Spatial Attention for Multi-Scale Feature Refinement for Object Detection

Haoran Wang, Zexin Wang, Meixia Jia, Aijin Li, Tuo Feng, Wenhua Zhang, Licheng Jiao
School of Artificial Intelligence, Xidian University
Xi'an, Shaanxi Province, 710071, China
wanghaoran@stu.xidian.edu.cn lchjiao@mail.xidian.edu.cn

Abstract

Scale variation is one of the primary challenges in the object detection, existing in both inter-class and intra-class instances, especially on the drone platform. The latest methods focus on feature pyramid for detecting objects at different scales. In this work, we propose two techniques to refine multi-scale features for detecting various-scale instances in FPN-based Network. A Receptive Field Expansion Block (RFEB) is designed to increase the receptive field size for high-level semantic features, then the generated features are passed through a Spatial-Refinement Module (SRM) to repair the spatial details of multi-scale objects in images before summation by the lateral connection. To evaluate its effectiveness, we conduct experiments on VisDrone2019 benchmark dataset and achieve impressive improvement. Meanwhile, results on PASCAL VOC and MS COCO datasets show that our model is able to reach the competitive performance.

1. Introduction

Convolutional neural network (CNN) achieves great breakthrough in various kinds of fields in computer vision [7], however, object detection on the drone platform is still a challenging task [31]. Since the objects are of various scales due to different viewpoints both in inter-class and intra-class instances, large scale variation has became one of the main factors that affect the performance of the detection task [24] [27] [32] [25] [33]. Moreover, the viewpoints change even more dramatically in the drone image detection task. As shown in figure 1, the first two rows show the images from VisDrone2019 benchmark dataset [31]. In the first example, when the drone is at very high altitude, even the object of large size like a bus becomes difficult to be recognized. However, the bus in the second image shows clear characteristic when the drone is at normal altitude. The same thing happens to the person in the second example. The third row demonstrates the general problem in COCO dataset [14], and by comparison, it also shows that the problem is more serious in drone image detection.

Figure 1. The three sets of images from VisDrone2019 benchmark dataset [31] and COCO dataset [14] demonstrate the scale variation phenomenon occurs in intra-class objects. e.g. The sheep in the left image occupy extremely small part of space of the whole scene with a relatively smaller body than sheep in the right image. Meanwhile, we can see the scale differences between classes. Thus, detectors should correctly recognize them from different views with not only high, but also quite low resolution at the same time.

Generally, we can divide these CNN methods into two types: one stage methods like Yolo [18] or SSD [16] which predicts the final boxes directly from feed-forward CNN, and two stage methods like Faster R-CNN [20] or R-FCN [3] which predicts the results through preliminary proposals and refined features extracted from it. However, since the feature map form a single layer of the convolutional neural network has limited capacity of representation, recent works focus on feature fusion for object detection. A classical method is to combine low-level and high-level features

through a summation or concatenation operation. In this case, the lateral connection is proposed to add the features of backbone network to improve the ability to characterize objects with simple appearances. For example, FPN [12] use a top-down architecture and lateral connections to combine features at different depths.

As we all know, the receptive field becomes very large while the resolution becomes small when the feature maps are in deeper layers. As shown in figure 1, the remote small sheep in left image occupies a small part of the whole space of original picture, it will even become a little point in the last layer because of that reason, result in the huge difficulty in recognition. On the contrary, the same-category sheep in right image shows a bigger body, which will get more and more blurry in surroundings as the layer goes deeper. Both of the two images jointly show a simple example of scale variation in object detection task. However, there are two reasons that simply incorporating high-dimensional without any refinement is not representative enough for detection task. First, as the backbone network is trained for ImageNet [21] classification task, it is unbefitting to use the features directly for object detection. Second, the high-level feature maps are of fairly low spatial resolution, so that lack of the unbroken information to localize the large objects accurately or recognize the small objects.

SNIP [22] made an analysis that the scale of the smallest and largest 10% of object instances in COCO is 0.024 and 0.472 respectively, which results in scale variations of almost 20 times. They proposed Scale Normalization training scheme for Image Pyramids to achieve better performancewhile leaded to complex in practice. STDN [30] tried to utilize the Scale-Transfer Module to expand the resolution of the feature map for object detection, which destroy the original spatial location relation and channel information. RFBNet [15] developed a Receptive Field Block to strengthen the deep features, however, it focused on the one stage method with the last layer for prediction, which resulted in a lack of multi-scale information.

