ATTENTION-ENHANCED AND MORE BALANCED R-CNN FOR OBJECT DETECTION

Ruohong Mei, Haiying Wang, Aidong Men

Beijing University of Posts and Telecommunications, Beijing, China

ABSTRACT

Attention mechanisms have been widely used in deep neural convolution networks and different fields, such as object detection and instance segmentation. Many attention mechanisms will cost too much calculation, so in this paper, we incorporate a kind of light attention mechanism, the attended residual module, into our object detection backbone to get an accuracy-efficiency trade-off. Besides, to solve the imbalance problem in region sample level, we use the cascade region proposal network(RPN) module to gain anchors of higher quality resulting in higher average recall(AR). Furthermore, we replace the non-local attention module in feature fusion level with the criss-cross attention module to reduce computation and improve performance. With them all, our method significantly improves the detection performance and achieves 43.6 AP in COCO *test-dev*.

***Index Terms*—** Object detection, attended residual module, cascade RPN, criss-cross attention module

1. INTRODUCTION

As attention mechanism has become so popular in natural language processing(NLP) that it has been quickly adopted by computer vision and achieved good performance. Different variants of attention networks, such as relation networks [1], nonlocal neural networks [2] and CCNet [3] have been used to detection and recognition tasks. But it is still a question of how attention mechanisms function and which variant of attention mechanism is more effective with respect to different tasks. Lately, an empirical study [4] provides some advice and also points out that there is still much room for improvements on spatial attention mechanisms.

Besides, with the development of deep convolution neural network, lots of object detection frameworks have been proposed and have made significant progress, such as Faster RCNN [5] of two-stage style and RetinaNet [6] of one-stage style. Whether it is a two-stage or a one-stage style algorithm, they all follow a general rule: sampling regions, extracting features from them, classifying objects, and regressing detection boxes through multi-task targets. This has led to a series of imbalance problems at different levels. Some researchers have started from this aspect and achieved good results. In sampling level, due to too many negative examples, iterative refinement for region proposals [7] and Cascade RPN(region proposal network) [8] are proposed to generate high-quality anchors to get high average recall(AR). In feature level, FPN [9], PANet [10] and BiFPN [11] are proposed to fuse multi-level features. In objective level, classification aware regression loss [12] is proposed to balance classification and localization tasks. Recently, Libra R-CNN [13] has gained good performance on these three parts, so in this paper, we take this as our baseline.

In this paper, we propose an attention-enhanced and more balanced R-CNN based on Libra R-CNN [13]. Compared with baseline, we summarize our main contributions:

(1) In backbone network, we add attended residual modules and gain significant improvements on mean average precision(AP) with comparison to the baseline.

(2) In sample level, we use cascade RPN instead of standard RPN to get more and positive samples of higher quality, and the AR increases.

(3) In feature fusion level, we replace the non-local module with the criss-cross attention module to refine fused features, which needs less computation and GPU memory usage, but gets higher mAP than the former.

In section 2, we will analyze our proposed methods in detail. In section 3, we will illustrate our experiments and show our improvements.

2. THE PROPOSED METHODS

The overall architecture is shown in Figure 1. “TAM” and “DCM” compose the attended residual module [4], which is used to enhance attention. Cascade RPN [8] is used to get positive samples with high quality to get more balanced samples resulting in high AR. Criss-cross attention module [3] is used to reduce computation and GPU memory usage for BFP in [13]. All components are to be detailed in the following sections.

2.1. Attended residual module

Attention mechanism has been widely used in various fields of deep learning in recent years. It can be seen frequently in

978-1-7281-6395-6/20/$31.00 ©2020 IEEE

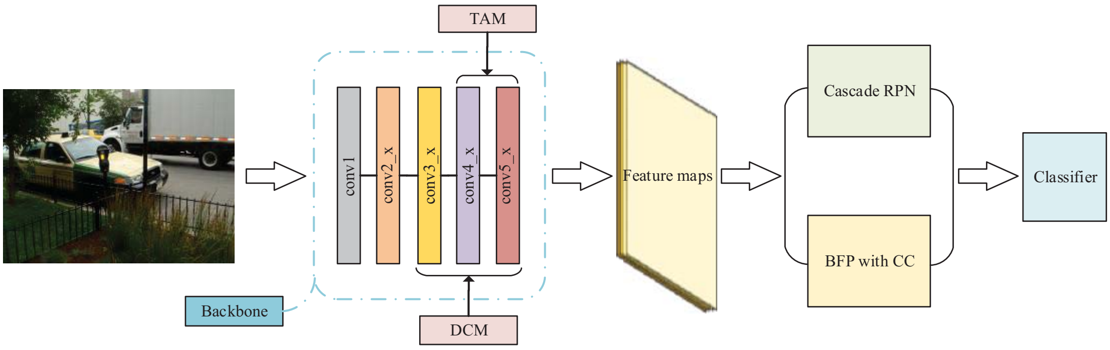


Fig. 1. Overall architecture of the proposed network. The backbone we use is ResNet [14]. "TAM" and "DCM" denote Transform attention module and deformable convolution module respectively. [15] which is a kind of attention mechanism as well as Transform attention module [16]. "BFP with CC" means balanced feature pyramid [13] with criss-cross module [3].

