detector optimization.

6.3. Small object detection performance

We modify the official RetinaNet [12]. We firstly remove the last two FPN layer. Then, we use a K-Means algorithm to get the prior sets. The modified version achieves better performance (21.7 mAP) than the official version (18.4 mAP) on VisDrone2018 *val* set. Figure 8 is the visualization comparison between the modified RetinaNet and our RRNet. Obviously, our RRNet performs better on small object detection.

6.4. Other tricks

Synchronous batch normalization (SyncBN) is widely adopted by many semantic segmentation models. We also synchronize the mean and standard-deviation of BN cross multiple GPUs. This trick can freely improves the mAP from 0.3207 to 0.3292.

(a) (b)

Figure 8. (a) The results of RetinaNet. (b) The results of RRNet. Discarding the prior anchors makes our RRNet performs well on small object detection.

Moreover, we further employ the multi-scale strategy. We scale the images by 1.1x, 1.2x, 1.3x, 1.4x, and 1.5x times, and use the soft-nms to merge all the bounding boxes of all the scales level. It can significantly improves the mAP of our models about 1.5 points.

7. Conclusion

In this paper, we proposed an adaptive resampling augmentation and a hybrid object detector, the RRNet, for object detection on images captured by UAVs or drones. It presents excellent performance on very small objects in a dense scene. Our experiments demonstrated that RRNet outperforms the state-of-the-art. We achieve the highest performance of AP50, AR10, and AR100 in the *ICCV VisDrone2019 Object Detection in Images Challenge* [23].

8. Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. 61971388, U1706218), the Key Research and Development Program of Shandong Province (No. GG201703140154), Natural Science Foundation of Shandong Province (No. ZR2018ZB0852), and Applied Basic Research Programs of Qingdao (No.18-2-2-38-jch). This work got the GPU computation support from Center for High Performance Computing and System Simulation, Pilot National Laboratory for Marine Science and Technology (Qingdao).

References

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Soft-NMS - Improving Object Detection with One Line of Code. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5562–5570, 2017.

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into High Quality Object Detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018.

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv*, 2019.

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[7] Mate Kisantal, Zbigniew Wojna, Jakub Murawski, Jacek Naruniec, and Kyunghyun Cho. Augmentation for small object detection. *arXiv*, 2019.

[8] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, and Jianbo Shi. FoveaBox: Beyond Anchor-based Object Detector. *arXiv*, 2019.

[9] Hei Law and Jia Deng. CornerNet: Detecting Objects as Paired Keypoints. In *2018 European Conference on Computer Vision (ECCV)*, pages 765–781, 2018.

[10] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. Light-head r-cnn: In defense of two-stage object detector. *arXiv preprint arXiv:1711.07264*, 2017.

[11] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017.

[12] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2018.

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *2014 European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot MultiBox Detector. In *2016 European Conference on Computer Vision (ECCV)*, pages 21–37, 2016.

[15] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked Hourglass Networks for Human Pose Estimation. In *2016 European Conference on Computer Vision (ECCV)*, pages 483–499, 2016.

[16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2016.

[18] Longyin Wen, Pengfei Zhu, Dawei Du, Xiao Bian, Haibin Ling, Qinghua Hu, Chenfeng Liu, Hao Cheng, Xiaoyu Liu, Wenya Ma, Qinqin Nie, Haotian Wu, Lianjie Wang, Asanka G. Perera, Baochang Zhang, Byeongho Heo, Chunlei Liu, Dongdong Li, Emmanouil Michail, Hanlin Chen, Hao Liu, Haojie Li, Ioannis Kompatsiaris, Jian Cheng, Jiaqing Fan, Jie Zhang, Jin Young Choi, Jing Li, Jinyu Yang, Jongwon Choi, Juanping Zhao, Jungong Han, Kaihua Zhang, Kaiwen Duan, Ke Song, Konstantinos Avgerinakis, Kyuewang Lee, Lu Ding, Martin Lauer, Panagiotis Giannakeris, Peizhen Zhang, Qiang Wang, Qianqian Xu, Qingming Huang, Qingshan Liu, Robert Laganière, Ruixin Zhang, Sangdoo Yun, Shengyin Zhu, Sihang Wu, Stefanos Vrochidis, Wei Tian, Wei Zhang, Weidong Chen, Weiming Hu, Wenhao Wang, Wenhua Zhang, Wenrui Ding, Xiaohao He, Xiaotong Li, Xin Zhang, Xinbin Luo, Xixi Hu, Yang Meng, Yangliu Kuai, Yanyun Zhao, Yaxuan Li, Yifan Yang, Yifan Zhang, Yong Wang, Yuankai Qi, Zhipeng Deng, and Zhiqun He. Visdrone-sot2018: The vision meets drone single-object tracking challenge results. In *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part V*, pages 469–495, 2018.