In order to obtain feature maps of sufficient representation, we propose a Receptive Field Expansion Block (RFEB) to increase the receptive field size for high-level semantic features, and a Spatial-Refinement Module (SRM) to repair the spatial details of multi-scale objects in images. In particular, we only add the SRM to the deep layers as the resolution of the shallow layers is large enough. Meanwhile, SRM take the spatial relation into consideration which is mainly different from the traditional FPN-based detectors, and the results show a consistent improvement in section 4.

Overall, Our contributions could be summarized into three-fold. 1) We introduce a Receptive Field Expansion Block and a Spatial-Refinement Module which generally refine features of traditional FPN-based model. 2) We achieve impressive improvement on VisDrone2019 benchmark dataset and competitive performance on two classic detection benchmarks. 3) We take the spatial relation into consideration to repair the spatial details of multi-scale objects in images to solve scale variation problem.

2. Related Work

Nowadays, almost all state-of-the-art methods are based on CNN networks, and have achieved dramatic breakthrough. We can divide these methods into two categories: one stage methods and two stage methods. YOLO [18] is the typical example of one stage methods, which avoids to generate the proposals, only uses a single feed-forward CNN network to directly predict the final bounding boxes. SSD [16] refers to the two stage method to add anchors mechanism, improving the accuracy further. The intuition of SSD is to use low-level features to detect small objects because high-level features suffer from the low resolution in a way that the information is highly aggregated in higher layers. However, feature maps from the first several convolution, which are crucial for detecting small objects ,was not really leveraged in SSD. DSSD [5] uses deconvolution layers to upgrade the SSD for more context information. In contrast, RCNN [6] series detectors are the base of the two stage methods, which utilize the backbone network to produce proposals first, then perform the predictions based on them.

Recently, scale variation brings great attention in object detection, an analysis of scale invariance in SNIP [22] shows that the variation in scale which a detector needs to handle is enormous, there is a big gap in size between largest and smallest scale object. Besides, the problem of domain-shift follows as the scale variation is really not the same as that on classification datasets.

Image pyramids was the mostly used method to deal with object detection at different scales before deep learning. Scale invariance is an inherent property of image pyramids since it is constructed by down-sampling the original image with Gaussian blur. However, an obvious limitation of this method is the computational resource it needs when processing one image, the model has to perform independent computation for images from all scales.

FPN addressed the problem in SSD by introducing the top-down connection to fuse features with high semantic meaning but low resolution and features with low semantic meaning but high resolution together. In order to match the size of these two kinds of feature maps, the feature map from higher layer was up-sampled before being added to the low-level feature map which is, at the same time, convolved by 1×1 kernels to obtain the same number of channels. It was called top-down architecture and lateral connections in FPN. Features at different depths were combined in this way to obtain multi-scale features for object detection.

