Mask Scoring R-CNN

Zhaojin Huang[†*] Lichao Huang[‡] Yongchao Gong[‡] Chang Huang[‡] Xinggang Wang[†]
[†]Institute of AI, School of EIC, Huazhong University of Science and Technology
[‡]Horizon Robotics Inc.
{zhaojinhuang,xgwang}@hust.edu.cn {lichao.huang,yongchao.gong,chang.huang}@horizon.ai

Abstract

Letting a deep network be aware of the quality of its own predictions is an interesting yet important problem. In the task of instance segmentation, the confidence of instance classification is used as mask quality score in most instance segmentation frameworks. However, the mask quality, quantified as the IoU between the instance mask and its ground truth, is usually not well correlated with classification score. In this paper, we study this problem and propose Mask Scoring R-CNN which contains a network block to learn the quality of the predicted instance masks. The proposed network block takes the instance feature and the corresponding predicted mask together to regress the mask IoU. The mask scoring strategy calibrates the misalignment between mask quality and mask score, and improves instance segmentation performance by prioritizing more accurate mask predictions during COCO AP evaluation. By extensive evaluations on the COCO dataset, Mask Scoring R-CNN brings consistent and noticeable gain with different models, and outperforms the state-of-the-art Mask R-CNN. We hope our simple and effective approach will provide a new direction for improving instance segmentation. The source code of our method is available at https://github.com/zjhuang22/maskscoring_rcnn.

1. Introduction

Deep networks are dramatically driving the development of computer vision, leading to a series of state-of-the-art in tasks including classification [22, 16, 35], object detection [12, 17, 32, 27, 33, 34], semantic segmentation [28, 4, 37, 18] *etc*. From the development of deep learning in computer vision, we can observe that the ability of deep networks is gradually growing from making image-level prediction [22] to region/box-level prediction [12], pixel-level prediction [28] and instance/mask-level prediction [15]. The ability of making fine-grained predictions requires not only more detailed labels but also more delicate network designing.

In this paper, we focus on the problem of instance segmentation, which is a natural next step of object detection to move from coarse box-level instance recognition to precise pixel-level classification. Specifically, this work presents a novel method to score the instance segmentation hypotheses, which is quite important for instance segmentation evaluation. The reason lies in that most evaluation metrics are defined according to the hypothesis scores, and more precise scores help to better characterize the model performance. For example, precision-recall curves and average precision (AP) are often used for the challenging instance segmentation dataset COCO [26]. If one instance segmentation hypothesis is not properly scored, it might be wrongly regarded as false positive or false negative, resulting in a decrease of AP.

However, in most instance segmentation pipelines, such as Mask R-CNN [15] and MaskLab [3], the score of the instance mask is shared with box-level classification confidence, which is predicted by a classifier applied on the proposal feature. It is inappropriate to use classification confidence to measure the mask quality since it only serves for distinguishing the semantic categories of proposals, and is not aware of the actual quality and completeness of the instance mask. The misalignment between classification confidence and mask quality is illustrated in Fig. 1, where instance segmentation hypotheses get accurate box-level localization results and high classification score, but the corresponding masks are inaccurate. Obviously, scoring the masks using such classification score tends to degrade the evaluation results.

Unlike the previous methods that aim to obtain more accurate instance localization or segmentation mask, our method focuses on scoring the masks. To achieve this goal, our model learns a score for each mask instead of using its classification score. For clarity, we call the learned score mask score.

Inspired by the AP metric of instance segmentation that

[*] The work was done when Zhaojin Huang was an intern in Horizon Robotics Inc.

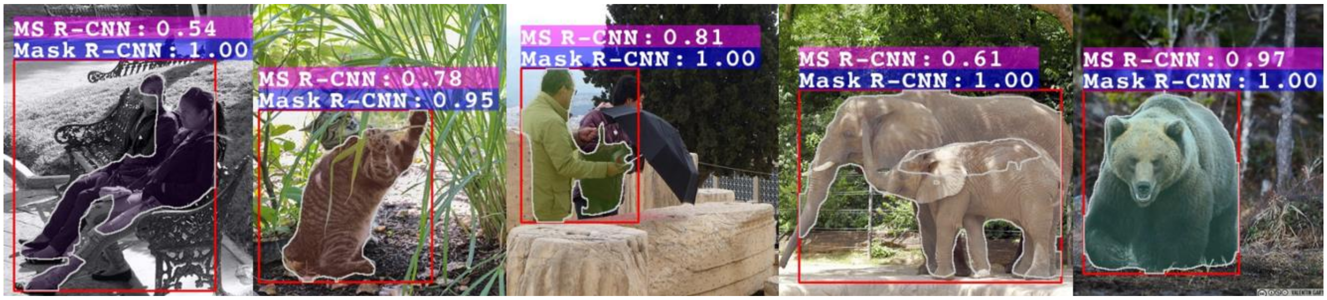

Figure 1. Demonstrative cases of instance segmentation in which bounding box has a high overlap with ground truth and a high classification score while the mask is not good enough. The scores predicted by both Mask R-CNN and our proposed MS R-CNN are attached above their corresponding bounding boxes. The left four images show good detection results with high classification scores but low mask quality. Our method aims at solving this problem. The rightmost image shows the case of a good mask with a high classification score. Our method will retrain the high score. As can be seen, scores predicted by our model can better interpret the actual mask quality.

