MaskPlus: Improving Mask Generation for Instance Segmentation

Shichao Xu Shuyue Lan Qi Zhu
Northwestern University, USA
{shichaoxu2023@u., shuyuelan2018@u., qzhu@}northwestern.edu

Abstract

Instance segmentation is a promising yet challenging topic in computer vision. Recent approaches such as Mask R-CNN typically divide this problem into two parts – a detection component and a mask generation branch, and mostly focus on the improvement of the detection part. In this paper, we present an approach that extends Mask R-CNN with five novel techniques for improving the mask generation branch and reducing the conflicts between the mask branch and the detection component in training. These five techniques are independent to each other and can be flexibly utilized in building various instance segmentation architectures for increasing the overall accuracy. We demonstrate the effectiveness of our approach with tests on the COCO dataset.

1. Introduction

Instance segmentation is a promising and challenging task in vision, with potential applications in medical imaging [5, 6], autonomous vehicles [1, 33], smart city [29], robotics [30, 3], etc. The problem has received significant interests in recent years. It can be viewed as a more complex case than semantic segmentation, as we not only need to segment and classify the objects, but also should identify each individual instance.

In the literature, researchers have proposed a number of approaches for instance segmentation. One popular idea is to cluster the similar contents in the image. For instance, Bert et al. [12] explored an approach of using convolutional neural networks (CNNs) to produce a representation that can be easily clustered into instances. Alireza et al. [13] proposed a fully convolutional embedding model to segment the instances by computing the likelihood of two pixels belonging to the same object. Alejandro et al.'s work in [25] also received significant interests. It introduced a new algorithm named associative embedding, which can teach networks to output joint detection and group assignments in a single stage. Similar idea also appeared in [31, 17, 37, 32].

One of the most successful object detection methods is Faster R-CNN [27], which extended the work in fast R-CNN [14] by adding the region proposal network (RPN) to speed up the region proposal process. This approach has also been applied to instance segmentation and led to a number of proposal-based methods. For instance, Dai et al. [11] fused the detection and classification steps with a cascade network, and obtained the top result in the 2015 MS-COCO instance segmentation challenge. Xu et al. [35] extracted regional, location and boundary features from gland histology images and created a CNN to identify the object individuals. Later, Li et al. [18] adopted the idea from InstanceFCN [11] and used position-sensitive score map to perform object segmentation and detection at the same time.

Recently, Mask R-CNN [15] and its extensions such as PANet [23] take advantage of the Faster R-CNN detection framework for the further improvement of instance segmentation. According to them, an instance segmentation task can be viewed as the combination of a detection problem and a segmentation problem. Their solutions first apply a detection component and then a mask generation branch, where segmentation is performed based on the Region of Interest (RoI) features. Such approaches avoid the problem of spurious edges in the FCIS method [18], and also eliminate the misalignment in the RoI-pooling process and get exact spatial locations. They represent the state-of-the-art results on instance segmentation.

However, most of these works focus on improving the detection component, while there are still *significant limitations in the mask branch that require further improvements*: 1) the mask branch can only get the features from the RoI and may suffer from the loss of global semantic information; 2) imperfect bounding boxes degrade the overall performance in the mask branch; 3) simple mask branch architecture with one deconvolutional layer and the lack of boundary refinement lead to coarse results; and 4) conflicts in multitask training (due to different learning pace of each part) may cause performance degradation.

To address these limitations, we developed a new instance segmentation framework **MaskPlus** in this work, which extends Mask R-CNN with five techniques to boost

the performance of mask generation: 1) contextual fusion, 2) deconvolutional pyramid module, 3) boundary refinement, 4) quasi-multitask learning, and 5) biased training. More specifically, the contributions of this work include:

- We create novel techniques – contextual fusion, quasi-multitask learning and biased training to incorporate global information into the mask generation branch, get better supervised mask training and reduce the conflicts that happen in multitask training.
- We further extend the existing techniques, including boundary refinement, deconvolutional pyramid module, to improve the overall performance and get finer mask results.
- We test our approach on the COCO instance segmentation dataset [22] and show our competitive results on CodaLab COCO leaderboard. We conduct ablation studies to illustrate the efficacy of each technique, and also evaluate the overall instance segmentation performance. It is demonstrated that our MaskPlus is effective and achieves the state-of-the-art results.