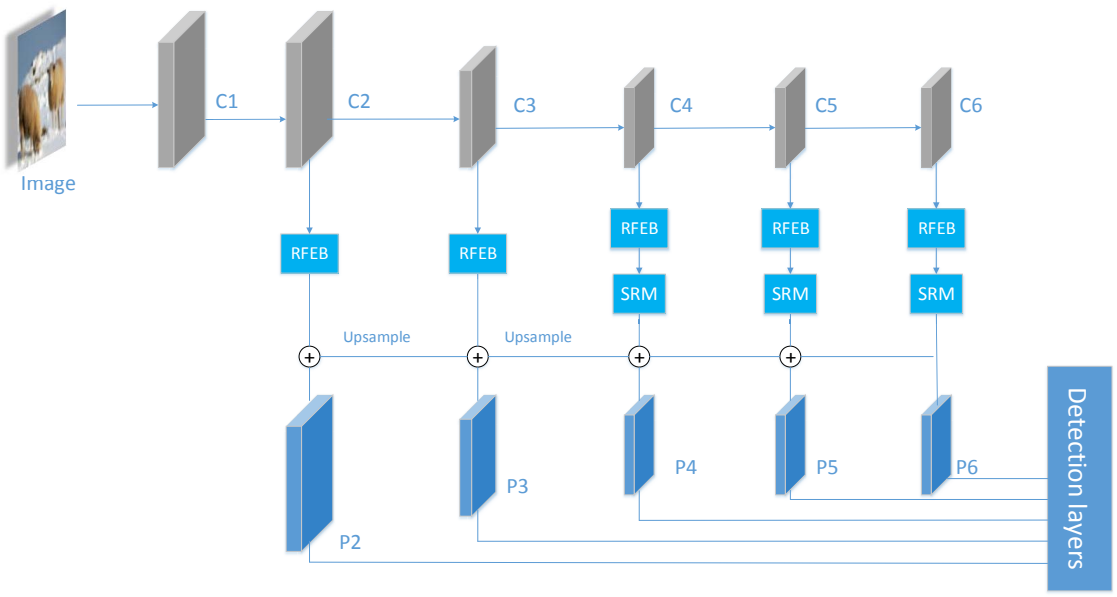

Figure 2. The proposed architecture is mainly based on the DetNet. It consists of three components. The top grey region represents the backbone feature extractor, including a ResNet-50 architecture and two extra layers. The bottom blue region represents the individual loss from stage 2 to stage 6, they make prediction independently. The middle region shows where the RFEB and SRM perform. Especially, the SRM is only added at the last three layers.

Recent advanced detectors take advantage of FPN architecture to solve the scale variation problems, such as RetinaNet [13] and RefineDet [28]. M2Det [29] designed a block of alternating joint Thinned U-shape Modules and Feature Fusion Modules to extract multi-scale features, then gathered up the feature maps with equivalent scales to construct the final feature, which increased computation because of the double network skeleton, greatly affecting the training and inference speed. Our method combines features of different layers and use Spatial-Refinement Module to repair feature maps with different resolutions, which can bring more context information than the other model without it, meanwhile keep the small additional computational cost.

3. Proposed Method

3.1. Network Architecture

In this section, we first introduce the base network which is our feature extraction network component, Spatial-Refinement Module (SRM) and Receptive Field Expansion Block (RFEB).

We adopt DetNet [11] as an example, which is a network modified on the basis of FPN. The improved backbone keeps the former 4 stages stay as original ResNet-50 [9], while adding the stage 5 and stage 6 with the same spatial resolution as the fourth stage. Finally, an arbitrary single-scale image is put into the network, it will output five feature maps from different fusion layers at multiple scales for prediction. In recent works, for a more in-depth study, researches on object detection have finely divided their views into the sub-problems of classification and localization respectively. As mentioned in [17], classification task requires the model to be invariant to various transformations while localization is more accurate if the model is transformation-sensitive. As we notice that scale variation influences both classification and localization, the model, on the one hand, has to be able to recognize the object at different scales, on the other hand, must adjust the bounding box accordingly. We propose RFE block and SRM to partially address the contradictory problem.

3.2. Receptive Field Expansion Block

Due to the increase of the receptive field of neurons, down-sampling does well in classification task, which is, however, not necessarily beneficial for object detection because localization may suffer from the absence of the fine location information. The proposed RFEB address the problem by using skip connection [9] to increase the receptive field. As shown in figure 3(a), the former two branches of RFEB decomposes a $k \times k$ convolution kernel into a $k \times 1$ and a $1 \times k$ kernels without any other activation function in between, leading to a bigger receptive field. [17] proposed a similar GCN module different from the separable kernels used by [23], enabling densely connections in a wider region in feature maps. What our block is different from both of them is that we add the third branch to integrate the o-

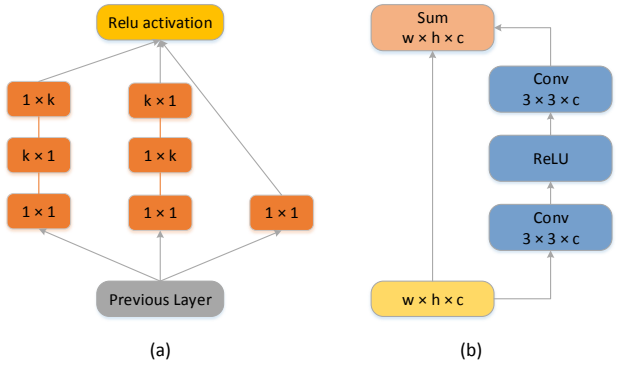

Figure 3. (a) shows the operation of Receptive Field Expansion Block (RFEB). From the bottom up, it decomposes a $k \times k$ convolution kernel into a $k \times 1$ and a $1 \times k$ kernels in the former two branches, with a skip connection operation in the third branch. (b) demonstrates the Spatial-Refinement Module (SRM) in detail, note it only added on the last three layers following RFEB.

riginal features back to it, guaranteeing the primary spatial relation to some extend.