uses pixel-level Intersection-over-Union (IoU) between the predicted mask and its ground truth mask to describe instance segmentation quality, we propose a network to learn the IoU directly. In this paper, this IoU is denoted as MaskIoU. Once we obtain the predicted MaskIoU in testing phase, mask score is reevaluated by multiplying the predicted MaskIoU and classification score. Thus, mask score is aware of both semantic categories and the instance mask completeness.

Learning MaskIoU is quite different from proposal classification or mask prediction, as it needs to "compare" the predicted mask with object feature. Within the Mask R-CNN framework, we implement a MaskIoU prediction network named MaskIoU head. It takes both the output of the mask head and RoI feature as input, and is trained using a simple regression loss. We name the proposed model, namely Mask R-CNN with MaskIoU head, as Mask Scoring R-CNN (MS R-CNN). Extensive experiments with our MS R-CNN have been conducted, and the results demonstrate that our method provides consistent and noticeable performance improvement attributing to the alignment between mask quality and score.

In summary, the main contributions of this work are highlighted as follows:

1. We present Mask Scoring R-CNN, the first framework that addresses the problem of scoring instance segmentation hypothesis. It explores a new direction for improving the performance of instance segmentation models. By considering the completeness of instance mask, the score of instance mask can be penalized if it has high classification score while the mask is not good enough.

2. Our MaskIoU head is very simple and effective. Experimental results on the challenging COCO benchmark show that when using mask score from our MS R-CNN rather than only classification confidence, the AP improves consistently by about 1.5% with various backbone networks.

2. Related Work

2.1. Instance Segmentation

Current instance segmentation methods can be roughly categorized into two classes. One is detection based methods and the other is segmentation based methods. Detection based methods exploit the state-of-the-art detectors, such as Faster R-CNN [33], R-FCN [8], to get the region of each instance, and then predict the mask for each region. Pinheiro *et al*. [31] proposed DeepMask to segment and classify the center object in a sliding window fashion. Dai *et al*. [6] proposed instance-sensitive FCNs to generate the position-sensitive maps and assembled them to obtain the final masks. FCIS [23] takes position-sensitive maps with inside/outside scores to generate the instance segmentation results. He *et al*. [15] proposed Mask R-CNN that is built on the top of Faster R-CNN by adding an instance-level semantic segmentation branch. Based on Mask R-CNN, Chen *et al*. [3] proposed MaskLab that used position-sensitive scores to obtain better results. However, an underlying drawback in these methods is that mask quality is only measured by the classification scores, thus resulting in the issues discussed above.

Segmentation based methods predict the category labels of each pixel first and then group them together to form instance segmentation results. Liang *et al*. [24] used spectral clustering to cluster the pixels. Other works, such as [20, 21], add boundary detection information during the clustering procedure. Bai *et al*. [1] predicted pixel-level energy values and used watershed algorithms for grouping. Recently, there are some works [30, 11, 14, 10] using metric learning to learn the embedding. Specifically, these methods learn an embedding for each pixel to ensure that pixels from the same instance have similar embedding. Afterwards, clustering is performed on the learned embed-