The rest of the paper is organized as follows. Section 2 introduces related works in more details. Section 3 presents our proposed MaskPlus framework, with details for each of the five techniques. Section 4 shows the experimental results of our ablation studies and overall evaluation. Section 5 concludes the paper.

2. Related Work

2.1. Instance Segmentation

One common approach for instance segmentation is clustering, which gathers the similar pixels to form the instances. Liang et al. [19] used proposal free network to generate the coordinates of the instance bounding box and the confidence scores of different categories for each pixel, and then added clustering as the post-processing module to generate the instance results. Bert et al. [12] presented an approach to produce a representation from CNNs that can be easily clustered into instances by applying the mean-shift algorithm to obtain cluster centers. Alireza et al. [13] learned a similarity metric by creating a deep embedding model and grouped similar pixels together. Similar ideas can also be found in [25, 31, 17, 37, 32].

Another type of approach leverages the success from object detection methods such as the Faster R-CNN model [27] and its region proposal network. Dai et al. [11] won 2015 MS-COCO instance segmentation challenge by building a cascade network and connected the steps of detection and segmentation. Xu et al. [35] split the entire instance segmentation task into sub-tasks and generated regional, location and boundary features from gland histology images to classify the objects. Li et al. [18] applied RoIs onto the position-sensitive score map to address instance segmentation. Mask R-CNN [15] presented one of the most promising methods in recent years. It is built on the Faster R-CNN framework and adds an FCN backend after the RoIs as the mask generation branch. It also solves the problem of misalignment in the RoI-pooling process with a RoIAlign layer. Later, Liu et al. [23] extended the Mask R-CNN model by improving the backbone networks, adding bottom-up path augmentation and adding fully-connected paths from the features of RoIs to the features after deconvolutional layer. These extensions mostly focus on the detection component and do not address the limitations in the mask generation branch, which is the focus of our paper.

2.2. Semantic Segmentation

For the related problem of semantic segmentation, deep learning methods have been widely used. In Long et al.'s FCN model [24], end-to-end algorithm was introduced and deconvolution was utilized for up-sampling. Later, Badrinarayanan et al. [2] improved the method by recording the position information during pooling and applied it in the up-sampling process. U-Net [28] enhanced the information delivery from earlier layers to the higher layers with specially designed U-shape network framework. Yu et al. [36] applied the dilated convolutions for semantic segmentation. Chen et al. [9] adopted a similar idea, and used atrous spatial pyramid pooling (ASPP) and fully connected CRF to make model more accurate. Other works include RefineNet [20], PSPNet [38], Large Kernel Matter [26], DeepLab v3 [10], etc.

3. Proposed MaskPlus Framework

In this section, we introduce the proposed MaskPlus Framework, and explain the details of the five techniques and how they address the limitations of previous mask generation methods in instance segmentation.

Figure 1 shows an overview of MaskPlus, which extends the Mask R-CNN framework. First, a Faster R-CNN model with FPN structure is applied as the backbone. It has a branch with two detection related outputs - a classification output and a bounding box output. The RoIAlign technique is used to replace the original RoI-pooling process to give the pixel-to-pixel alignment for the results. Then the mask generation branch is applied to the output of RoI features to generate the mask output. The novelty of MaskPlus resides on the five techniques for improving the mask generation branch, which are introduced below.

3.1. Contextual Fusion

In the mask branch of Mask R-CNN, the features are only generated from the RoIs. We believe that this could limit the generation of mask prediction because of the lacking of contextual information: First, the semantic informa-