varieties of tasks, such as NLP, speech recognition, and image processing. In this paper, we also implement attention mechanisms on our algorithm. There are four possible attention factors:(1) the query and key content, (2) the query content and relative position, (3) the key content only, and (4) the relative position only. The query and key are visual elements such as image pixels or regions of interest in recognition tasks such as object detection and instance segmentation. These four factors are expressed as a sum of 4 parameters $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ in Transformer attention [16]. Let q denote a query element with content c_q, and k denote a key element with content c_k. So we compute Transform attention as

$$A_m^{Trans}(q, k, c_q, c_k) \propto \exp(\sum_{j=1}^{4} \beta_j^{Trans} \lambda_j), \tag{1}$$

where $A_m^{Trans}(q, k, c_q, c_k)$ indicates Transform attention weights in the m-th attention head, which is normalized by $\sum_{k\in\Omega_q} A_m^{Trans}(q, k, c_q, c_k) = 1$ where Ω_q specifies the supporting key region for the query, and β_j^{Trans} takes values in $\{0, 1\}$ to denote whether we use λ_j or not. For example, "0010" means we consider λ_3 only.

Together with Transform attention, we use deformable convolution module [15] which is computed as

$$A_m^{deform}(q, k, c_q) = G(k, q + p_m + w_m^T x_q). \tag{2}$$

where p_m indicates a predetermined offset, and $w_m^T x_q$ projects the query content x_q to a defomation offset according to a learnable vector w_m. $G(\cdot)$ is the bilinear interpolation kernel in N-dimension space.

Now we can combine these two attention mechanisms to build the attended residual module as is shown in Figure 2. In this paper, we use ResNet [14] as our backbone. The Transformer attention module [16] is incorporated by applying it on 3*3 output in the residual block of ResNet [14]. And it replaces 3*3 standard convolution with deformable convolution [15]. In deformable convolution [15], learnable offsets are used to adjust the sample position of the key elements and predicted according to the query content, thus dynamic to the input, so deformable convolution can also be viewed as a kind of attention mechanism. Meanwhile, the deformable convolution is similar to λ_2, which is also base on query content and relative positions. But, compare to λ_2, deformable convolution samples a sparser set of key elements for each query so that it is much faster than λ_2 for image recognition and semantic segmentation. So in [4], for object detection and semantic segmentation, the configuration of "0010 + deformable" is recommended for an optimal accuracy-efficiency trade-off compared with other configurations. The experiment details are explained in Section 3.

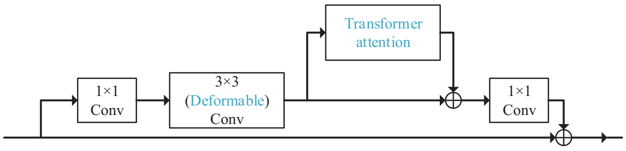


Fig. 2. Attended residual module

2.2. Cascade RPN

For standard RPN, anchors are defined by multiple scales and ratios. Based on this design, we can get multi-scale anchors in a single scale image. However, a large number of negative samples will be generated through standard RPN, even after non-maximum suppression(NMS). Libra R-CNN [13] adopts IoU-balanced sampling after standard RPN. It divides anchors according to the IoU with ground truths and samples more anchors with higher IoU than random sampling. Compared with IoU-balanced sampling, cascade RPN balances anchors when generating anchors. Two-stage style cascade RPN is shown in Figure 3.

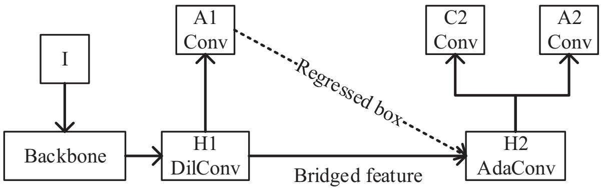
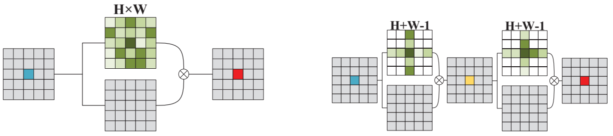


Fig. 3. The architecture of cascade RPN. ”I”, ”H”, ”A”, ”C” mean input image, network head, anchor regressor and classifier respectively. ”DilConv”and ”AdaConv” denote dilated convolution [17] and adaptive convolution [8] respectively.