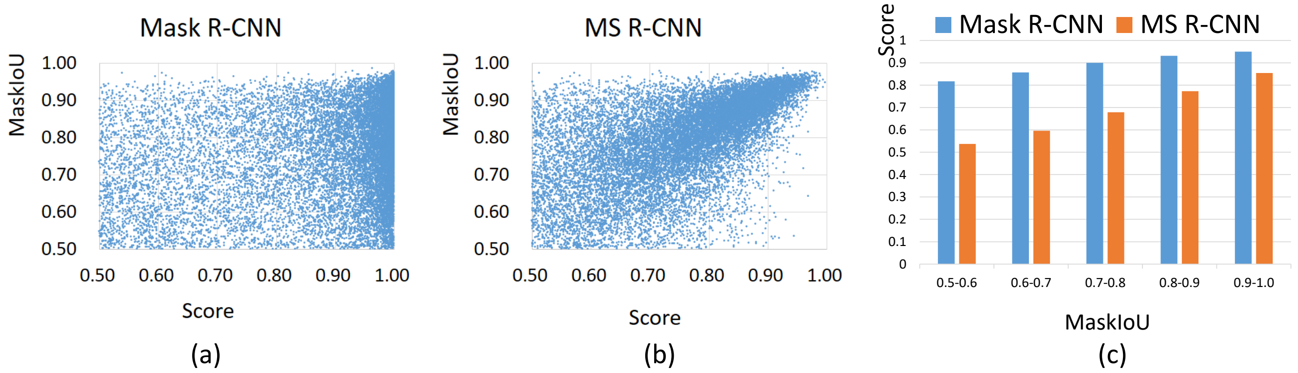

Figure 2. Comparisons of Mask R-CNN and our proposed MS R-CNN. (a) shows the results of Mask R-CNN, the mask score has less relationship with MaskIoU. (b) shows the results of MS R-CNN, we penalize the detection with high score and low MaskIoU, and the mask score can correlate with MaskIoU better. (c) shows the quantitative results, where we average the score between each MaskIoU interval, we can see that our method can have a better correspondence between score and MaskIoU.

ding to obtain the final instance labels. As these methods do not have explicit scores to measure the instance mask quality, they have to use the averaged pixel-level classification scores as an alternative.

Both classes of the above methods do not take into consideration the alignment between mask score and mask quality. Due to the unreliability of mask score, a mask hypothesis with higher IoU against ground truth is vulnerable to be ranked with low priority if it has a low mask score. In this case, the final AP is consequently degraded.

2.2. Detection Score Correction

There are several methods focusing on correcting the classification score for the detection box, which have a similar goal to our method. Tychsen-Smith *et al*. [36] proposed Fitness NMS that corrected the detection score using the IoU between the detected bounding boxes and their ground truth. It formulates box IoU prediction as a classification task. Our method differs from this method in that we formulate mask IoU estimation as a regression task. Jiang *et al*. [19] proposed IoU-Net that regressed box IoU directly, and the predicted IoU was used for both NMS and bounding box refinement. In [5], Cheng *et al*. discussed the false positive samples and used a separated network for correcting the score of such samples. SoftNMS [2] uses the overlap between two boxes to correct the low score box. Neumann *et al*. [29] proposed Relaxed Softmax to predict temperature scaling factor value in standard softmax for safety-critical pedestrian detection.

Unlike these methods that focus on bounding box level detection, our method is designed for instance segmentation. The instance mask is further processed in our MaskIoU head so that the network can be aware of the completeness of instance mask, and the final mask score can reflect the actual quality of the instance segmentation hypothesis. It is a new direction for improving the performance of instance segmentation.

3. Method

3.1. Motivation

In the current Mask R-CNN framework, the score of a detection (*i.e.*, instance segmentation) hypothesis is determined by the largest element in its classification scores. Due to the problems of background clutter, occlusion *etc*., it is possible that the classification score is high but the mask quality is low, as the examples shown in Fig. 1. To quantitatively analyze this problem, we compare the vanilla mask score from Mask R-CNN with the actual IoU between the predicted mask and its ground truth mask (MaskIoU). Specifically, we conduct experiments using Mask R-CNN with ResNet-18 FPN on COCO 2017 validation dataset. Then we select the detection hypotheses after Soft-NMS with both MaskIoU and classification scores larger than 0.5. The distribution of MaskIoU over classification score is shown in Fig. 2 (a) and the average classification score in each MaskIoU interval is shown in blue in Fig. 2 (c). These figures show that *classification score and MaskIoU is not well correlated in Mask R-CNN*.

In most instance segmentation evaluation protocols, such as COCO, a detection hypothesis with a low MaskIoU and a high score is harmful. In many practical applications, it is important to determine when the detection results can be trusted and when they cannot [29]. This motivates us to learn a calibrated mask score according to MaskIoU for every detection hypothesis. Without loss of generality, we work on the Mask R-CNN framework, and propose Mask Scoring R-CNN (MS R-CNN), a Mask R-CNN with an additional MaskIoU head module that learns the MaskIoU aligned mask score. The predicted mask scores of our framework are shown in Fig. 2 (b) and the orange histogram in Fig. 2 (c).

3.2. Mask scoring in Mask R-CNN

Mask Scoring R-CNN is conceptually simple: Mask R-CNN with MaskIoU Head, which takes the instance feature and the predicted mask together as input, and predicts the IoU between input mask and ground truth mask, as shown in Fig. 3. We will present the details of our framework in the following sections.

Mask R-CNN: We begin by briefly reviewing the Mask R-CNN [15]. Following Faster R-CNN [33], Mask R-CNN consists of two stages. The first stage is the Region Proposal Network (RPN). It proposes candidate object bounding boxes regardless of object categories. The second stage is termed as the R-CNN stage, which extracts features using RoIAlign for each proposal and performs proposal classification, bounding box regression and mask predicting.

Mask scoring: We define s_{mask} as the score of the predicted mask. The ideal s_{mask} is equal to the pixel-level IoU between predicted mask and its matched ground truth mask, which is termed as MaskIoU before. The ideal s_{mask} also should only have positive value for ground truth category, and be zero for other classes, since a mask only belong to one class. This requires the mask score to works well on two task: classifying the mask to right category and regressing the proposal's MaskIoU for foreground object category.