3.3. Spatial-Refinement Module

SRM repairs the spatical information of the features refined by the RFEB. The activation in a deeper layer depends normally more on the center of the receptive field rather than the periphery which is, however, more important for localization. Thus, There are two convolution layers in the skip connection branch to model the residual between the input and the output of SRM, through which the contribution of the periphery of the receptive field is enhanced in order to boost localization. The SRM aims to increase the sensitiveness of boundary so as to reduce the impact of invalid context for regression on object location. Notably, when the network is relatively shallow and the feature map is relatively lager, the SRM is used prematurely to enforce regression boundaries can lose some meaningful context information, so the SRM is only applied on the C4, C5 and C6, rather than C2 and C3.

4. Experimental Results and Discussions

Data Augmentation. We use several data augmentation strategies presented in [16] to construct a robust model to adapt to variations of objects. That is, we randomly expand and crop the original training images with additional random photometric distortion and flipping to generate the training samples. Please refer to [16] for more details.

Hard Negative Mining. In the matching step, most of the anchor boxes are negatives, we use hard negatives mining to mitigate the extreme foreground-background class imbalance. We select some negative anchor boxes with top loss values to make the ratio between the negatives and positives below 1:1, rather than using all negative anchors.

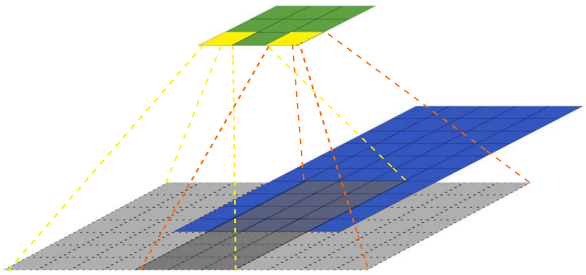

Figure 4. Two yellow activations from the green feature map has the same valid receptive field, which is the overlap of two gray receptive fields on the input blue image.

4.1. Discussion on Receptive Field of ResNet-50

ResNet-50 architecture is taken as the backbone of our model. The sizes of the receptive field for single activation after each convolution stage are 35×35, 99×99, 323×323 and 419×419 pixels, respectively. As illustrated in the figure 4 (the color channel is neglected for the brevity), the green one represents one channel from the feature map after several convolutions. Taking the two yellow activation as an example, even though the receptive fields are not identical (two squared gray regions), they overlap the same valid image region. More formally, denote the receptive field as RF, $RF(f_k(i,j))$ is the same for any j given a fixed i and k. Ideally, activation of the same channel should represent the same type of feature at different spatial positions. But an overlapped valid image region implies the homogeneity along the horizontal direction, thus, making the horizontal spatial relation less informative. So it is not necessary to distinguish the horizontal spatial positions in this case. The surroundings of the objects get more and more blurry due to this reason, especial for large instances.

4.2. VisDrone2019 benchmark dataset

Datasets and Protocols. We participate VisDrone-DET2019 challenge and it contains a large-scale drone-based object detection dataset, including 8599 images of ten object categories. The split is 6471 for training, 548 for validation, and 1580 for testing. The dataset was collected in different scenarios under various weather and lighting conditions. As a result, it is extremely challenging due to various factors, including large scale and pose variations, occlusion, and clutter background. We follow the protocol in [31] for VisDrone2019 benchmark dataset, and use the official evaluation toolkit.

Implementation Details. We adopt the Cascade RCNN [2] as our base network in VisDrone-DET2019 challenge