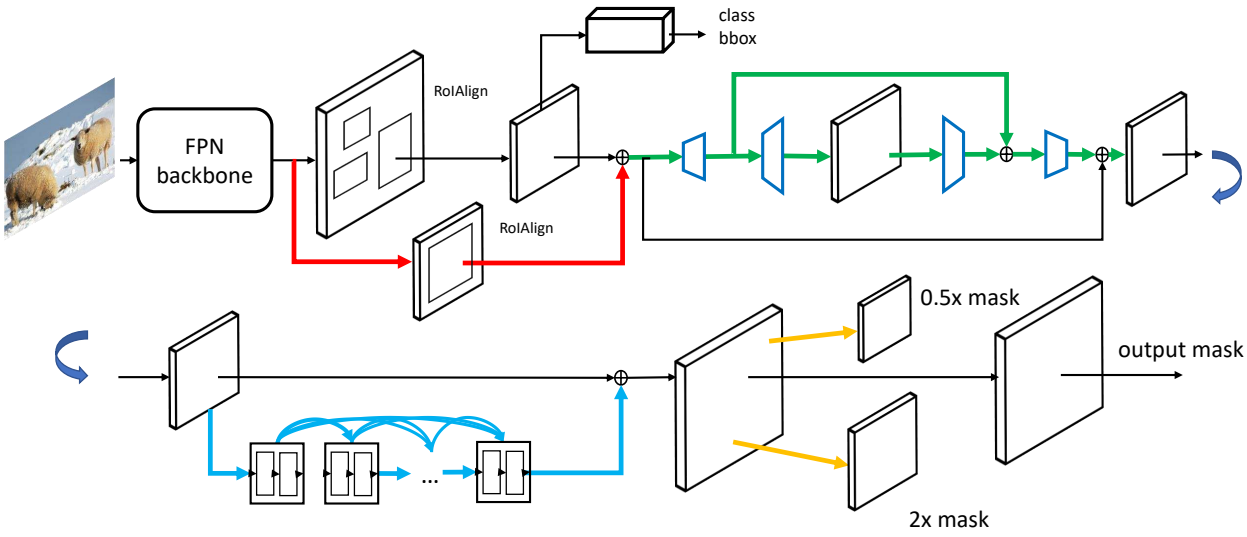

Figure 1. **Overview of the MaskPlus Framework.** Given an input image, MaskPlus outputs a generated segmentation mask. It extends the Mask R-CNN framework with various techniques on mask generation, including contextual fusion (in red), deconvolutional pyramid module (in green), improved boundary refinement (in blue), quasi-multitask learning (in yellow), and biased training (not shown in the figure). The figure is best viewed in color.

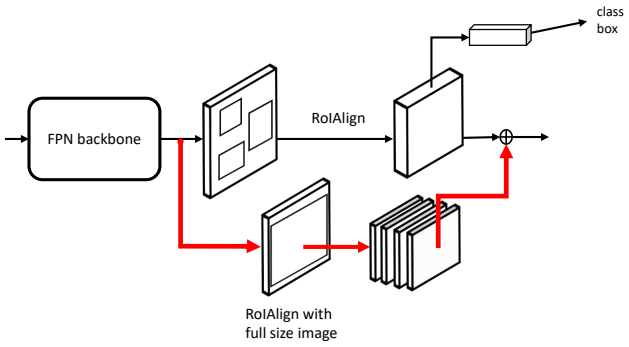

Figure 2. Contextual fusion.

tion of one object could also contain other objects, as they may have spatial relation, semantic relation, etc. For instance, a car may exist on a road but not in the sky (not yet), and thus the road could have some part of the information of car objects. It may not be effective to separate the objects and segment them one by one. Second, the RoIs are not always perfect. They may not contain some parts of the target object, which could make it difficult to segment these fragmentary objects. Third, some parts of the object that do not belong to the RoI's category may exist at the boundary. They may be mis-classified and disturb the mask generation, as they also lack their own semantic meaning and the neural networks cannot recognize them.

To address these challenges, we create a new branch from the features just before the RoIAlign module, as shown with the red lines in Figure 2. In the fusion procedure of global features, we take FPN last-layer features and apply only one full-image-size proposal at a newly-created RoIAlign layer. Then through 3 conv layers (kernel size is 3, stride is 1, filter number are 512, 256, 256, respectively), the outcomes on this new branch (the red-line path in Figure 1) will be added to the RoIAlign features from the original mask branch. The newly created RoIAlign layer and the old one have the same configuration and size of feature outputs. Such fusion helps the RoI features to get more contextual information.

Note that our approach is very different from the Fully-connected Fusion technique in [23].In [23], the input features forwarded to the up-sampling layer contain two concatenated parts – the output features from the ROI Align layer ($f1$) and the output features of the Fully-connected Fusion layer ($f2$). However, the input of the Fully-connected Fusion layer is just $f1$, which is a set of features of ROIs. It has lost the global spatial relationship between these ROIs, and the spatial information it contained is limited within individual proposals. In our approach, the input features are full-size features before the ROI Align layer, which contain the global spatial relationship of each object. Besides, the motivation and used methods are also different in the two approaches. [23] aims to utilize the strong points of fully connected layer, which does not exist in our module.

