Multi-Task Learning via Scale Aware Feature Pyramid Networks and Effective Joint Head

Feng Ni
Peking University
nifeng@pku.eu.cn

Yuehan Yao
South China University of Technology
guiguzhixing@gmail.com

Abstract

As a concise and classic framework for object detection and instance segmentation, Mask R-CNN achieves promising performance in both two tasks. However, considering stronger feature representation for Mask R-CNN fashion framework, there is room for improvement from two aspects. On the one hand, performing multi-task prediction needs more credible feature extraction and multi-scale features integration to handle objects with varied scales. In this paper, we address this problem by using a novel neck module called SA-FPN (Scale Aware Feature Pyramid Networks). With the enhanced feature representations, our model can accurately detect and segment the objects of multiple scales. On the other hand, in Mask R-CNN framework, isolation between parallel detection branch and instance segmentation branch exists, causing the gap between training and testing processes. To narrow this gap, we propose a unified head module named EJ-Head (Effective Joint Head) to combine two branches into one head, not only realizing the interaction between two tasks, but also enhancing the effectiveness of multi-task learning. Comprehensive experiments show that our proposed methods bring noticeable gains for object detection and instance segmentation. In particular, our model outperforms the original Mask R-CNN by 1˜2 percent AP in both object detection and instance segmentation task on MS-COCO benchmark. [1]

1. Introduction

In the past few years, object detection and instance segmentation results were rapidly improved by the vision community for the powerful baseline system Mask R-CNN([11]), which extends Faster R-CNN([9]) by adding a branch for predicting an object mask in parallel with the existing part for bounding box recognition. The mask branch is a small FCN([20]) applied to each RoI([13]), predicting a segmentation mask in a pixel-to-pixel manner. This method is conceptually natural and offers extensibility and robustness, shows a surprisingly smooth, flexible, and fast system for instance segmentation results. All in all, Mask R-CNN is a breakthrough in the field of instance segmentation.

However, this remarkable multi-task learning method suffers from a common problem of modern detection methods. That is scale variation, since Convolutional Neural Network is sensitive to scales. And what's more, performing multi-tasks needs more credible feature extraction execution and multi-scale complementary features integration. Therefore, it is urgent to tackle this problem.

Feature pyramid is a common practice. FPN([15]) augmented a top-down path with lateral connections for object detection. It exploits the inherent multi-scale, pyramidal hierarchy of deep convolutional networks to construct feature pyramids with marginal extra cost. PANet([18]) extends FPN with an additional bottom-up path augmentation and proposes adaptive feature pooling from multi-scale features.

Rethinking the extracted multi-scale features of general FPN, we find that the top-down pathway FPN only introduces high-level semantic information to low-level feature, while ignore the role of low-level feature for localization. This inspires us to design a novel FPN called SA-FPN (Scale Aware Feature Pyramid Networks) to address the challenge of scale variation. SA-FPN, combining Top-Down style FPN and Bottom-Up style FPN, absorbs the characteristics of them and becomes a more accurate module for scale variation perception.

Another aspect for improvement of Mask R-CNN in multi-task learning is about the parallel isolated branches of Mask R-CNN. The segmentation branch of Mask R-CNN is based on the output of Region Proposal Network (RPN) in training stage, which ignores the inherently tie in those two tasks and is inconsistent with testing processes. In common sense, instance segmentation is connected detection based on the bounding box strictly, which is more meticulous than the bounding box. However, the bounding box is easy to obtain than the masking label. It is worthy of trying to explore and enhance the interrelations between object detection and

[1] Code will be available soon.

instance segmentation.

In this paper, we make a natural extension of Mask R-CNN architecture, merging the detection branch and the instance segmentation branch into single branch. This smart framework contribution of our work named EJ-Head (Effective Joint Head), including three operations: "Interleaving","Heavy feature" and "Shared Conv". EJ-Head promotes both two tasks consistently and provides an example for improving multi-task learning.

In general, we improve two aspects in Mask R-CNN by proposing SA-FPN (Scale Aware Feature Pyramid Networks) and EJ-Head (Effective Joint Head). Experimental results on the challenging COCO benchmark show that when using our proposed modules, detection and instance segmentation performances are improved by about 1.6 and 1.4 percent AP increment in box AP and mask AP, respectively.

