Joint COCO and Mapillary Workshop at ICCV 2019: COCO Instance Segmentation Challenge Track

Technical Report: Bags of Improvements for Instance Segmentation

Qi Wang* Li Hu* Bang Zhang Pan Pan
Alibaba Group
{wilson.wq, hooks.hl, zhangbang.zb, panpan.pp}@alibaba-inc.com

[1]

Abstract

In this report, we propose several effective improvements for the proposal-box-mask pipeline in instance segmentation. We achieve 51.2 Mask AP on COCO [8] test-dev dataset, which is 2.2 higher than the winning approach of COCO Challenge 2018.

1. Introduction

Proposal-box-mask is an effective pipeline for instance segmentation, such as Mask RCNN [4]. However, This method relies on the good enough results of every stage. The final results are determined by the quality of proposals, the accuracy of boxes and correct evaluation of mask results. In this work, we proposed some improvements for every part in this pipeline to further boost the performance. Together with all improvements, we achieve 51.2 Mask AP on COCO test-dev dataset, which is 2.2 higher than the winning approach of COCO Challenge 2018 as illustrated in Figure 1.

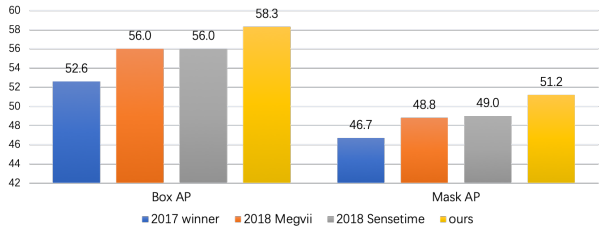


Figure 1: Final result on COCO test-dev dataset

2. Methods

Overview: We follow Hybrid Task Cascade (HTC) [3] to begin our work. Compared to existing frameworks, it is distinctive in several aspects: (1) Enhanced Box Header is proposed to predict accurate boxes. (2) Cascade Mask Re-Scoring aims to evaluate mask quality in cascade pipeline. (3) Refining strategy is introduced for training from coarse to fine. (4) Scale Aware Inference is proposed to handle scale problem.

2.1. Enhanced Box Header

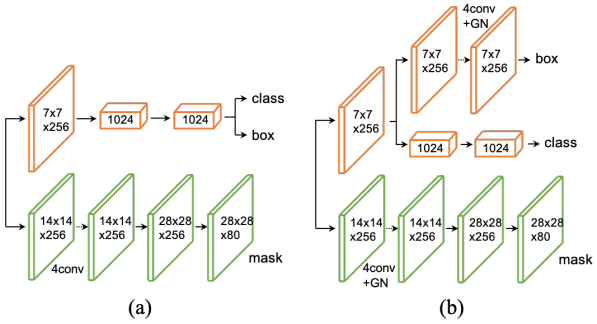
Header Structure: In original Mask RCNN and HTC, box header uses 2 Fully Connected layers (FCs) for box regression as shown in Figure 2 (a). FCs may lead to the missing of location information which is extremely important for bounding box prediction, while convolutional layers focus on the local features and keep the location information well. These properties make convolutional layers more suitable for location-related tasks, such as bounding box prediction etc. Thus, we disentangle the classification and box regression tasks through two separated headers, and meanwhile use fully convolutional layers for box regression instead of FCs. Specifically, 2 FCs are designed for classification and 4 Convs for box regression as shown in Figure 2 (b). Furthermore, we also apply Group Normalization [15] to convolutional header.

Enhanced key point module: Recently, key point based method for object detection becomes a popular topic. To further boost the performance of box, we propose an additional enhanced module with a key point based header as the fourth stage. We follow Grid RCNN [10] to generate key points for proposals and merge them into boxes. At training, box predictions of the third regression stage are selected as training samples for the key point stage as shown in Figure 3 (a). At inference, predictions of the key point stage and the third regression stage are averaged to get the final box predictions as shown in Figure 3 (b).