3.2. Deconvolutional Pyramid Module

Motivated by the structure of Feature Pyramid Networks(FPN) [21], which builds a pyramid module to fuse multi-level features in the early stages of the network, we define a deconvolutional pyramid module as a set of deconvolutional layers (stride = 2) followed by the equal numbers of convolutional layers (stride = 2), as shown with the green lines in Figure 1. Our design is different from the FPN though – instead of first applying down-sampling and then up-sampling, our module up-samples first and then down-samples, as shown with more details in Figure 3.

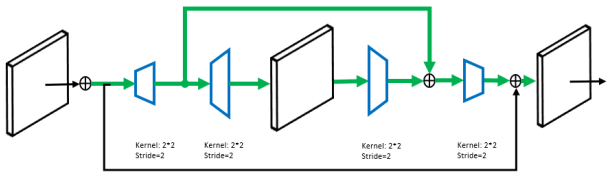

Figure 3. Deconvolutional pyramid module.

We believe that, instead of just delivering the original features to a single up-sampling layer, our module can fine-tune the existing features and combine multi-level semantic meanings together to generate better mask prediction. We observe that adding this module can improve the mask accuracy among all sizes (S, M, L) in our experiments (details in Section 4).

3.3. Improved Boundary Refinement

In the mask generation of instance segmentation, we often observe blurring boundaries – as the larger scores in the feature map mainly focus on the center part of the objects score map rather than staying at the boundary, it is often difficult to clearly identify the mask boundary. To address this challenge, we propose to learn the boundary by adding another branch, as shown with the blue lines in Figure 1 and detailed in Figure 4.

Our approach is inspired by the work from Peng et al. [26], which uses a residual block to sharpen the boundary. The difference is that we think it is insufficient to just learn the boundary with only two convolutional layers that act as a single residual block. Instead, we create a branch that consists of several convolutional modules with dense connections for better learning ability to refine the boundary information. In our experiments, we demonstrate the effectiveness of this improved boundary refinement by comparing its segmentation performance with both the original boundary refinement in [26] and the model without boundary refinement.

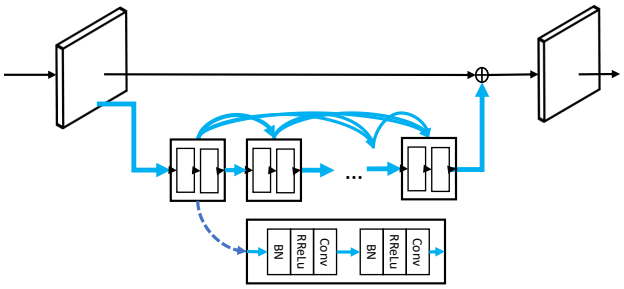

Figure 4. Improved boundary refinement.

3.4. Quasi-multitask Learning

Deep convolutional neural networks are thought to be strongly rotation invariant and scale invariant, however using them may still be insufficient to provide the desired robustness. Thus, researchers have proposed image-augmentation techniques (that include pre-processing for image rotations and rescaling to multiple scales) and feature rescaling and extracting methods such as FPN [21].

In this work, instead of learning the features from different scales of images (augmentation at the top of the network) or creating a feature pyramid architecture (augmentation in the middle of a network), we consider how the different scales labels can be a guide for helping the networks to enhance scale invariant ability (we call it – augmentation at the end of the network) while not affecting original networks' output scale/architecture/capacity.

In the past papers, most segmentation models used smaller size mask labels is because of the limitation of computational resources. Large mask labels can cost much while decreasing the mask size leads to performance lose, and researchers need to make a balance between the resource consumption and performance.

We think in another direction and develop a quasi-multitask learning approach, aka quasi-multitask as we perform similar tasks, to increase the robustness of our framework, as shown with the yellow lines in Figure 1.

More specifically, we tested the effect of combined training on the original size of mask (resized to 28 * 28), with the 0.5x size of mask (14 * 14), or the 2x size of mask(56 * 56). These branches are parallel to each other and inserted after ROIAlign features (See Section 4.7 for more details). And the most important thing is that – regardless of what other scales we add, we never use these different scale branches as the output of the final mask. That is to say, the parameters or FLOPs never increase in test stage and the output scale is never changed, while the performance increases. They are similar to a scaffolding which will be removed after the training finished. To the best of our knowledge, we are the first to explore this approach.