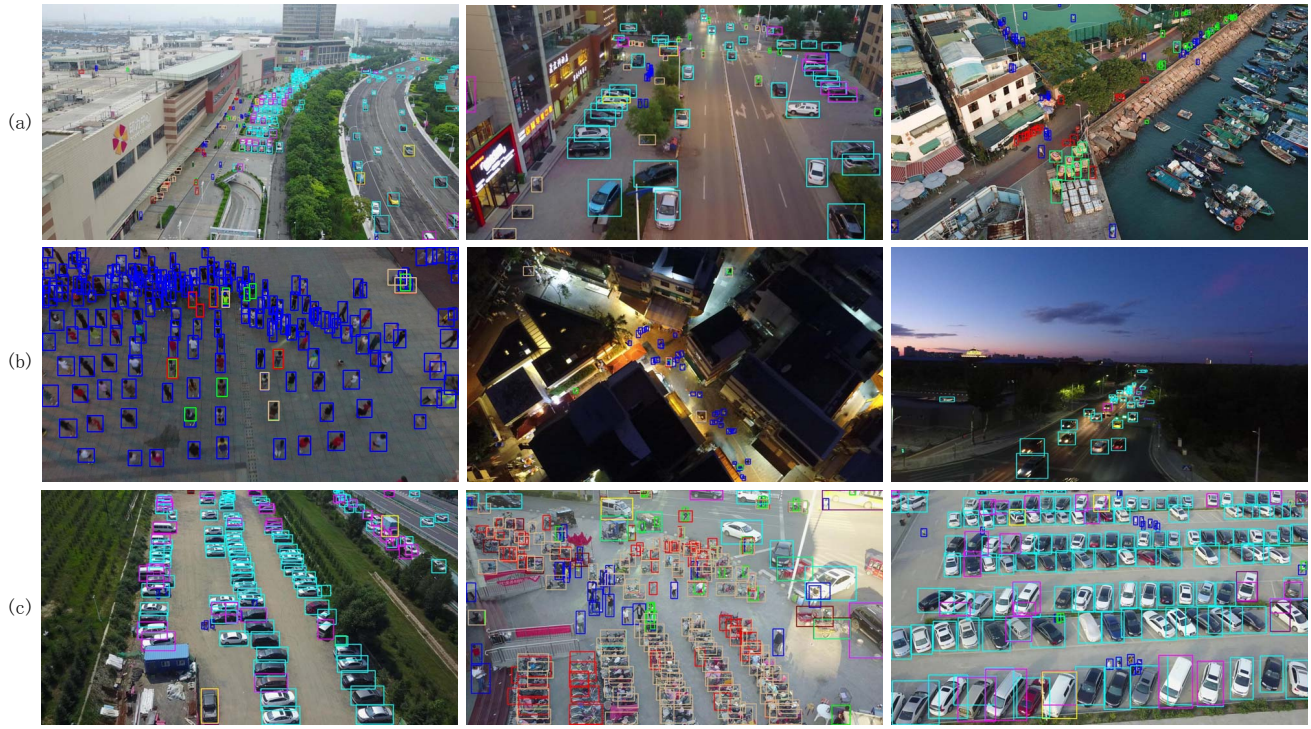

Figure 5. Results on VisDrone2019 benchmark dataset. (a), (b), (c) shows three different challenges in the data set, namely, small distant targets, light variation and dense targets with occlusion. And we can see that the scale of the object is very obvious both in intra-class and inter-class, including vehicles and people in different scenes.

Method	Avg. Precision, IoU:			Avg. Recall, maxDets:			
	0.5:0.95	0.5	0.75	1	10	100	500
Cascade RCNN	32.21	55.97	32.04	0.61	4.63	25.87	45.62
SAMFR-Cascade RCNN	33.72	58.62	33.88	0.53	3.40	22.60	46.03

Table 1. Comparison between the main results from cascade RCNN model and SAMFR-Cascade RCNN on VisDrone2019 validation dataset.

Method	pedestrian	people	bicycle	car	van	truck	tricycle	awning-tricycle	bus	motor
Cascade RCNN	32.40	20.62	17.71	58.07	35.89	30.44	23.13	11.32	44.83	28.24
SAMFR-Cascade RCNN	34.46	23.12	21.27	59.96	40.72	30.32	26.48	13.12	47.47	31.35

Table 2. Comparison between the results of ten categories from cascade RCNN model and SAMFR-Cascade RCNN on VisDrone2019 validation dataset.

and have made some changes for uav image detection. Improved by our two modules, the SAMFR-Cascade RCNN is proposed. Cascade R-CNN have four stages, one RPN and three for detection with IoU = $\{0.5, 0.6, 0.7\}$. The sampling of the first detection stage follows Fast R-CNN. In the following stages, resampling is implemented by simply using the regressed outputs from the previous stage. Our model uses the SGD as optimizer, with a weight decay of 0.0001 and momentum of 0.9 as default. We train the model with a minibatch size 2 per GPU. We start the learning rate at 0.02, and decrease it by a factor of 0.1. To warm-up the beginning 500 iteration for training, we use smaller learning rate of 0.02×0.3.

4.2.1 Experimental Results

We evaluate our method on VisDrone2019 benchmark dataset. The results are shown in figure 5. We show three common problems with data sets of small distant targets, light variation and dense targets with occlusion. Moreover, we have tested the performance on validation set, comparison between the results from cascade RCNN model and our SAMFR-Cascade RCNN are shown in table 1. All average precisions show consistent improvement, especially when