It is hard to train the two tasks only using a single objective function. For simplify, we can decompose the mask score learning task into mask classification and IoU regression, denoted as $s_{\text{mask}} = s_{\text{cls}} \cdot s_{\text{iou}}$ for all object categories. s_{cls} focuses on classifying the proposal belong to which class and s_{iou} focuses on regressing the MaskIoU.

As for s_{cls}, the goal of s_{cls} is to classify the proposal belonging to which class, which has been done in the classification task in the R-CNN stage. So we can directly take the corresponding classification score. Regressing s_{iou} is the target of this paper, which is discussed in the following paragraph.

MaskIoU head: The MaskIoU head aims to regress the IoU between the predicted mask and its ground truth mask. We use the concatenation of feature from RoIAlign layer and the predicted mask as the input of MaskIoU head. When concatenating, we use a max pooing layer with kernel size of 2 and stride of 2 to make the predicted mask have the same spatial size with RoI feature. We only choose to regress the MaskIoU for the ground truth class (for testing, we choose the predicted class) instead of all classes. Our MaskIoU head consists of 4 convolution layers and 3 fully connected layers. For the 4 convolution layers, we follow Mask head and set the kernel size and filter number to 3 and 256 respectively for all the convolution layers. For the 3 fully connected (FC) layers, we follow the RCNN head and set the outputs of the first two FC layers to 1024 and the output of the final FC to the number of classes.

Training: For training the MaskIoU head, we use the RPN proposals as training samples. The training samples are required to have a IoU between proposal box and the matched ground truth box larger than 0.5, which are the same with the training samples of the Mask head of Mask R-CNN. For generating the regression target for each training sample, we firstly get the predicted mask of the target class and binarize the predicted mask using a threshold of 0.5

Then we use the MaskIoU between the binary mask and its matched ground truth as the MaskIoU target. We use the ℓ_2 loss for regressing MaskIoU, and the loss weight is set to 1. The proposed MaskIoU head is integrated into Mask R-CNN, and the whole network is end to end trained.

Inference: During inference, we just use MaskIoU head to calibrate classification score generated from R-CNN. Specifically, suppose the R-CNN stage of Mask R-CNN outputs N bounding boxes, and among them top-k (*i.e.* k = 100) scoring boxes after SoftNMS [2] are selected. Then the top-k boxes are fed into the Mask head to generate multi-class masks. This is the standard Mask R-CNN inference procedure. We follow this procedure as well, and feed the top-k target masks to predict the MaskIoU. The predicted MaskIoU are multiplied with classification score, to get the new calibrated mask score as the final mask confidence.

4. Experiments

All experiments are conducted on the COCO dataset [26] with 80 object categories. We follow COCO 2017 settings, using the 115k images *train* split for training, 5k *validation* split for validation, 20k *test-dev* split for test. We use COCO evaluation metrics AP (averaged over IoU thresholds) to report the results, including AP@0.5, AP@0.75, and AP_S, AP_M, AP_L (AP at different scales). AP@0.5 (or AP@0.75) means using an IoU threshold 0.5 (or 0.75) to identify whether a predicted bounding box or mask is positive in the evaluation. Unless noted, AP is evaluated using mask IoU.

4.1. Implementation Details

We use our reproduced Mask R-CNN for all experiments. We use ResNet-18 based FPN network for ablation study and ResNet-18/50/101 based on Faster R-CNN/FPN/DCN+FPN [9] for comparing our method with other baseline results. For ResNet-18 FPN, input images