3.5. Biased Training

The original training strategy in Mask R-CNN trains the detection component and the mask generation branch to-

gether. In the Faster R-CNN's architecture, the RPN could be trained in advance, however it still does not solve the problem that the mask branch may not get a good RoI feature and thus provide ineffective feedback to the early stages of the training process. In some cases, the feedback from the mask branch may even disturb the training of the detection component, and then the disturbed detection results will have a negative impact on mask branch itself in the later stages.

In this work, we try to lower the influence of the training of the mask branch in the early stages, while still keeping the end-to-end learning pattern. First, we multiply the loss in the mask branch with a weight greater than 1, and define the multitask loss on each sampled RoI as $L = L_{cls} + L_{box} + \alpha(L_{mask})$. L_{cls} and L_{box} are the classification loss and bounding-box loss, respectively, as defined in [14]. L_{mask} is the mask loss as defined in [15]. The parameter α is initially larger than 1 (e.g., chosen as 1.5 in our experiments).

Intuitively, increasing one part of the loss seems to address it and makes it better. But in our design, it happens in opposite direction. The novelty that should be addressed is that using such loss function has the same effect as increasing the learning rate of the mask branch, which will force the mask branch to converge faster in the early stages. In this way, the potential negative influence of the mask branch at the early stages can be largely reduced. Then, α is set to 1 during the normal training process in the later stages. This is because this technique also causes worse mask result in the long run. Thus we need another stage of normal training to mitigate the gap.

4. Experimental Results

4.1. Dataset

All of our experiments are performed on the challenging COCO dataset [22], which is also used by Mask R-CNN. The reason why we choose COCO dataset is that it is currently the most popular and general segmentation dataset which contains a huge amount of images with different scales. The dataset has 115k training images and 5k validation images on 80 object categories. It also contains 41k test images for online testing, whose ground-truth labels are not publicly available. Our framework is trained on the train-2017 subset and perform the ablation study on the val-2017 subset. The standard COCO metrics includes AP (averaged over IoU thresholds), AP_{50}, AP_{75}, and AP_S, AP_M, AP_L (AP for images at different scales: small, medium, large). The following experiments are evaluated using mask IoU, unless specifically mentioned as detection results (AP^{bb}).

4.2. Training Configuration

Our implementation is based on the Tensorpack framework [34]. We used the re-implemented version of Mask R-CNN in Tensorpack as the baseline, which shows better mask AP than the original paper. The pretrained model is publicly available from the Tensorpack model zoo. Image centric training [14] is applied so that the images are resized to 800 pixels on the shorter edge, 1333 pixels on the longer edge, without changing the aspect ratio. Each image has 512 sampled RoIs, and their positive to negatives ratio is 1:3. We use 8 Titan RTX GPUs in training and single image per GPU for 360000 iterations. The learning rate is 0.02, weight decay is 0.0001, and momentum is 0.9. Other configurations are the same as Mask R-CNN. The RPN is trained separately and do not share the weights with Mask R-CNN. In addition, all ablation studies are tested based on the ResNet-50-FPN backbone for faster training/testing speed.

4.3. Examples with visual explanation

Please zoom in to see the visual explanation for techniques about architectures in Figure 5. We can see the powerful effects that help Mask R-CNN do better work.

4.4. Contextual Fusion Results

Table 1 shows the results for adding the contextual fusion technique to the original Mask R-CNN framework, and we can see the improvements from using such technique. We observe that the configuration of a stack of convolutional layers between the new RoIAlign layer and the adding layer also affects the improvement. We tried different configurations and find the best to be [720, 512, 512, 256] (the numbers represent the filter numbers of these consecutive convolutional layers, number of layers change according to configuration), and even with deeper layers or wider filter numbers, the performance will decrease on the contrary (because simply adding more capacity on a big network will increase the difficulties on training optimization.).

With the help of the contextual fusion, the AP of mask branch increases from 35.1 to 35.5, while the precision of the detection component is not affected. More improvement is gained for middle- and large-size objects. This validates the function of the contextual fusion.