The main contributions of our work highlighted as follows. (1) We propose SA-FPN (Scale Aware Feature Pyramid Networks), which effectively integrates multi-scale complementary feature and solves the problem of scale variation in an innovative way. (2) We slickly mix the mask branch and detection branch into one branch and introduce EJ-Head (Effective Joint Head), which can reinforce each task and also eliminate the gap between training and testing processes. (3) We propose a newly enhanced Mask R-CNN, which provides a reference and is helpful for further research on multi-task learning.

2. RELATED WORK

2.1. Deep Object Detectors

Deep learning based methods([10, 13]) have tremendously pushed forward the remarkable progress in object detection over a short period of time. Mainstream object detection frameworks roughly fall into two categories. Two-stage methods like Faster R-CNN([9]), R-FCN([6]), Mask R-CNN([11]) generate a sparse set of candidate proposals that contain all objects while filtering out the majority of negative locations in the first stage, and then classify the proposals into foreground classes or background in the second stage. Single-stage approaches, such as SSD([19]),YOLO([22]),RetinaNet([16]) directly regress to predict the bounding boxes. Detection frameworks with multiple stages like Cascade R-CNN([5]) are also popular and bring tremendous improvement for object detection. But the huge training time and resources make them not commonly used like Faster R-CNN([9]), no matter in industry or in academia.

2.2. Scale Variation

Scale variation across object instances has been treated as one of the most knotty problem in modern development of detection. To address this challenge, several approaches have been proposed. Image pyramid is an intuitive way, SNIP ([24]) and SNIPER([25]) select a specific scale for each resolution during multi-scale training. However, these methods suffer a lot from the inevitable increase of inference time. Instead of taking multiple images as input, feature pyramid method uses multi-level features of different layers. FPN([15]) is the most famous representative of this strategy. It takes the fused feature map with the highest resolution to pool features and achieves superior performance. SSD([19]), DSSD([7]) and MS-CNN([1]) perform object detection at multiple layers for objects of different scales. PANet([18]) boosts information flow in proposal-base instance segmentation framework, enhances the entire feature hierarchy with accurate localization signals in lower layers by bottom-up path augmentation, which shortens the information path between lower and top layers feature.

2.3. Instance Segmentation

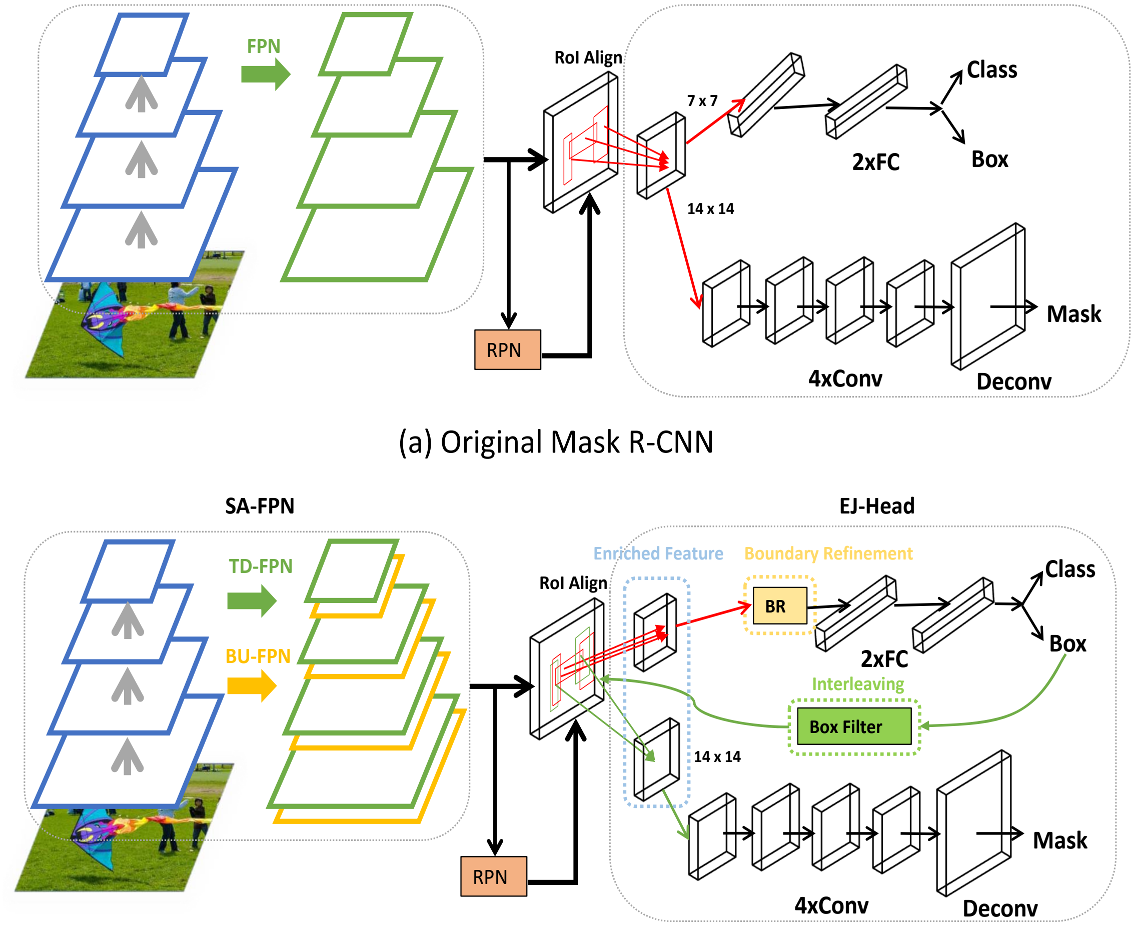
Instance segmentation is a task to predict class label and pixel-wise instance mask in an image. There are mainly two streams of methods in instance segmentation, segmentation based methods and detection based methods. The former are less commonly used nowadays. Segmentation based methods like ([26]) first predict the labels of each pixel and then identify object instances therefrom. However the performances of these methods are always unsatisfactory.