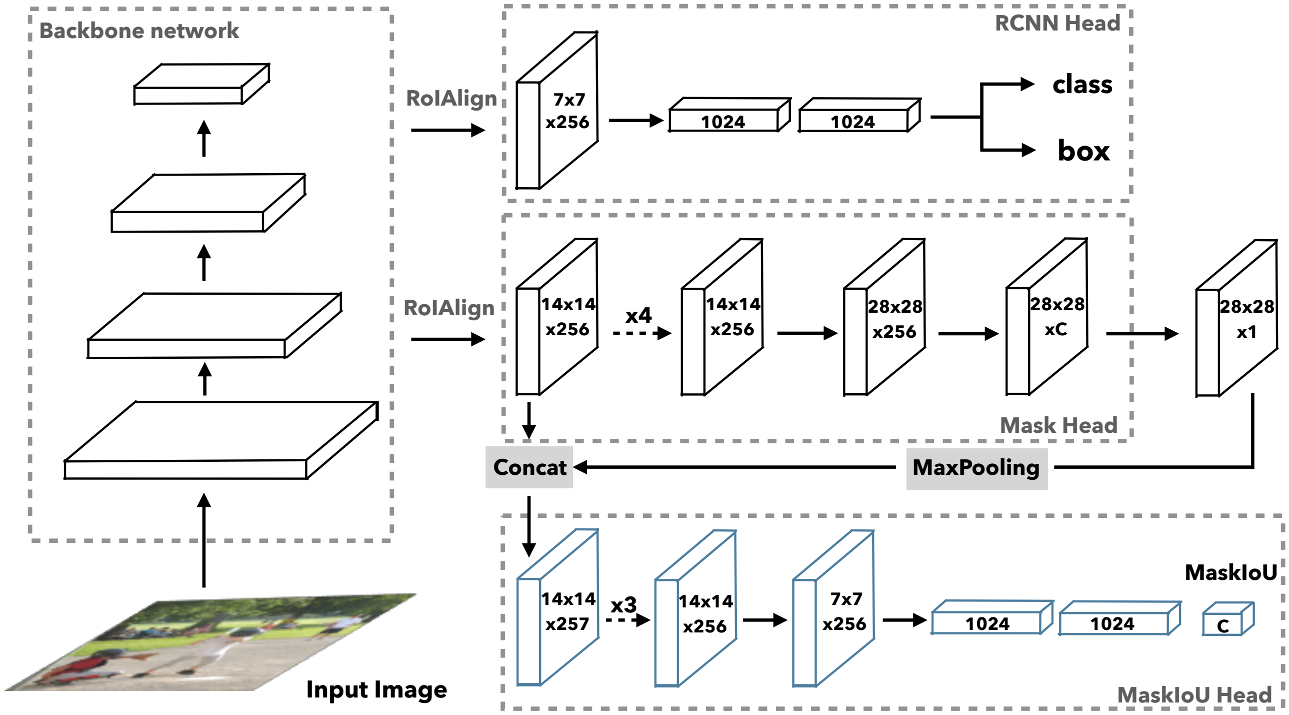

Figure 3. Network architecture of Mask Scoring R-CNN. The input image is fed into a backbone network to generate RoIs via RPN and RoI features via RoIAlign. The RCNN head and Mask head are standard components of Mask R-CNN. For predicting MaskIoU, we use the predicted mask and RoI feature as input. The MaskIoU head has 4 convolution layers (all have kernel=3 and the final one uses stride=2 for downsampling) and 3 fully connected layers (the final one outputs C classes MaskIoU.)

Table 1. COCO 2017 validation results. We report both detection and instance segmentation results. AP_m denotes instance segmentation results and AP_b denotes detection results. The results without ✓ are those of Mask R-CNN, while with ✓ are those of our MS R-CNN. The results show that our method is insensitive to different backbone networks.

Backbone	MaskIoU head	AP_m	AP_m@0.5	AP_m@0.75	AP_b	AP_b@0.5	AP_b@0.75
ResNet-18 FPN		27.7	46.9	29.0	31.2	50.4	33.2
	✓	29.3	46.9	31.3	31.5	50.8	33.5
ResNet-50 FPN		34.5	55.8	36.7	38.6	59.2	42.5
	✓	36.0	55.8	38.8	38.6	59.2	42.5
ResNet-101 FPN		36.6	58.6	39.0	41.3	61.7	45.9
	✓	38.2	58.4	41.5	41.4	61.8	46.3

Table 2. COCO 2017 validation results. We report detection and instance segmentation results. AP_m denotes instance segmentation results and AP_b denotes detection results. In the results area, rows 1&2 use the Faster R-CNN framework; rows 3&4 additionally use FPN framework; rows 5&6 additionally use the DCN+FPN. The results show that consistent improvement of the proposed MaskIoU head.

Backbone	MaskIoU head	FPN	DCN	AP_m	AP_m@0.5	AP_m@0.75	AP_b	AP_b@0.5	AP_b@0.75
ResNet-101				33.9	53.9	36.2	38.6	57.3	42.8
	✓			35.0	54.0	37.7	38.7	57.4	43.0
		✓		36.6	58.6	39.0	41.3	61.7	45.9
	✓	✓		38.2	58.4	41.5	41.4	61.8	46.3
		✓	✓	37.7	60.3	40.0	42.9	63.4	47.8
	✓	✓	✓	39.1	60.0	42.4	43.1	63.5	47.7

Table 3. Comparing different instance segmentation methods on COCO 2017 test-dev.