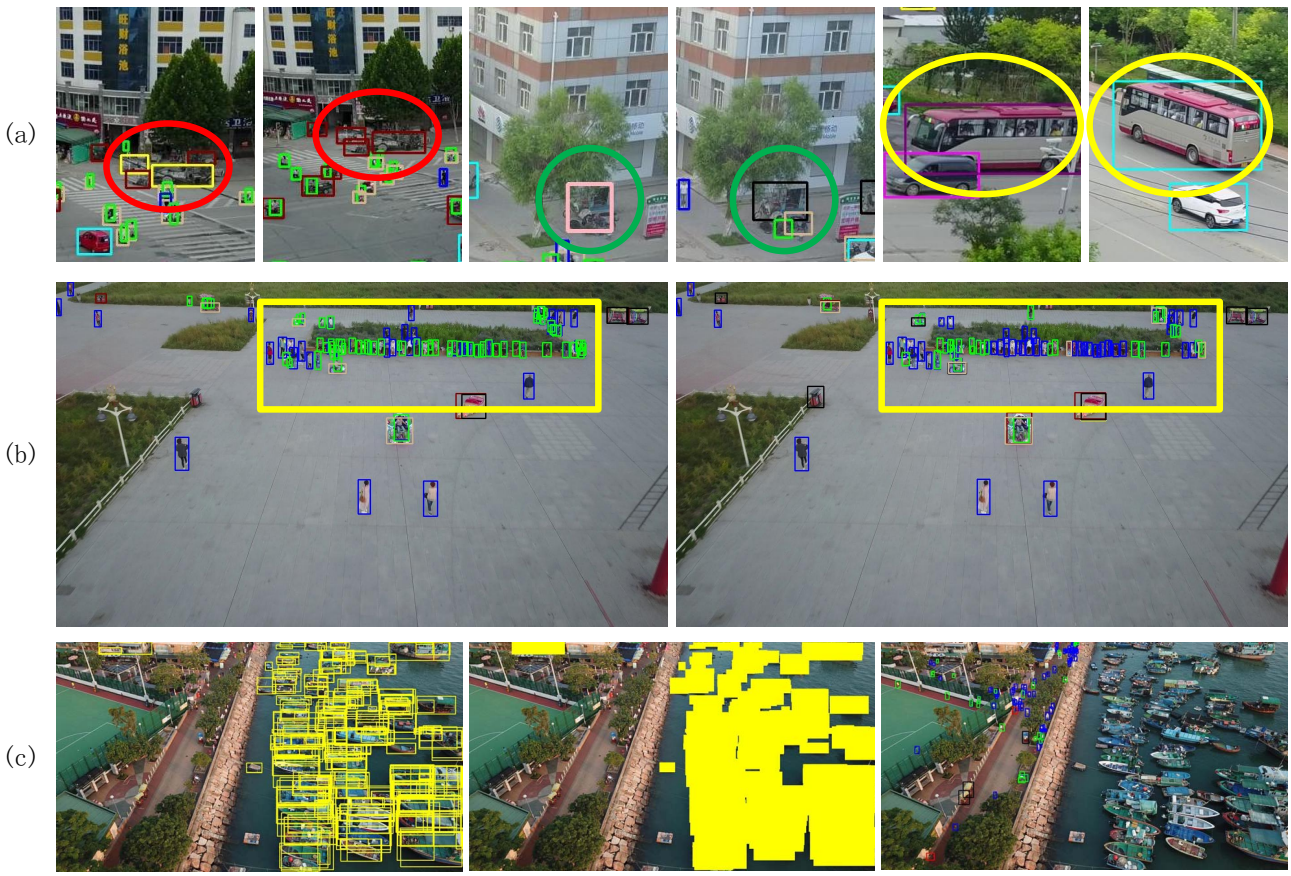

Figure 6. (a) shows incorrect labeling in training and validation set, (b) shows the model learned that the key to distinguishing people from pedestrians is the absence of feet. (c) shows the postprocessing with trained model on COCO for removing boats.

Method	Avg. Precision, IoU:			Avg. Recall, maxDets:			
	0.5:0.95	0.5	0.75	1	10	100	500
Retinanet [13]	11.81	21.37	11.62	0.21	1.21	5.31	19.29
RefineDet [28]	14.9	28.76	14.08	0.24	2.41	18.13	25.69
DetNet[11]	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
FPN [12]	16.51	32.2	14.91	0.33	3.03	20.72	24.93
Light-RCNN[26]	16.53	32.78	15.13	0.35	3.16	23.09	25.07
CornerNet[10]	17.41	34.12	15.78	0.39	3.32	24.37	26.11
SAMFR-Cascade RCNN	20.18	40.03	18.42	0.46	3.49	21.6	30.82

Table 3. Comparisons between the results from baseline methods and SAMFR-Cascade RCNN on VisDrone2019 test dataset.

the IoU threshold is 0.5. From comparison between the results of ten categories as shown in table 2, we can see improvements in almost every category with the exception of trunk which sometimes occupies the full image. Moreover, the results from the official on testing set are shown in table 3. We can see that the baseline method of Cascade RCNN obtains an mAP of 16.09% with [0.5:0.95] IoU, and our SAMFR-Cascade RCNN achieves a 4.09% improvement, which shows a consistent refinement on networks with our modules.