So we will mainly discuss the detection based method in this paper. Detection based methods follow a similar diagram: get the region of each instance and then predict the mask, showing a strong connection to object detection. Instance-FCN([5]) proposes instance-sensitive FCNs to generate the position-sensitive to obtain the final masks. DeepMask([8]) learns to propose segment candidates in a sliding window fashion. FCIS([14]) takes position-sensitive maps with inside/outside scores to generate the instance segmentation results, but exhibits systematic errors on overlapping instances and creates spurious edges. MaskLab([4]) produces instance-aware masks by combining semantic and direction predictions. Cascade Mask R-CNN is a multi-stage object detection and instance segmentation framework derived from Cascade R-CNN ([2]), which comprises multiple stages where the output of each stage is fed into the next one for higher quality refinement. But this multi-stage mechanism inevitably brings extra computation overhead of inference.

3. Framework

In this section, we will describe our proposed new framework for object detection and instance segmentation in details.

Overview. Based on Mask R-CNN, as shown in Figure 1, it is distinctive in two aspects: (1) It aims to remedy the



(a) Original Mask R-CNN

(b) Mask R-CNN with SA-FPN and EJ-Head

Figure 1. Architectures of the original Mask R-CNN in (a) and our proposed model in (b). Obviously, our model is different from the original model by reformimg two modules. SA-FPN (Scale Aware Feature Pyramid Networks) combines TD-FPN (Top-Down style FPN, as shown in green) and BU-FPN (Bottom-Up style FPN, as shown in yellow) together. And EJ-Head (Effective Joint Head) proposes three operations. “Enriched feature” represents enhencing the extracted RoI feature, “Boundary Refinement” means adding additional convolutions on this pathway for optimization of the boundary , “Interleaving” is an operation to filter the predicted bounding boxes which get a high IoU (Intersection-over-Union) with ground truth and then feed positive samples into the instance segmentation branch.

problem of scale variation by fusing Top-Down style FPN and Bottom-Up style FPN into a novel neck module called SA-FPN. (2) It explores the interrelations between object detection and instance segmentation, eliminating the gap between training and testing processes and enhancing both two tasks.

Overall, these changes to the framework architecture effectively improve two tasks and also provides a reference for further research on multi-task learning.

3.1. Scale Aware Feature Pyramid Networks

Scale variation across object instances is one of the most challenging problems in object detection, especially for very small or huge objects. Simply using multi-scale image pyramids has some improvement in accuracy but suffers a lot from increasing inference time. Feature pyramid is popular to deal with this knotty problem, and Top-Down style FPN in ([15]) is the most widely used feature pyramid structure.