Method	Backbone	AP	AP@0.5	AP@0.75	AP_S	AP_M	AP_L
MNC [7]	ResNet-101	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [23]	ResNet-101	29.2	49.5	-	-	-	-
FCIS+++ [23]	ResNet-101	33.6	54.5	-	-	-	-
Mask R-CNN [15]	ResNet-101	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN [15]	ResNet-101 FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN [15]	ResNeXt-101 FPN	37.1	60.0	39.4	16.9	39.9	53.5
MaskLab [3]	ResNet-101	35.4	57.4	37.4	16.9	38.3	49.2
MaskLab+ [3]	ResNet-101	37.3	59.8	36.6	19.1	40.5	50.6
MaskLab+ [3]	ResNet-101 (JET)	38.1	61.1	40.4	19.6	41.6	51.4
Mask R-CNN	ResNet-101	34.3	55.0	36.6	13.2	36.4	52.2
MS R-CNN		35.4	54.9	38.1	13.7	37.6	53.3
Mask R-CNN	ResNet-101 FPN	37.0	59.2	39.5	17.1	39.3	52.9
MS R-CNN		38.3	58.8	41.5	17.8	40.4	54.4
Mask R-CNN	ResNet-101 DCN+FPN	38.4	61.2	41.2	18.0	40.5	55.2
MS R-CNN		39.6	60.7	43.1	18.8	41.5	56.2

are resized to have 600px along the short axis and a maximum of 1000px along the long axis for training and testing. Different from the standard FPN [25], we only use C4, C5 for RPN proposal and feature extractor in ResNet-18. For ResNet-50/101, input images are resized to 800 px for the short axis and 1333px for the long axis for training and testing. The rest configurations for ResNet-50/101 follow Detectron [13]. We train all the networks for 18 epochs, decreasing the learning rate by a factor of 0.1 after 14 epochs and 17 epochs. Synchronized SGD with momentum 0.9 is used as optimizer. For testing, we use SoftNMS and retain the top-100 score detection for each image.

4.2. Quantitative Results

We report our results on different backbone networks including ResNet-18/50/101 and different framework including Faster R-CNN/FPN/DCN+FPN [9] to prove the effectiveness of our method. Results are shown in Table 1 and Table 2. We use AP_m to report instance segmentation results and AP_b to report detection results. We report our reproduced Mask R-CNN results and our MS R-CNN results. As Table 1 shows, comparing with Mask R-CNN, our MS R-CNN is not sensitive to the backbone network and can achieve stable improvement on all backbone networks: Our MS R-CNN can get a remarkable improvement (about 1.5 AP). Especially for AP@0.75, our method can improve baseline by about 2 points. Table 2 indicates that our MS R-CNN is robust to different framework including Faster R-CNN/FPN/DCN+FPN. Beside, our MS R-CNN does not harm bounding box detection performance; in fact, it improves bounding box detection performance slightly. The results of *test-dev* are reported in Table 3, only the instance segmentation results are reported.

4.3. Ablation Study

We comprehensively evaluate our method on COCO 2017 validation set. We use ResNet-18 FPN for all the ablation study experiments.

The design choices of MaskIoU head input: We first study the design choices of the MaskIoU head input, which is the fusion of predicted mask score map ($28 \times 28 \times C$) from the mask head and the RoI features. There are a few design choices shown in Fig. 4 and explained as follows:

(a) Target mask concatenates RoI feature: The score map of the target class is taken, max-pooled and concatenated with RoI feature.

(b) Target mask multiplies RoI feature: The score map of the target class is taken, max-pooled and multiplied with RoI feature.

(c) All masks concatenates RoI feature: All the C classes mask score map are max-pooled and concatenated with RoI feature.

(d) Target mask concatenates High-resolution RoI feature: The score map of the target class is taken and concatenated with 28×28 RoI features.

The results are shown in Table 4. We can see that the performance of MaskIoU head is robust to different ways of fusing mask prediction and RoI feature. Performance gain is observed in all kinds of design. Since concatenating the target score map and RoI feature obtains the best results, we use it as our default choice.

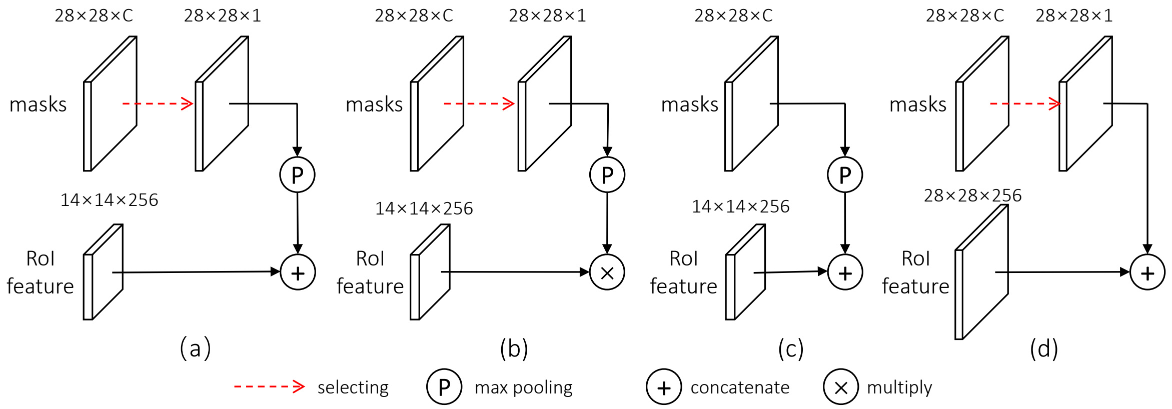

Figure 4. Different design choices of the MaskIoU head input.