4.5. Deconvolutional Pyramid Module Results

In our deconvolutional pyramid module, two deconvolutional layers are followed by two convolutional layers, as shown in Figure 3. They have strides of 2 and filter size of 256. The features are added instead of concatenated. The experimental results of applying this module are shown in Table 2. We can see that using this technique increases the mask AP from 35.1 to 35.4. Moreover, the accuracy is mainly improved for small- and middle-size objects.

Figure 5. 1). In the blue box, Mask R-CNN is at left. MaskPlus with only contextual fusion is at right – things like potted plant that should not appear in that scenario are removed, and the boundary fragment is correctly classified. 2). In the green box, Mask R-CNN is at left. MaskPlus with only deconvolutional pyramid module is at right – redundancy is removed with the help of multi-scale feature information. 3). In the yellow box, Mask R-CNN is at upper left. Original boundary refinement is at middle left. Only using improved boundary refinement is at lower left. The right side is the result from using improved boundary refinement in MaskPlus – the boundary is refined and has better quality from top to bottom. 4). In the purple box, Mask R-CNN is at left. MaskPlus with only quasi-multitask learning is at right – multi-scale supervision prevents some misclassifications from single-scale supervision.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN	35.1	56.6	37.5	18.4	38.4	48.3
Mask R-CNN + Contextual fusion 256-256	35.3	56.7	37.6	18.6	38.6	48.6
Mask R-CNN + Contextual fusion 256-256-256	35.4	56.9	37.7	18.4	38.5	48.7
Mask R-CNN + Contextual fusion 720-512-512-256	35.5	57.0	37.8	18.5	38.7	48.8

Table 1. Ablation study of contextual fusion.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN	35.1	56.6	37.5	18.4	38.4	48.3
Mask R-CNN + Deconvolutional pyramid module	35.4	56.9	37.6	18.8	38.6	48.3

Table 2. Ablation study of deconvolutional pyramid module.

4.6. Improved Boundary Refinement Results

We choose a stack of convolutional modules with dense connections to learn the boundary deficiency. Specifically, each convolutional module consists of six layers in order: BatchNorm, PReLu, Conv (filters = 16), BatchNorm, PReLu, and Conv (filters = 4). The later modules will concatenate the input features from all previous modules as its own input features. The concatenation will be repeated for 4 modules. Figure 3 reflects this kind of design pattern.

We compare our improved boundary refinement method with the original Mask R-CNN and the Mask R-CNN with boundary refinement method described in [26]. The results are presented in Table 3. Our approach provides improvements on mask precision over both cases. Note that the precision is mostly improved for middle- and large-size objectives (no improvement for small-size objects).

4.7. Quasi-multitask Learning Results

Instead of augmenting in front or middle of the networks, we developed the quasi-multitask learning technique to augment in the end (after the last layer of the networks). In the original Mask R-CNN configuration, the features from the RoIAlign layer will come across several convolutional layers with small filter numbers (these layers are the original settings in Mask R-CNN, noted to be L), and then they are

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN	35.1	56.6	37.5	18.4	38.4	48.3
Mask R-CNN + Original boundary refinement	35.2	56.6	37.6	18.3	38.2	48.4
Mask R-CNN + Improved boundary refinement	35.5	56.9	38.0	18.3	38.8	49.1

Table 3. Ablation study of improved boundary refinement.

up-sampled to 2x size (ground truth mask is resized to 28 * 28). Here we create another branch instead (parallel to L) follows the RoIAlign layer features. This branch has the same layers as L except for one layer – we will delete the last deconvolutional layer or add a deconvolutional layer at last with 2x upsampling scale to keep the output features to be 0.5x or 2x scale. Then the output mask will compute loss with 0.5x size of the original configuration (14 * 14) or 2x size (56 * 56). And for the most important thing is that the output mask is still the original branch, not the newly created one (whose purpose is only to help calculate the quasi-multitask learning loss). As shown in Table 4, both the 0.5x and 2x quasi-multitask learning demonstrate improvements on the mask accuracy.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN (1.x mask)	35.1	56.6	37.5	18.4	38.4	48.3
Mask R-CNN (0.5x mask)	34.3	56.4	36.7	17.9	37.7	46.4
Mask R-CNN (1.x mask) + 0.5x quasi-multitask learning	35.4	56.7	37.7	18.4	38.6	48.7
Mask R-CNN (1.x mask) + 2x quasi-multitask learning	35.2	56.5	37.5	18.5	38.3	48.4

Table 4. Ablation study of quasi-multitask learning.