Postprocessing We attempt many postprocessing methods on testing set, and achieved performance improvement visually. For the first, we believe that there are some wrong labels in training set which have a serious influence on performance, for example, the objects in consecutive identical

Method	Backbone	Avg. Precision, IoU:			Avg. Precision, Area:		
		0.5:0.95	0.5	0.75	S	M	L
SSD300 [16]	VGG	25.1	43.1	25.8	6.6	25.9	41.4
SSD321 [5]	ResNet-101	28.0	45.4	29.3	6.2	28.3	49.3
DSSD321 [5]	ResNet-101	28.0	46.1	29.2	7.4	28.1	47.6
DSSD513 [5]	VGG	33.2	53.3	35.2	13.0	35.4	51.1
SSD513 [16]	ResNet-101	31.2	50.4	33.3	10.2	34.5	49.8
STDN513 [30]	DenseNet-169	31.8	51.0	33.6	14.4	36.1	43.4
YOLOv2 544 [19]	Darknet	34.9	55.7	37.4	15.6	38.7	50.9
RFBNet300 [15]	VGG	30.3	49.3	31.8	11.8	31.9	45.9
RFBNet512 [15]	VGG	33.8	54.2	35.9	16.2	37.1	47.4
Faster R-CNN [20]	VGG	24.2	45.3	23.5	7.7	26.4	37.1
R-FCN [3]	ResNet-101	29.9	51.9	-	10.8	32.8	45.0
CoupleNet [34]	ResNet-101	34.4	54.8	37.2	13.4	38.1	50.8
Faster R-CNN [20]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Deformable R-FCN [4]	Aligned-Inception-ResNet	37.5	58.0	40.8	19.4	40.1	52.5
Mask R-CNN [8]	ResNext-101	37.1	60.0	39.4	16.9	39.9	53.5
RetinaNet[13]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
Ours	DetNet-59	40.7	62.6	43.7	23.2	42.8	50.0

Table 5. The top box in the tabel are one-stage methods, and bottom box are two-stage methods. Detection results on MS COCO $test-dev$ set.

Method	Backbone	FPS	mAP
SSD300 [16]	VGG	120	77.2
YOLOv2 544 [19]	Darknet	40	78.6
SSD512 [16]	VGG	50	79.8
RFBNet300 [15]	VGG	83	80.5
RFBNet512 [15]	VGG	38	82.2
Faster [20]	VGG	7	73.2
Faster [20]	ResNet-101	5	76.4
DSSD321 [5]	VGG	9.5	78.6
R-FCN [3]	ResNet-101	9	80.5
STDN513 [30]	DenseNet-169	-	80.9
Ours	DetNet-59	10	81.9

Table 4. Comparison between our model and other state-of-the-art methods. Attention that our backbone is ResNet-101 with fewer parameters.

scenes are labeled differently as shown in figure 6 (a). We first remove the images with the ignored areas, and then we select some of good quality from them. The visual results change a lot as shown in figure 6 (b), especially in people and pedestrians. It seems that the model learns that the key to distinguishing people from pedestrians is the absence of feet. Second, we analyze the domain style between training set and testing set which has a big gap, there are many scenarios and objects that never appear before, the angle of view and shooting height are also obviously different. We adopt the semi-supervised training strategy for the problem to learn the specific characteristic of some vehicles on testing set. Meanwhile, we use the model trained on COCO dataset to remove some interferential objects that belong to boats. The results are demonstrated in figure 6 (c). At the end, we used soft-NMS technique, the difference is that we also implement it between fine-grained classes, such as people and pedestrian.

4.3. PASCAL VOC and MS COCO

Datasets and Protocols. The experiments are also conducted on two large-scale datasets PASCAL VOC and MS COCO that have 20 and 80 object categories respectively. We adopt DetNet as our backbone network. We follow the protocol in [6] for PASCAL VOC, test on the VOC2007 test set while use VOC2007 trainval and VOC2012 trainval for training. We follow the standard protocol for COCO, train on the 120k images in the trainval and test on the 20k images in the test-dev. Meanwhile, we use the mean average precision (mAP) scores for evaluation, we test the mAP scores using IoU thresholds at 0.5. For COCO, we report the results following the standard metric. We use a similar setting to [11]. The backbone model is a pre-trained modified ResNet-50 from ImageNet, and the FPN-based DetNet is used in both two experiments. And we follow the same training strategies described as in VisDrone2019 4.2.