As illustrated in Figure 2 (a), a general FPN is a Top-Down style structure that feeds the last output of the backbone into the top of feature pyramid, and the information flow direction is from shallow to deep in the backbone and from the top to bottom in feature pyramid. Up-sampling operation and lateral connection fusion are applied step by step to get a larger feature until the bottom of the pyramid. Thus,

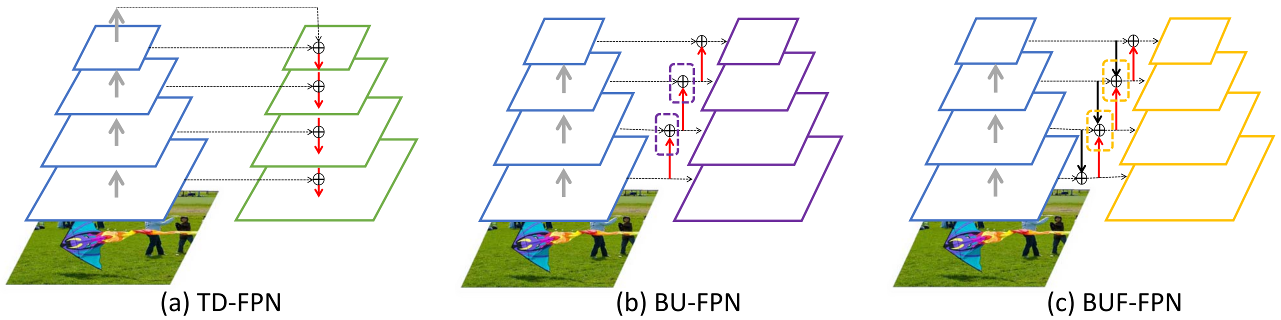


Figure 2. Three different architectures of FPNs. (a) is the most widely used Top-Down style FPN, which is used in the baseline model. (b) and (c) are two newly designed Bottom-Up style FPNs. The main difference is that in (b) each layer of feature pyramid combines with only neighboring shallower layer, while in (c) both neighboring shallower and deeper features are fused. So we call (c) Bottom-Up-Fusion FPN (BUF-FPN). Details are shown in purple dotted boxes in (b) and in orange dotted boxes in (c). Experiments in Table 3 show that BU-F FPN performs better. And when we concatenate (a) and (c) together and add 1x1 convolutions to reduce dimensions, we propose SA-FPN (Scale Aware FPN), as illustrated in Figure 1 (a).

deep feature map with high semantic information and shallow feature map with high resolution are combined. This greatly boosts the performance on small objects detection and therefore improves the overall detection performance.

However, the top-down pathway of general FPN only introduces high-level semantic information to low-level feature, but ignore the subsidiary role of low-level feature like edges and textures which also are important for accurately localizing instances. Most of the improvements caused by Top-Down FPN come from more accurate detection on small objects. Shallow features with high resolution contain useful low-level features. Different layers are of different resolutions, the fusion operation between neighbor shallower or deeper feature is also important.

Following the discussion above, we design two new Bottom-Up style FPNs, as illustrated in Figure 2 (b) BU-FPN and (c) BUF-FPN. Experiments in Table 3 show that (c) BUF-FPN performs better. The main difference between (b) and (c) is the feature integration operation, Details are shown in purple dotted boxes in (b) and in orange dotted boxes in (c). In (b) each layer of feature pyramid combines with only neighboring shallower layer, while in (c) both neighboring shallower and deeper layers. So we use (c) BUF-FPN as the default Bottom-Up style FPN for further experiments.

BU-FPN takes every level output of the backbone as the input of the next step and applies down-sampling and fusion operation step by step to build a feature pyramid from bottom to top. Letting low-level information guide high-level semantic information, BU-FPN is especially better at accurate detection of large objects.

When we combine TD-FPN (Top-Down FPN) and BUF-FPN (Bottom-Up-Fusion FPN) together, a scale sensitive neck module called SA-FPN (Scale Aware FPN) is proposed. Experiments in Table 3 show that concatenating performs better then element-wise addition. Then we add 1x1 convolutions to reduce dimensions. In line with expectations, SA-FPN absorbs the advantages of two style FPNs and improves both detection and instance segmentation in different object scales.