[19] Li Zeming, Peng Chao, Yu Gang, Zhang Xiangyu, Yangdong. Deng, and Sun Jian. DetNet: Design Backbone for Object Detection. In *2018 European Conference on Computer Vision (ECCV)*, pages 339–354, 2018.

[20] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li. Single-shot refinement neural network for object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[21] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as Points. *arXiv*, 2019.

[22] Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, and Qinghua Hu. Vision Meets Drones: A Challenge. *arXiv*, 2018.

[23] Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, and Qinghua Hu. The VisDrone 2019. *http://aiskyeye.com/*, 2019.

[24] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Haibin Ling, Qinghua Hu, Qinqin Nie, Hao Cheng, Chenfeng Liu, Xiaoyu Liu, Wenya Ma, Haotian Wu, Lianjie Wang, Arne Schumann, Chase Brown, Qian Chen, Chengzheng Li, Dongdong Li, Emmanouil Michail, Fan Zhang, Feng Ni, Feng Zhu, Guanghui Wang, Haipeng Zhang, Han Deng, Hao Liu, Haoran Wang, Heqian Qiu, Honggang Qi, Honghui Shi, Hongliang Li, Hongyu Xu, Hu Lin, Ioannis Kompatsiaris,

Jian Cheng, Jianqiang Wang, Jianxiu Yang, Jingkai Zhou, Juanping Zhao, K. J. Joseph, Kaiwen Duan, Karthik Suresh, Bo Ke, Ke Wang, Konstantinos Avgerinakis, Lars Wilko Sommer, Lei Zhang, Li Yang, Lin Cheng, Lin Ma, Liyu Lu, Lu Ding, Minyu Huang, Naveen Kumar Vedurupaka, Nehal Mamgain, Nitin Bansal, Oliver Acatay, Panagiotis Giannakeris, Qian Wang, Qijie Zhao, Qingming Huang, Qiong Liu, Qishang Cheng, Qiuchen Sun, Robert Laganière, Sheng Jiang, Shengjin Wang, Shubo Wei, Siwei Wang, Stefanos Vrochidis, Sujuan Wang, Tiaojio Lee, Usman Sajid, Vineeth N. Balasubramanian, Wei Li, Wei Zhang, Weikun Wu, Wenchi Ma, Wenrui He, Wenzhe Yang, Xiaoyu Chen, Xin Sun, Xinbin Luo, Xintao Lian, Xiufang Li, Yangliu Kuai, Yali Li, Yi Luo, Yifan Zhang, Yiling Liu, Ying Li, Yong Wang, Yongtao Wang, Yuanwei Wu, Yue Fan, Yunchao Wei, Yuqin Zhang, Zexin Wang, Zhangyang Wang, Zhaoyue Xia, Zhen Cui, Zhenwei He, Zhipeng Deng, Zhiyao Guo, and Zichen Song. Visdrone-det2018: The vision meets drone object detection in image challenge results. In *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part V*, pages 437–468, 2018.

[25] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Haibin Ling, Qinghua Hu, Haotian Wu, Qinqin Nie, Hao Cheng, Chenfeng Liu, Xiaoyu Liu, Wenya Ma, Lianjie Wang, Arne Schumann, Dan Wang, Diego Ortego, Elena Luna, Emmanouil Michail, Erik Bochinski, Feng Ni, Filiz Bunyak, Gege Zhang, Guna Seetharaman, Guorong Li, Hongyang Yu, Ioannis Kompatsiaris, Jianfei Zhao, Jie Gao, José M. Martínez, Juan C. SanMiguel, Kannappan Palaniappan, Konstantinos Avgerinakis, Lars Wilko Sommer, Martin Lauer, Mengkun Liu, Noor M. Al-Shakarji, Oliver Acatay, Panagiotis Giannakeris, Qijie Zhao, Qinghua Ma, Qingming Huang, Stefanos Vrochidis, Thomas Sikora, Tobias Senst, Wei Song, Wei Tian, Wenhua Zhang, Yanyun Zhao, Yidong Bai, Yinan Wu, Yongtao Wang, Yuxuan Li, Zhaoliang Pi, and Zhiming Ma. Visdrone-vdt2018: The vision meets drone video detection and tracking challenge results. In *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part V*, pages 496–518, 2018.

[26] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning Data Augmentation Strategies for Object Detection. *arXiv*, 2019.