2.2. Cascade Mask Re-Scoring

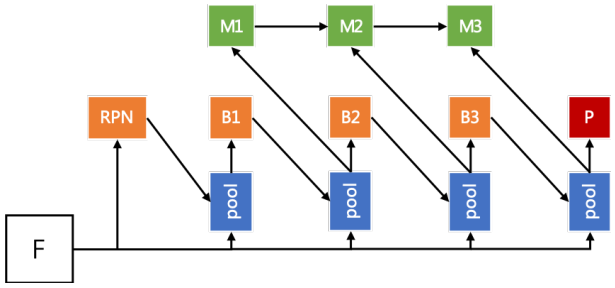
Mask Scoring RCNN [7] has demonstrated its effectiveness on instance segmentation task. The evaluation of mask

[1] *Equal contribution.

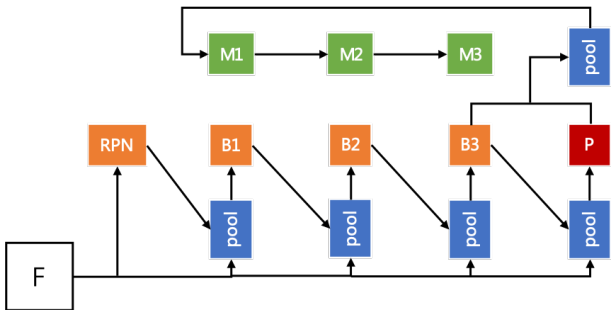


(a) (b)

Figure 2: (a) is used in Mask RCNN and HTC. (b) is the proposed header structure



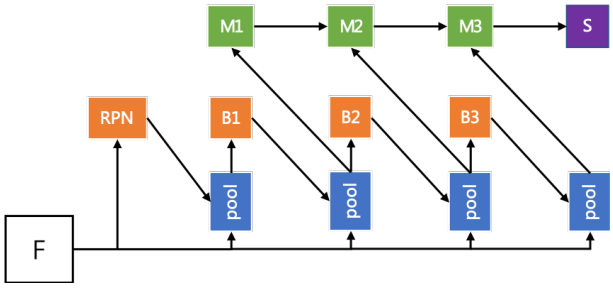
(a) Training pipeline



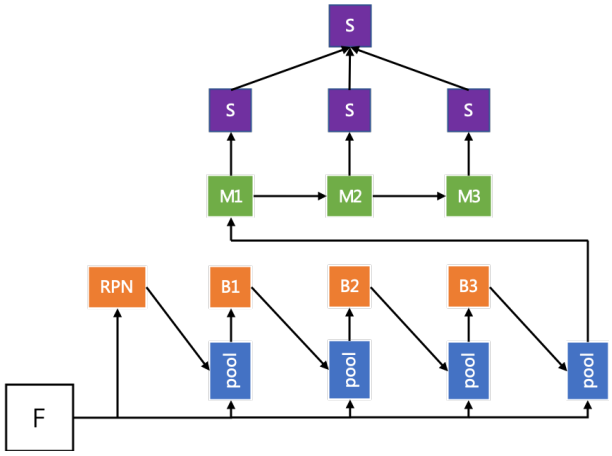
(b) Testing pipeline

Figure 3: Enhanced key point module

quality is also expected in cascade pipeline. Considering there are three stages in HTC, a straightforward way is to add mask-scoring branch at every stage of HTC header. However this is not an appropriate choice because three mask-scoring branches make the HTC header too heavy. Here we proposed Cascade Mask Re-Scoring which has only one single MaskIoU header but performs as well as the above-mentioned method. At training, we use the prediction of the third mask stage and its ROI feature as the input of this MaskIoU header as illustrated in Figure 4 (a). At inference, we calculate MaskIoU for every mask prediction from all three stages and average them as the final MaskIoU as shown in Figure 4 (b). Following Mask Scoring RCNN, we use the product of classification score and MaskIoU prediction to re-score the mask quality.