4.8. Biased Training Results

As introduced in Section 3, we try to reduce the (negative) influence of the mask branch training in the first half stages on the detection component. Recall that the multi-task loss is: $L = L_{cls} + L_{box} + \alpha(L_{mask})$, and we initially set $\alpha = 1.5$ to increase the learning rate of the mask branch and force it to converge faster in early stages ($\alpha = 1$ in the later stages). Table 5 shows the results of such biased training in detection, and Table 6 shows its mask results. We can see that both detection component and mask branch gain benefits from biased training.

4.9. Overall Effectiveness of MaskPlus

We adopt all methods in best settings introduced above to generate a final model, aka MaskPlus. We compare our

Method	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{bb}_S	AP^{bb}_M	AP^{bb}_L
Mask R-CNN	38.3	59.8	41.6	21.8	41.9	50.3
Mask R-CNN + Biased training	38.6	60.0	41.8	21.7	42.1	51.3

Table 5. Ablation study of biased training on the detection component.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN	35.1	56.6	37.5	18.4	38.4	48.3
Mask R-CNN + Biased training	35.4	56.7	37.8	18.2	38.5	48.8

Table 6. Ablation study of biased training on the mask results.

MaskPlus with state-of-the-art approaches in the literature. As shown in Table 7, our approach clearly shows the state-of-the-art performance. To be specific, MaskPlus outperforms original Mask R-CNN in all provided metrics. And even compared with concurrent works [16] [7] which are also improved works based on Mask R-CNN, we can still give a competitive results. And we also create MaskPlus+ which is achieved based on a naive cascade version of Mask R-CNN as described in [4] (also used in [7]), which takes 0.5 times longer training steps for better model convergence. Note that some other techniques such as ResNeXt-101, multi-GPU synchronized batch normalization, Atrous Spatial Pyramid Pooling etc, that used in these concurrent works are not engaged in our work(thus there is potential for further improvement and collaboration). Finally, some results are visualized in Figure 6. And our result can be seen on CodaLab COCO leaderboard.

5. Conclusion

In this paper, we presented five methods for improving the mask generation in instance segmentation. It contains three novel techniques contextual fusion, quasi-multitask learning and biased training, and we also extend the existing techniques, including boundary refinement, deconvolutional pyramid module, to further improve the accuracy. We incorporated these techniques into Mask R-CNN to build our MaskPlus framework, and conducted tests on the COCO dataset. The experiments demonstrate our results in details and shows the state-of-the-art performance.

Acknowledgement

We gratefully acknowledge the support from the US National Science Foundation awards 1724341 and 1834701.

References

[1] H. A. Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother. Augmented reality meets deep learning for car instance segmentation in urban scenes. In *British Machine Vision Conference*, volume 1, page 2, 2017.

Figure 6. **MaskPlus visual results on the COCO dataset.** Masks are shown in color, and bounding box, category, and confidences are also presented. The figure is best viewed in color.

Method	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS + OHEM	ResNet-101-dilated	29.2	49.5	-	7.1	31.3	50.0
Mask R-CNN	ResNet-101-FPN	36.9	59.0	39.5	19.9	39.7	48.3
MaskLab+ [8]	ResNet-101	37.3	59.8	39.6	19.1	40.5	50.6
MaskLab+ [8]	ResNet-101 (JFT)	38.1	61.1	40.4	19.6	41.6	51.4
Mask Scoring R-CNN [16]	ResNet-101-FPN	38.3	58.8	41.5	17.8	40.4	54.4
HTC [7]	ResNet-101-FPN	39.7	61.8	43.1	21.0	42.2	53.5
MaskPlus	ResNet-101-FPN	38.1	59.9	41.0	20.2	40.8	50.4
MaskPlus+	ResNet-101-FPN	40.9	63.0	44.5	23.5	43.6	52.3

Table 7. The overall performance of our MaskPlus framework, compared with previous methods (including newly-added concurrent works [7, 16]) on the COCO test-dev subset. Final results are on the CodaLab COCO leaderboard. MaskPlus+ is achieved based on a simple cascade version of Mask R-CNN as described in [7, 4]. Note that some other techniques such as ResNeXt-101, multi-GPU synchronized batch normalization, Atrous Spatial Pyramid Pooling, etc, that used in these concurrent works are not used in our work (thus there is potential for further improvement).