Feature maps obtained from the backbone will be put into the cascade RPN module. In the first stage, we use dilated convolution [17] to adjust predefined anchors and align features to generate regressed anchors. The features in the first stage are ”bridged” to the next stage. In the generating stage, features of classifying foreground and background and anchors are generated by the adaptive convolution generating $2k$ classification scores and $4k$ coordinates (if there are k anchors) through two separate convolution layers like standard RPN. At last, it generates final region proposals after NMS. Then we will get proposals of higher quality through cascade RPN, resulting in more positive samples and higher IoU samples to gain higher AR, which balances the number of positive and negative samples.

2.3. Criss-cross attention module

In [13], the non-local attention module [2] serves as a refine module which can be realized by convolutions directly while the latter works less stable. It should be noted that non-local module will consume high GPU memory and need lots of computation FLOPs. In this paper, we replace the non-local module with a criss-cross attention module [3], which is GPU memory friendly and needs fewer FLOPs. The diagrams of non-local module and criss-cross module are shown in Figure 4.



(a) Non-local module (b) Criss-cross module

Fig. 4. Comparison between the non-local module and the criss-cross module

For each position, criss-cross module generates a coarse map with weights of H+W-1. After recurrent operation, each position in the final output feature map can capture long-distance correlation from all pixels. By stacking these two criss-cross modules sequentially, we can reduce complexity in time and space from $\mathcal{O}((H*W)*(H*W))$ which is needed by non-local module to $\mathcal{O}((H*W)*(H+W-1))$.

3. EXPERIMENTS

3.1. Datasets and experiments settings

In this paper, we use the challenging MS COCO dataset [18] to evaluate our experiments. For fair comparisons, we implement Libra R-CNN and our method on mmdetection [19]. We train our models with a single RTX 2080Ti for 12 epochs and use SGD to optimize. The initial learning rate is set to 0.02, and is decreased by 0.1 after 8 and 11 epochs.

3.2. Main results

We compare our proposed method with the baseline, Libra R-CNN, and some other state-of-the-art approaches on the COCO *test-dev* in Table 1. All these results follow COCO’s standard AP and AR metrics. Through the overall design, our method achieves 43.6 AP with ResNet-101-FPN on COCO *test-dev*. Compared with Libra R-CNN*, our method brings 3.9 points higher AP with ResNet-50 and 3.1 points higher AP with ResNet-101 respectively.

3.3. Ablation study

All these ablation experiments have been tested on COCO *val-2017*. Besides, for the sake of brevity, “R-50-F” and “R-101-F” mean ResNet-50-FPN and ResNet-101-FPN respectively.

3.3.1. Overall ablation study

Results are shown in Table 2. Only the attended residual module enhances AP largely, which is 3.1 points higher. The cascade RPN module focuses on proposals generation and higher AR, so it contributes a little to AP. The criss-cross attention module uses fewer parameters but gains a little higher AP than the non-local module used in Libra R-CNN [13].

Table 2. Overall ablation study on COCO *val-2017*. Backbone is R-101-F. “ARM” denotes the attended residual module.

Cascade RPN	Criss-cross	ARM	AP
			40.3
√			40.6
	√		40.7
		√	43.4
√	√	√	**43.6**

Table 1. Comparisons with Libra R-CNN[13] and other state-of-the-art methods on COCO *test-dev*.The symbol * means our re-implemented metheds. The 1× training schedule follows the configurations detailed in Detectron[20].

Method	Backbone	Schedule	AP	AP_{50}	AP_{75}	AP_{S}	AP_{M}	AP_{L}
Faster R-CNN[5]	ResNet-101-FPN	-	36.2	59.1	39.0	18.2	39.0	48.2
Mask R-CNN[21]	ResNet-101-FPN	-	38.2	60.3	41.7	20.1	41.1	50.2
RetinaNet[6]	ResNet-101-FPN	-	39.1	59.1	42.3	21.8	42.7	50.2
Libra R-CNN[13]	ResNet-50-FPN	1×	38.7	59.9	42.0	22.5	41.1	48.7
Libra R-CNN[13]	ResNet-101-FPN	1×	40.3	61.3	43.9	22.9	43.1	51.0
Libra R-CNN*	ResNet-50-FPN	1×	38.5	59.6	41.9	22.8	42.1	48.9
Libra R-CNN*	ResNet-101-FPN	1×	40.5	61.2	44.2	23.4	44.3	53.0
Ours	ResNet-50-FPN	1×	**42.4**	**61.4**	**46.6**	**23.3**	**44.9**	**55.9**
Ours	ResNet-101-FPN	1×	**43.6**	**62.6**	**47.8**	**23.9**	**46.5**	**57.8**

3.3.2. *Attended residual module*

As mentioned in Sec 3, we use the configuration of "0010 + dcn". For "0010" Transformer attention modules, we embed them into conv4_x and conv5_x. For deformable convolution module, we embedded them into conv3_x, conv4_x and conv5_x which could be seen in Figure 1. It should be noted that there are 6 Basic modules in conv4_x of ResNet-50 while 23 Basic modules in conv4_x of ResNet-101. To reduce calculation, we just add "0010" Transformer attention modules in the first 6 Basic modules of ResNet-50 and ResNet-101. Results are shown in Table 3 . We get 3.8 points and 3.1 points higher than the baseline with R-50-F and R-101-F respectively.