4.3.1 PASCAL VOC

All models are trained on the VOC2007 and VOC 2012 trainval set, tested on the VOC 2007 test set. We set the learning rate to 4×10^3 learning rate, use the default batch size 32 in training, and only adopt DetNet-59 as the back-

bone network for all the experiment on the PASCAL VOC dateset, including VOC 2007 and VOC 2012.

Table 4 shows our results evaluated on the PASCAL VOC 2007 testing set, the proposed architecture achieves competitive performance on the backbone of DetNet-59 with fewer parameters as shown in table 4. In these results, the final feature maps used for prediction in one stage methods are extracted from different single layer respectively, which means it predicts the results without fusing features, demonstrating a relatively smaller value than FPN-based methods. However, they show a big advantage in the aspect of speed. Most of the state-of-the-art methods utilize ResNet-101 as their backbone, because of more parameters with more powerful capacity. It demonstrates that our network achieves the comparative mAP based on the DetNet backbone, which is a modified version of ResNet-50, the accuracy is even 1.4% higher than R-FCN which is with ResNet-101 backbone. At the mean while, we still get the fastest speed of all the two stage methods.

4.3.2 MS COCO

To further validate the proposed method, we carry out experiments on the MS COCO dataset. With a larger scale than PASCAL VOC, the detection methods with ResNet-101 usually achieve better performance than those with VGG on MS COCO. In addition, note that test-dev dataset is different from mini-validation dataset used in experiments. It has no disclosed labels and is evaluated on the server. Following the protocol in MS COCO, we use the trainval 35k set [1] for training and evaluate the results from test-dev evaluation server.

Table 5 shows the results on Ms COCO test-dev set. As shown in the last column, the AP of large objects gets lower improvement than the other two scales, but in the other columns we get consistent increase, achiveing an mAP of 40.7% with [0.5:0.95] IoU, 62.6% with 0.5 IoU and 43.7% with 0.75 IoU.

5. Conclusion

In this paper, we focus on the problem of scale variation in object detection, especially on the drone platform. We propose a Receptive Field Expansion Block (RFEB) to increase the receptive field size for high-level semantic features, then the generated features are passed through a Spatial-Refinement Module (SRM) to repair the spatial details of multi-scale features, two available FPN-based networks are implemented to verify feasibility and generality. Based on them, we achieve impressive improvement on VisDrone2019 benchmark dataset. Meanwhile, experiments on two general datasets also show that the our techniques can help model to achieve the competitive performance compared with other advanced methods. Furthermore, we expect a broader range of applications on the drone platform in the future.

References

[1] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2874–2883, 2016.

[2] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018.

[3] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[4] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[5] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[6] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[11] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Detnet: A backbone network for object detection. 2018.

[12] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[14] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick. Microsoft coco: Common objects in context. 2014.

[15] S. Liu, D. Huang, et al. Receptive field block net for accurate and fast object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 385–400, 2018.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[17] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361, 2017.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[19] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[20] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[22] B. Singh and L. S. Davis. An analysis of scale invariance in object detection snip. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3578–3587, 2018.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[24] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: Fully convolutional one-stage object detection. 2019.

[25] L. Wen, P. Zhu, D. Du, X. Bian, H. Ling, and Q. H. et al. 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.

[26] X. Wu, R. He, Z. Sun, and T. Tan. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 13(11):2884–2896, 2018.

[27] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling. Clustered object detection in aerial images. *CoRR*, abs/1904.08008, 2019.

[28] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. Single-shot refinement neural network for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4203–4212, 2018.

[29] 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. *arXiv preprint arXiv:1811.04533*, 2018.

[30] P. Zhou, B. Ni, C. Geng, J. Hu, and Y. Xu. Scale-transferrable object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 528–537, 2018.

[31] P. Zhu, L. Wen, X. Bian, L. Haibin, and Q. Hu. Vision meets drones: A challenge. *arXiv preprint arXiv:1804.07437*, 2018.

[32] P. Zhu, L. Wen, D. Du, X. Bian, and H. L. et al. 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.

[33] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, and Q. N. et al. 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.

[34] Y. Zhu, C. Zhao, J. Wang, Z. Xu, W. Yi, and H. Lu. Couplenet: Coupling global structure with local parts for object detection. 2017.