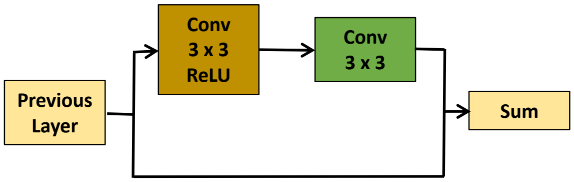


Figure 3. Detail of Boundary Refinement(BR) module in EJ-Head. There are two branches, one is two stacked convolutions, the other is a shortcut. Finally two branches are added at the pixel level, which brings the boundary alignment effect.

3.2. Effective Joint Head.

For multi-task learning, parallel execution of tasks is a common practice. However, instance segmentation in Mask R-CNN is such a detection based method that relies heavily on detection. Thus, the interaction between these two tasks is worth further exploring.

Firstly, we simply connect the segmentation branch behind the detection head in series when training, feeding the bounding box predictions of the detection into the mask head. As shown in Figure 1 (b), green dotted boxes “Interleaving” represents this execution. “Interleaving” is an operation to filter predicted bounding box which have IoU(Intersection over Union) with ground truth boxes of at least 0.5 and feed positive samples of mask into instance segmentation branch. In this way, the segmentation branch can take advantage of the detection process, realizing the interaction between two tasks. Moreover, the training and testing pipelines are highly consistent. And experiments in

Table 1. Comparison with state-of-the-art methods on COCO *test-dev* dataset. Note that Our main research is about single-stage methods. Multi-stage mechanism like Cascade Mask R-CNN inevitably brings extra computation overhead of inference.

Method	Backbone	box AP	AP$_{50}$	AP$_{75}$	mask AP	AP$_{50}$	AP$_{75}$
Faster R-CNN + RoI Align	ResNet101-FPN	38.0	58.7	40.6	-	-	-
FCIS	ResNet101	-	-	-	29.2	49.5	-
FCIS+	ResNet101	-	-	-	33.6	54.5	-
MaskLab	ResNet101	39.6	60.2	43.3	35.4	57.4	37.4
MaskLab+	ResNet101	41.9	62.6	46.0	37.3	59.8	39.6
Mask R-CNN	ResNet101-FPN	40.9	62.3	44.3	37.0	59.1	39.4
PANet	ResNet101-FPN	**42.8**	64.0	**46.4**	38.0	60.5	40.4
Ours	ResNet101-FPN	42.5	**64.0**	46.1	**38.4**	**60.6**	**40.9**
Cascade Mask R-CNN	ResNet101-FPN	44.4	65.6	48.3	**38.4**	60.2	**41.4**

Table 4 prove this operation indeed improved performance.

Merging heavy mask branch into box branch may slightly increase the burden of box branch. We enlarge the extracted RoI features from 7 x 7 to 14 x 14, and name it “Enriched Feature” as shown in blue dotted boxes in Figure 1 (b). The quality of RoI features is enhanced by the enlarged receptive field. Richer features further help enhance both the detection and instance segmentation. Experiments in Table 4 show the effectiveness of our operation.

The whole flow of information goes through an unified path. Despite increased resolution, we can add additional convolutions on this pathway to further boost localization especially for details and edges of objects. As shown in Figure 3 the yellow dotted boxes “Boundary Refinement” means this execution. Boundary Refinement module is a residual structure for boundary alignment. Experiments in Table 5 show that “Boundary Refinement” is more superior than directly two stacked 3 x 3 simple convolutions together.

4. Experiments

4.1. Datasets and Evaluation Metrics

Datasets Description. We perform all experiments on MS-COCO([17]), which is the most representative and challenging dataset for object detection as well as instance segmentation. Following COCO-2017 settings, we train our models on 2017*train* (115k images) and present experimental results on 2017*val* and 2017*test-dev*.

Evaluation Metrics. We report the standard COCO-style evaluation metric AP (averaged over IoU thresholds) on two tasks, including AP$_{50}$, AP$_{75}$ (AP at different IoU thresholds) and AP$_S$, AP$_M$, AP$_L$ (AP at different scales). Both box AP and mask AP are evaluated.

4.2. Implementation Details

In all experiments, we use ResNet([12]) as backbones. In experiments based on ResNet-50, we use 8 NVIDIA TITAN Xp GPUs (2 images per GPU). To ensure the consistency of the overall batch size for heavier backbone network ResNet-101, we utilize 16 GPUs (one image per GPU) in our experiments. The input shape of images are resized to 1333 and 800 for training and testing. We train all models with PyTorch([21]) and mmdetection([3]). We adopt different training strategies to save the training time. Specifically, unless noted, for ablation studies, we train for 12 epochs with an initial learning rate of 0.02, and decrease it by 0.1 after 9 and 11 epochs. While for the models used to compare with state-of-the-art methods, we train for 20 epochs with the same learning rate, and decrease it by 0.1 after 16 and 19 epochs.