(a) Training pipeline



(b) Testing pipeline

Figure 4: Cascade Mask Re-Scoring

2.3. Refining Strategy with High-Quality Proposals

Guideded Anchoring [14] and Cascade RPN [13] demonstrate the idea that high quality proposals benefit the detection performance. Progressively strengthening the requirements through the stages is a useful principle to design the pipeline. In proposal learning, starting out with an anchor-free metric followed by anchor-based metrics in the ensuing stages improves the performance of proposals [13]. And in cascade headers, IoU threshold to determine positive examples increases in three stages. Motivated by these ideas, we find that the refining process can also be introduced to training schedule, not just in network pipeline. Thus we propose a progressive training strategy with a two stage training schedule. Based on HTC, the first training stage uses original RPN and the IoU threshold in header is (0.5, 0.6, 0.7). The second refining stage uses precomputed high quality proposals and the IoU threshold in header is (0.6, 0.7, 0.8).

2.4. Scale Aware Inference

Scale variation is one of the key challenge in instance-level task such as object detection and instance segmentation. As a solution, multi-scale training and multi-scale testing are commonly used in recent state of the art methods.

Methods	Box AP	Mask AP
Baseline	42.1	37.3
Seperated Header	43.1	38.0
+ Enhanced key point module	43.4	38.1

Table 1: Ablation studies of Enhanced Box Header

Methods	Box AP	Mask AP
Baseline	42.1	37.3
Cascade Mask Re-Scoring	42.2	38.1

Table 2: Ablation studies of Cascade Mask Re-Scoring

However, in multi-scale testing, it is non-trivial to merge different predictions from different scales. Small and large instances are hard to be predicted in extreme image resolutions. Specifically, large instances rely on low resolution and small instances rely on high resolution while medium sized instances are less affected by extreme resolution. Thus we propose scale aware inference and apply it to the prediction of proposal, box and mask. For proposal and box, we weighted average medium sized predictions from all scales based on their scores. And for small and large predictions, we collect and mix them from specific scales. Finally, we use soft-NMS [1] to combine the results. For mask, the difference is that for small and large predictions, we also conduct weighted average but ignore those from extreme resolutions.

3. Experiments

3.1. Datasets

We perform experiments on the Challenging COCO dataset with instance annotations and COCO-stuff [2] annotations. **No other dataset** is used in our final submissions. We train our models on the split of 2017 train (115k images) and report results on 2017 val and 2017 test-dev.

3.2. Implementation Details

In ablation study, we use ResNet-50 [5] as the backbone for fair comparison. We train networks with 16 GPUs (two images per GPU) for 1x schedule (12 epoches) with an initial learning rate of 0.04, and decrease it by 0.1 after 8 and 11 epoches, respectively. Input images are resized to 800 for the short axis and 1333 for the long axis for training and testing. No data augmentation except random horizontal flip is adopted during training.

In the final result, we train networks with 32 GPUs (one image per GPU) for 2x schedule (20 epoches) with an initial learning rate of 0.04, and decrease it by 0.1 after 16 and 19 epoches, respectively. Multi-scale training is adopted. At inference, six scales are selected (900x600, 1200x800, 1500x1000, 1800x1200, 2100x1400, 2400x1600) to conduct scale aware inference. Soft-NMS is applied and the top 100 detections are selected as the final result.

Methods	Box AP	Mask AP
Baseline	43.0	37.9
End-to-End (0.5,0.6,0.7)	43.3	38.0
End-to-End (0.6,0.7,0.8)	43.1	38.1
Refining Strategy	43.8	38.6

Table 3: Ablation studies of Refining Strategy

Methods	Box AP	Mask AP
Multi-scale Testing	52.0	44.9
Scale Aware Inference	52.7	45.5

Table 4: Ablation studies of Scale Aware Inference

3.3. Ablation Study

Our baseline achieves 42.1 Box AP and 38.3 Mask AP in 1x training schedule (12 epoches) and 43.0 Box AP and 37.9 Mask AP in 2x training schedule (20 epoches).

Effectiveness of Enhanced Box Header: We study how the Enhanced Box Header helps boost the performance. As shown in Table 1, the new header structure improves the Box AP by 1.0 and Mask AP by 0.7. The enhanced key point module contributes to a further 0.3 improvement of box.

Effectiveness of Cascade Mask Re-Scoring: We compare the effectiveness of Cascade Mask Re-Scoring. As shown in Table 2, it achieves remarkable improvement by 0.8 of Mask AP.