Table 4. Results of different design choices of the MaskIoU head input.

Setting	AP	AP@0.5	AP@0.75
Mask R-CNN baseline	27.7	46.9	29.0
(a) Target mask + RoI	29.3	46.9	31.3
(b) Target mask × RoI	29.1	46.6	30.9
(c) All masks + RoI	29.1	46.6	30.8
(d) Target mask + HR RoI	29.1	46.7	31.1

The choices of the training target: As mentioned before, we decompose the mask score learning task as mask classification and MaskIoU regression. Is it possible to learn the mask score directly? In addition, a RoI may contain multiple categories of objects. Should we learn MaskIoU for all categories? How to set the training target for MaskIoU head still need exploration. There are many different choices of training target:

1. Learning the MaskIoU of the target category, meanwhile the other categories in the proposal are ignored. This is also the default training target in this paper, and the control group for all experiments in this paragraph.
2. Learning the MaskIoU for all categories. If a category does not appear in the RoI, its target MaskIoU is set to 0. This setting denotes using regression only to predict MaskIoU, which requires the regressor to be aware of the absence of unrelated categories.
3. Learning the MaskIoU of all the positive categories, where a positive category means the category appears in the RoI region. And the rest categories in the proposal are ignored. This setting is used to see whether perform regression for more categories in the RoI region could be better.

Table 5. Results of using different training targets.

Setting	AP	AP@0.5	AP@0.75
Mask R-CNN baseline	27.7	46.9	29.0
Setting #1: Target ins.	29.3	46.9	31.3
Setting #2: All cls.	24.5	41.6	25.6
Setting #3: Positive ins.	28.2	45.5	30.2

Table 5 shows the results for above training targets. By comparing setting #1 with setting #2, we can find that training MaskIoU of all categories (regression only based MaskIoU prediction) will degrade the performance drastically, which verifies our opinion that training classification and regression using a single objective function is difficult.

It is reasonable that the performance of setting #3 is inferior to setting #1, since regressing MaskIoU for all positive categories increases the burden of MaskIoU head. Thus, learning the MaskIoU of the target category is used as our default choice.

How to select training samples: Since the proposed MaskIoU head is built on top of the Mask R-CNN framework, all the training samples for the MaskIoU head have a box-level IoU larger than 0.5 with its ground truth bounding box according to the setting in the Mask R-CNN. However, their MaskIoU may not exceed 0.5.

Given a threshold τ, we use the samples whose MaskIoU are larger than τ to train the MaskIoU head. Table 6 shows the results. The results show that training using all the examples obtains the best performance.

4.4. Discussion

In this section, we will first discuss the quality of the predicted MaskIoU, and then investigate the upper bound performance of Mask Scoring R-CNN if the prediction of MaskIoU is perfect, and analyze the computational complexity of MaskIoU head at last. In the discussions, all

Table 6. Results of selecting different training samples for the MaskIoU head.

Threshold	AP	AP@0.5	AP@0.75
$\tau = 0.0$	29.3	46.9	31.3
$\tau = 0.3$	29.2	46.6	31.1
$\tau = 0.5$	29.0	46.5	30.9
$\tau = 0.7$	28.8	46.9	30.5

the results are obtained on COCO 2017 validation set using both a weak backbone network, *i.e*., ResNet-18 FPN and a strong backbone network, *i.e*., ResNet-101 DCN+FPN.

The quality of the predicted MaskIoU: We use correlation coefficient between ground truth and predicted MaskIoU to measure the quality of our prediction. Reviewing our testing procedure, we choose the top 100 scoring boxes after SoftNMS according to the classification scores, fed the detected boxes to Mask head and get the predicted mask, then use the predicted mask and RoI feature as the input of MaskIoU head. The output of MaskIoU head and classification score are further integrated into final mask score.

We keep 100 predicted MaskIoU for each image in the COCO 2017 validation dataset, collecting $500,000$ predictions from all $5,000$ images. We plot each predictions and their corresponding ground truth in Fig. 5. We can see that the MaskIoU predictions have good correlation with their ground truth, especially for those prediction with high MaskIoU. The correlation coefficient between predictions and their ground truth is around 0.74 for both ResNet-18 FPN and ResNet-101 DCN+FPN backbone networks. It indicates that the quality of the prediction is not sensitive to the change of backbone networks. This conclusion is also consistent with Table 1. Since there is no method works on predicting MaskIoU before, we refer to a previous work [19] on predicting bounding box IoU. [19] obtains a 0.617 correlation coefficient, which is inferior to ours.