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[3] A. Bewley, V. Guizilini, F. Ramos, and B. Upcroft. Online self-supervised multi-instance segmentation of dynamic objects. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1296–1303. IEEE, 2014.

[4] Z. Cai and N. Vasconcelos. Cascade R-CNN: delving into high quality object detection. *CoRR*, abs/1712.00726, 2017.

[5] H. Chen, X. Qi, L. Yu, Q. Dou, J. Qin, and P.-A. Heng. Dcan: Deep contour-aware networks for object instance segmentation from histology images. *Medical image analysis*, 36:135–146, 2017.

[6] H. Chen, Y. Zheng, J.-H. Park, P.-A. Heng, and S. K. Zhou. Iterative multi-domain regularized deep learning for anatomical structure detection and segmentation from ultrasound images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 487–495. Springer, 2016.

[7] K. Chen, J. Pang, et al. Hybrid task cascade for instance

segmentation. *CoRR*, abs/1901.07518, 2019.

[8] L. Chen, A. Hermans, et al. Masklab: Instance segmentation by refining object detection with semantic and direction features. *CoRR*, abs/1712.04837, 2017.

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[10] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[11] 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.

[12] B. De Brabandere, D. Neven, and L. Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017.

[13] A. Fathi, Z. Wojna, V. Rathod, P. Wang, H. O. Song, S. Guadarrama, and K. P. Murphy. Semantic instance segmentation via deep metric learning. *arXiv preprint arXiv:1703.10277*, 2017.

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

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[16] Z. Huang, L. Huang, et al. Mask scoring R-CNN. *CoRR*, abs/1903.00241, 2019.

[17] S. Kong and C. C. Fowlkes. Recurrent pixel embedding for instance grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9018–9028, 2018.

[18] 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.

[19] X. Liang, Y. Wei, X. Shen, J. Yang, L. Lin, and S. Yan. Proposal-free network for instance-level object segmentation. *arXiv preprint arXiv:1509.02636*, 2015.

[20] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE conference on computer vision and pattern recognition (CVPR)*, volume 1, page 3, 2017.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] A. Newell, Z. Huang, and J. Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems*, pages 2277–2287, 2017.

[26] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun. Large kernel mattersimprove semantic segmentation by global convolutional network. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1743–1751. IEEE, 2017.

[27] 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.

[28] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[29] M. Sajjad, S. Khan, Z. Jan, K. Muhammad, H. Moon, J. T. Kwak, S. Rho, S. W. Baik, and I. Mehmood. Leukocytes classification and segmentation in microscopic blood smear: a resource-aware healthcare service in smart cities. *IEEE Access*, 5:3475–3489, 2016.

[30] A. A. Shvets, A. Rakhlin, A. A. Kalinin, and V. I. Iglovikov. Automatic instrument segmentation in robot-assisted surgery using deep learning. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 624–628. IEEE, 2018.

[31] N. Silberman, D. Sontag, and R. Fergus. Instance segmentation of indoor scenes using a coverage loss. In *European Conference on Computer Vision*, pages 616–631. Springer, 2014.

[32] J. Uhrig, M. Cordts, U. Franke, and T. Brox. Pixel-level encoding and depth layering for instance-level semantic labeling. In *German Conference on Pattern Recognition*, pages 14–25. Springer, 2016.

[33] J. van den Brand, M. Ochs, and R. Mester. Instance-level segmentation of vehicles by deep contours. In *Asian Conference on Computer Vision*, pages 477–492. Springer, 2016.

[34] Y. Wu et al. Tensorpack. https://github.com/tensorpack/, 2016.

[35] Y. Xu, Y. Li, Y. Wang, M. Liu, Y. Fan, M. Lai, I. Eric, and C. Chang. Gland instance segmentation using deep multichannel neural networks. *IEEE Transactions on Biomedical Engineering*, 64(12):2901–2912, 2017.

[36] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[37] Z. Zhang, A. G. Schwing, S. Fidler, and R. Urtasun. Monocular object instance segmentation and depth ordering with cnns. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2614–2622, 2015.

[38] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.