Effectiveness of Refining Strategy with High-Quality Proposals: Refining Strategy is conducted with extra 1x refining schedule. High-Quality Proposals are predicted by ResNet-50 Guided Anchoring. To investigate the effectiveness, we compare our strategy with end-to-end training in which original RPN is replaced by Guided Anchoring. We also compare different IoU threshold used in headers. These experiments are conducted in 2x schedule for fair comparison with Refining Strategy. From Table 3, we find that our refining strategy outperforms other methods.

Effectiveness of Scale Aware Inference: We perform experiment on a powerful model with ResNext-101-64x4d [16] backbone, Deformable ConvNetV2 [17] and multi-scale training. The result is illustrated in Table 4 , with an improvement of 0.6 on Mask AP compared with multi-scale testing.

3.4. Final Result

With the proposed methods, we achieve single model result of 49.4 Mask AP and ensemble model result of 51.2 Mask AP with 2.2 absolute improvement compared to the winning entry last year. Besides, Other common tricks are also applied including Deformable ConvNet V2 and Synchronized Batch Normalization [12]. For

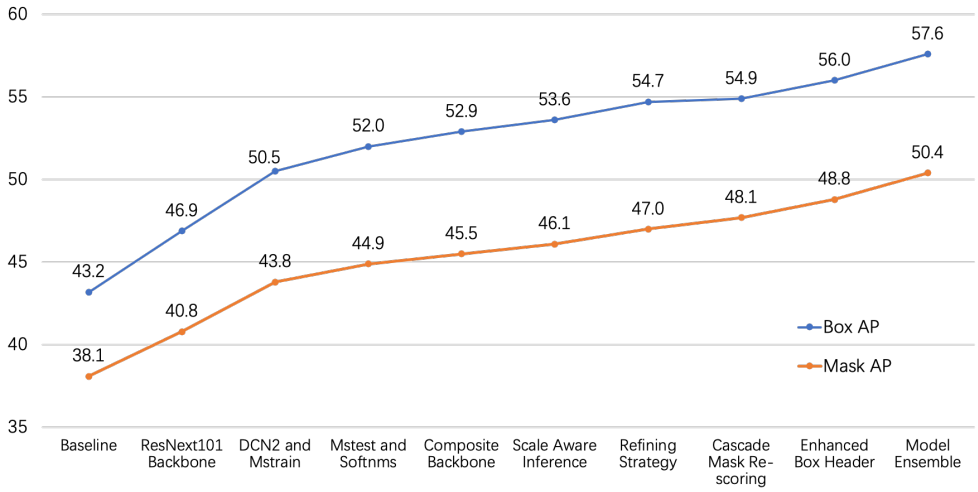


Figure 5: Details on COCO val dataset

model ensemble, we utilize six networks with different backbones: ResNeXt-101-64x4d-CBNet(composite backbone [9]), ResNeXt-101-32x8d-CBNet, SENet-154 [6], ShuffleNet-V2 [11], ResNeXt-101-64x4d and ResNeXt-101-32x8d. The details on COCO validation set can be found in Figure 5.

References

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 3

[2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, 2018. 3

[3] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4974–4983, 2019. 1

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[5] 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. 3

[6] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 4

[7] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6409–6418, 2019. 1

[8] 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 *ECCV*, 2014. 1

[9] Yudong Liu, Yongtao Wang, Siwei Wang, TingTing Liang, Qijie Zhao, Zhi Tang, and Haibin Ling. Cbnet: A novel composite backbone network architecture for object detection. *arXiv preprint arXiv:1909.03625*, 2019. 4

[10] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7363–7372, 2019. 1

[11] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 116–131, 2018. 4

[12] Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun. Megdet: A large mini-batch object detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6181–6189, 2018. 3

[13] Thang Vu, Hyunjun Jang, Trung X Pham, and Chang D Yoo. Cascade rpn: Delving into high-quality region proposal network with adaptive convolution. *arXiv preprint arXiv:1909.06720*, 2019. 2

[14] Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2965–2974, 2019. 2

[15] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 1

[16] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 3

[17] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 3