4.3. Benchmarking Results

We compare our model with the state-of-the-art object detection and instance segmentation approaches on the COCO dataset. In Table 1, we can see that our method exhibits substantial improvements compared to the single task (*i.e.*, object detection or instance segmentation) methods. In particular, our model is 4.5% absolutely better than Faster R-CNN([23]) (with RoI Align) in the criterion of box AP. Our method also outperforms FCIS([14]) model by 3.0% in mask AP.

We then conduct the comparisons with multi-task learning methods that jointly predict object bounding boxes and segmentation masks. Notably, comparing with Mask R-CNN, our model reports 1.6% and 1.4% gains in terms of box AP and mask AP, respectively. Obviously, our model performs better than MaskLab+([4]). Compared with PANet([18]) which is the state-of-the-art single-stage based method, our model performs even a little better. Although the proposed model performs slightly worse than Cascaded Mask R-CNN, the multi-stage method consumes more computing resources and time.

4.4. Ablation Study

Component-wise Analysis. To evaluate the effectiveness and generalization ability of two main components (*i.e.*, SA-FPN and or EJ-Head), we conduct comparative experiments on different backbones and methods. From Table 2,

Table 2. Effects of each component in our design. Results are reported on COCO 2017*val*.

Model	Backbone	box AP	mask AP
Mask R-CNN (Baseline)	ResNet50-FPN	37.2	34.1
Mask R-CNN + SA-FPN	ResNet50-FPN	37.8	34.6
Mask R-CNN + EJ-Head	ResNet50-FPN	38.2	34.8
Mask R-CNN + SA-FPN + EJ-Head (Ours)	ResNet50-FPN	**38.9**	**35.5**
Mask R-CNN	ResNet101-FPN	39.4	35.9
Mask R-CNN + SA-FPN + EJ-Head (Ours)	ResNet101-FPN	**41.0**	**37.3**
Cascade Mask R-CNN	ResNet50-FPN	41.3	35.7
Cascade Mask R-CNN + SA-FPN + EJ-Head	ResNet50-FPN	**42.8**	**36.9**

Table 3. Ablation study of FPN designs on COCO 2017*val*. BU-FPN (shown in Figure 2 (b)) means using the fusion operation in Bottom-Up style FPN to compensate shallow feature by fusing only shallower neighboring feature. While BUF-FPN (shown in Figure 2 (c)) fuse both higher and lower level neighboring feature. AP$_S$, AP$_M$, AP$_L$ means AP at small, middle and large scales. Note that in the following discussion, we treat BUF-FPN as the default Buttom-Up style FPN.

FPN design	box AP	AP$_S$	AP$_M$	AP$_L$	mask AP	AP$_S$	AP$_M$	AP$_L$
TD-FPN	37.2	33.8	56.7	65.0	34.1	28.0	52.4	62.8
BU-FPN	34.6	30.0	53.0	65.2	31.8	24.8	49.0	63.0
BUF-FPN	36.8	32.0	55.9	**66.1**	33.7	26.6	50.9	**63.9**
TD-FPN add BUF-FPN	37.0	32.8	56.3	65.2	33.9	27.1	51.3	63.0
SA-FPN (TD-FPN concat BUF-FPN)	**37.9**	**34.5**	**57.4**	65.6	**34.8**	**29.4**	**52.7**	63.3

we can see that the SA-FPN module improves the box AP and mask AP by 0.6% and 0.5% respectively, compared to the Mask R-CNN methods based on ResNet-50-FPN backbone. Accordingly, the EJ-Head contributes to 1.0% and 0.7% improvement under the same settings. The combination of SA-FPN and EJ-Head modules brings significant improvements by 1.7% and 1.4% respectively, demonstrating that these two modules work in a complementary manner. Besides, consistent improvement is achieved based on the ResNet-101 backbone. It is also worthy to note that the proposed SA-FPN and EJ-Head modules also provide considerable improvements for multi-stage method like Cascaded Mask R-CNN, showing that our method can serve as plugin units for existing methods.