Table 3. Ablation study for "0010+dcn" attended residual module. The symbol * means our re-implements.

Method	Backbone	AP
Libra R-CNN*	R-50-F	38.5
Libra R-CNN*	R-101-F	40.3
+ "0010+dcn"	R-50-F	**42.3**
+ "0010+dcn"	R-101-F	**43.4**

3.3.3. *Cascade RPN*

We implement two-stage Cascade RPN into our method, and results are shown in Table 4. Our method gets 0.6 and 0.5 points enhancement in AR^{100} with R-50-F and R-101-F respectively.

3.3.4. *Criss-cross attention module*

We replace the non-local module of BFP in Libra R-CNN with the criss-cross module, and results are shown in Table 5. We report the GPU memory as the maximum value of "torch.cuda.max_memory_allocated()". Memory slightly decreases, but AP slightly increases because this module is just a small part of the whole neural network resulting in limited improvements.

Table 4. Ablation study for cascade RPN. "CRPN" denotes cascade RPN. The symbol * means our re-implements. Backbone is ResNet-50-FPN.

Method	Backbone	AR^{100}	AR_{S}	AR_{M}	AR_{L}
Libra RPN*	R-50-F	53.8	35.4	58.1	67.2
Libra RPN*	R-101-F	55.5	37.2	60.0	69.5
+ "CRPN"	R-50-F	**54.4**	**36.0**	**59.1**	**69.7**
+ "CRPN"	R-101-F	**56.0**	**37.6**	**60.9**	**72.0**

Table 5. Ablation study for criss-cross module. "CC" denotes criss-cross module.The symbol * means our re-implements.

Method	Backbone	Memory(GB)	AP
Libra R-CNN*	R-50-F	4.3	38.5
Libra R-CNN*	R-101-F	6.3	40.3
+ "CC"	R-50-F	**4.0**	**38.8**
+ "CC"	R-101-F	**5.9**	**40.6**

4. CONCLUSION

In this paper, we have proposed an attention-enhanced and more balanced R-CNN for object detection, which achieves better performance than the baseline. The attended residual module, the cascade RPN module, and the criss-cross attention module improve the performance of our method in AP, AR, and GPU memory usage respectively. With all of them, our method significantly improves detection performance than the baseline and other state-of-the-art approaches.

5. ACKNOWLEDGEMENTS

This work is partially supported by MoE-CMCC ”Artifical Intelligence” Project No. MCM20190701 and Beijing Key Laboratory of Network System and Network Culture(Beijing University of Posts and Telecommunications).

6. REFERENCES

[1] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei, "Relation networks for object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[2] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He, "Non-local neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[3] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[4] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai, "An empirical study of spatial attention mechanisms in deep networks," *CoRR*, vol. abs/1904.05873, 2019.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[6] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar, "Focal loss for dense object detection," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[7] Qiaoyong Zhong, Chao Li, Yingying Zhang, Di Xie, Shicai Yang, and Shiliang Pu, "Cascade region proposal and global context for deep object detection," *ArXiv*, vol. abs/1710.10749, 2017.

[8] Thang Vu, Hyunjun Jang, Trung X Pham, and Chang D Yoo, "Cascade rpn: Delving into high-quality region proposal network with adaptive convolution," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[9] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature pyramid networks for object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[10] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia, "Path aggregation network for instance segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[11] Mingxing Tan, Ruoming Pang, and Quoc V. Le, "Efficientdet: Scalable and efficient object detection," *ArXiv*, vol. abs/1911.09070, 2019.

[12] Yuhang Cao, Kai Chen, Chen Change Loy, and Dahua Lin, "Prime sample attention in object detection," *CoRR*, vol. abs/1904.04821, 2019.

[13] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin, "Libra r-cnn: Towards balanced learning for object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[15] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei, "Deformable convolutional networks," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017.

[17] Fisher Yu and Vladlen Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," *arXiv e-prints*, p. arXiv:1511.07122, Nov 2015.

[18] 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 *Computer Vision – ECCV 2014*, 2014.

[19] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[20] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He, "Detectron," https://github.com/facebookresearch/detectron, 2018.

[21] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick, "Mask r-cnn," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.