SA-FPN Design. In this paper, SA-FPN module is proposed to address the problem of scale variation. As shown in Table 3, TD-FPN performs better for small objects while BU-FPN and BUF-FPN are better for large objects. BUF-FPN fuses neighboring feature of both higher and lower level performs better than BU-FPN which only exploits neighboring feature from lower layer. And concatenating performs better then element-wise addition, so we add 1x1 convolutions to reduce dimensions. This demonstrating the bi-directional fusion manner obtains better feature representation in Bottom-Up style FPN. Our SA-FPN module combines Top-Down style FPN (TD-FPN) and Buttom-Up style FPN (BUF-FPN) and obtains 0.7% and 0.6% improvements respectively, indicating that the combination facilitates the handling of objects with varied scales.

EJ-Head Design. Next we investigate the effect of key ingredients of EJ-Head including "Enriched feature", "Boundary Refinement" and "Interleaving". As shown in Table 4, each component performs better than the Mask R-CNN. In particular, "Boundary Refinement" module brings the most significant improvement over baseline model by 0.6% for box AP, indicating that the quality of RoI is enhanced by the enlarged receptive field. In addition, "Interleaving" with "Enriched Feature" exhibits the superior improvements over 0.6% for mask AP, which verifies the effectiveness of the proposed modules. Overall, EJ-Head achieves 1.1% and 0.7% on box AP and mask AP, respectively.

In Table 5 we study the various designs of "Boundary Refinement" module in EJ-Head. The proposed Boundary Refinement module performs better than directly stacked convolutions. The residual structure plays an important role for boundary alignment. Stacked convolutions also have a little bit boost, but not very obvious as Boundary Refinement. When stacked more convolutions, it even will harm the performance of mask prediction.

Qualitative results of multi-task learning results are illustrated in Figure 4. These results are based on ResNet-101-FPN, achieving a box AP of 42.5 and mask AP of 38.4. Masks are shown in color, bounding boxes and categories are also shown. It can be clearly seen that people and cars of different scales achieve accurate detection and robust instance segmentation, which shows the effectiveness of our methods. Moreover, in real-world scenarios such as autonomous driving, video surveillance and even in the wild, small objects are well parsed and sensed, and large objects have clear contour boundaries and detailed information.

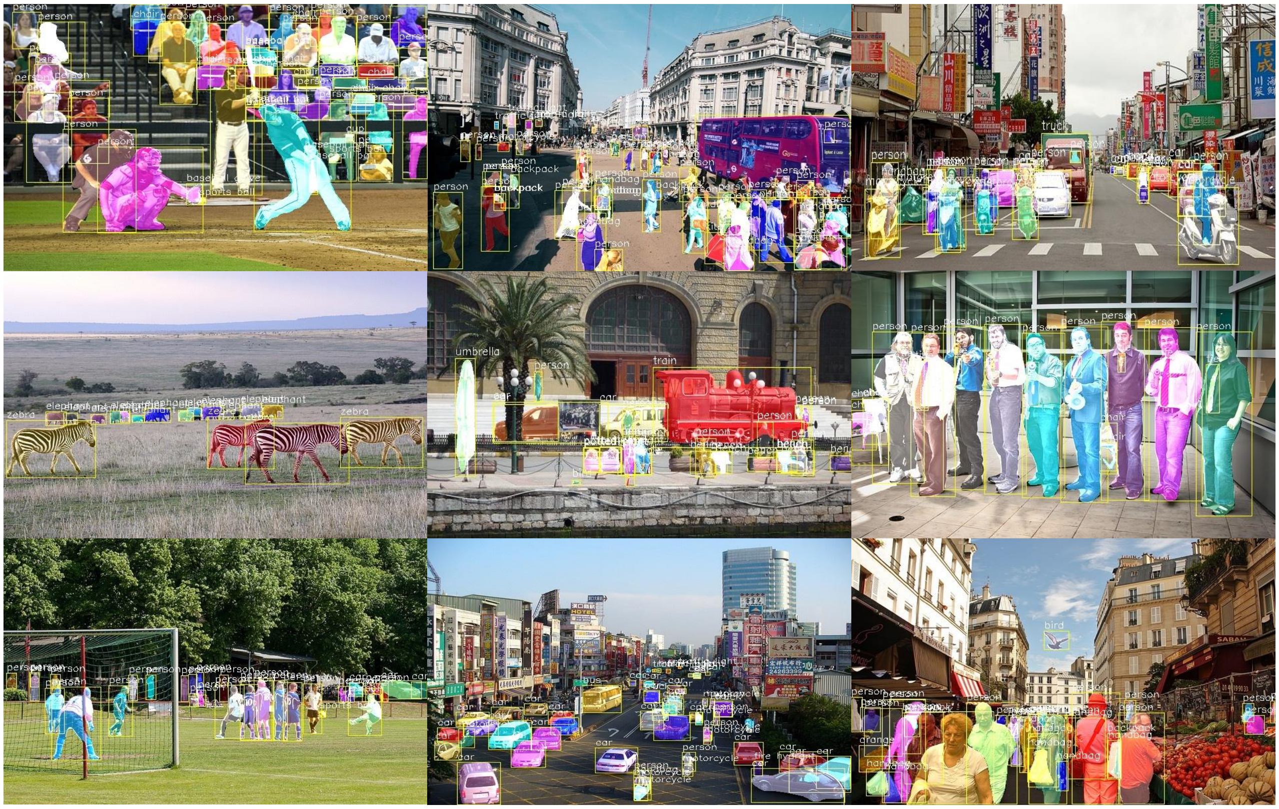


Figure 4. Examples of multi-task learning results on COCO 2017*test-dev*. Predicted detection results are shown in yellow bounding boxes, masks are also shown.

Table 4. Ablation study of EJ-Head on COCO 2017*val*. The original Mask R-CNN uses none of three operations as shown in the first row in the table.

Method	Interleaving	Enriched Feature	Boundary Refinement	box AP	mask AP
Mask R-CNN(Baseline)				37.2	34.1
Mask R-CNN	✓			37.2	34.5
Mask R-CNN		✓		37.6	34.4
Mask R-CNN			✓	37.8	34.2
Mask R-CNN	✓	✓		37.7	34.7
Mask R-CNN + EJ-Head	✓	✓	✓	**38.3**	**34.8**

Table 5. Ablation study of the design of "Boundary Refinement" on COCO-2017*val*. "Simple Conv" means using common 3 x 3 stacked convolutions. "Simple Conv x 2 + Shortcut" is shown in Figure 3.

Boundary Refinement(BR)	box AP	mask AP
None	37.2	34.1
Simple Conv x 1	37.4	34.1
Simple Conv x 2	37.6	34.1
Simple Conv x 3	37.6	34.0
Simple Conv x 2 + Shortcut	**37.8**	**34.2**

5. Conclusions and Future Work

In this paper we improve Mask R-CNN by proposing SA-FPN and EJ-Head for multi-task learning. This framework progressively integrates complementary multi-scale features in SA-FPN and enhances both detection and instance segmentation tasks with EJ-Head. It shows an innovative way to remedy scale variation issue, and also interweaves detection and segmentation branches for multi-task learning. Without bells and whistles, our overall system obtains remarkable improvements on COCO test-set, achieving 42.5 box AP and 38.4 mask AP. We hope our simple and effective approach will serve as a new baseline and contribute to both object detection and instance segmentation.

References

[1] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European conference on computer vision*, pages 354–370. Springer, 2016.

[2] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018.

[3] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. mmdetection. `https://github.com/open-mmlab/mmdetection`, 2018.

[4] L.-C. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, and H. Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4013–4022, 2018.

[5] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158, 2016.

[6] 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.

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

[8] J. Gao, B. Wang, Z. Lin, W. Xu, and Y. Qi. Deepmask: Masking dnn models for robustness against adversarial samples. 2017.

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

[10] 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.

[11] 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.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. 2015.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[14] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2359–2367, 2017.

[15] 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.

[16] 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.

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[18] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018.

[19] 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.

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[21] A. Paszke, S. Gross, S. Chintala, and G. Chanan. Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. *PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration*, 6, 2017.

[22] J. Redmon and A. Farhadi. Yolo: Real-time object detection. *Pjreddie. com*, 2016.

[23] 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.

[24] 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.

[25] B. Singh, M. Najibi, and L. S. Davis. Sniper: Efficient multi-scale training. In *Advances in Neural Information Processing Systems*, pages 9310–9320, 2018.

[26] Z. Zhang, S. Fidler, and R. Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *Computer Vision Pattern Recognition*, 2016.