The upper bound performance of MS R-CNN: Here we will discuss the upper bound performance of our method. For each predicted mask, we can find its matched ground truth mask; then we just use the ground truth MaskIoU to replace the predicted MaskIoU when the ground truth MaskIoU larger than 0. The results are shown in Table 7. The results show that Mask Scoring R-CNN consistently outperforms Mask R-CNN. Compared to the ideal prediction of Mask Scoring R-CNN, there is still a room to improve the practical Mask Scoring R-CNN, which are 2.2% AP for ResNet-18 FPN backbone and 2.6% AP for ResNet-101 DCN+FPN backbone.

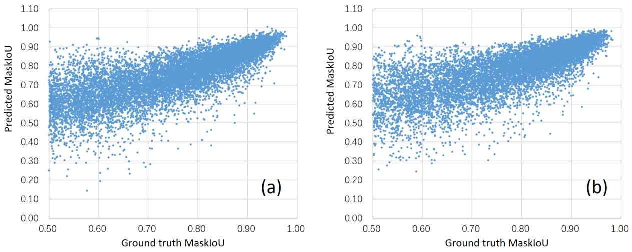

Figure 5. Visualizations of MaskIoU predictions and their ground truth. (a) Results with ResNet-18 FPN backbone and (b) results with ResNet-101 DCN+FPN backbone. The x-axis presents the ground truth MaskIoU and the y-axis presents the predicted MaskIoU of the proposed MaskIoU head.

Table 7. Results of Mask R-CNN, MS R-CNN and the ideal case of MS R-CNN (MS R-CNN*) using ResNet-18 FPN and ResNet-101 DCN+FPN as backbones on COCO 2017 validation set.

Method	Backbone	AP
Mask R-CNN	ResNet-18 FPN	27.7
MS R-CNN		29.3
MS R-CNN*		31.5
Mask R-CNN	ResNet-101 DCN+FPN	37.7
MS R-CNN		39.1
MS R-CNN*		41.7

Model size and running time: Our MaskIoU head has about 0.39G FLOPs while Mask head has about 0.53G FLOPs for each proposal. We use one TITAN V GPU to test the speed (sec./image). As for ResNet-18 FPN, the speed is about 0.132 for both Mask R-CNN and MS R-CNN. As for ResNet-101 DCN+FPN, the speed is about 0.202 for both Mask R-CNN and MS R-CNN. The computation cost of MaskIoU head in Mask Scoring R-CNN is negligible.

5. Conclusion

In this paper, we investigate the problem of scoring instance segmentation masks and propose Mask Scoring R-CNN. By adding a MaskIoU head in Mask R-CNN, scores of the masks are aligned with MaskIoU, which is usually ignored in most instance segmentation frameworks. The proposed MaskIoU head is extremely effective and easy to implement. On the COCO benchmark, extensive results show that Mask Scoring R-CNN consistently and obviously outperforms Mask R-CNN. It also can be applied to other instance segmentation networks to obtain more reliable mask scores. We hope our simple and effective approach will serve as a baseline and help the future research in instance segmentation task.

References

[1] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2858–2866, 2017. 2

[2] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Softnmsimproving object detection with one line of code. In *IEEE International Conference on Computer Vision*, pages 5562–5570, 2017. 3, 4

[3] 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. *arXiv preprint arXiv:1712.04837*, 2017. 1, 2, 6

[4] 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*, pages 834–848, 2018. 1

[5] B. Cheng, Y. Wei, H. Shi, R. Feris, J. Xiong, and T. Huang. Revisiting rcnn: On awakening the classification power of faster rcnn. In *European Conference on Computer Vision*, pages 473–490, 2018. 3

[6] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. In *European Conference on Computer Vision*, pages 534–549, 2016. 2

[7] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158, 2016. 6

[8] 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. 2

[9] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *IEEE International Conference on Computer Vision*, pages 764–773, 2017. 4, 6

[10] B. De Brabandere, D. Neven, and L. Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017. 2

[11] 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. 2

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. 1

[13] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. `https://github.com/facebookresearch/detectron`, 2018. 6

[14] A. W. Harley, K. G. Derpanis, and I. Kokkinos. Segmentation-aware convolutional networks using local attention masks. In *IEEE International Conference on Computer Vision*, pages 5048–5057, 2017. 2

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision*, pages 2980–2988, 2017. 1, 2, 4, 6

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

[17] L. Huang, Y. Yang, Y. Deng, and Y. Yu. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874*, 2015. 1

[18] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu. Ccnet: Criss-cross attention for semantic segmentation. *arXiv preprint arXiv:1811.11721*, 2018. 1

[19] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang. Acquisition of localization confidence for accurate object detection. In *European Conference on Computer Vision*, pages 816–832, 2018. 3, 8

[20] L. Jin, Z. Chen, and Z. Tu. Object detection free instance segmentation with labeling transformations. *arXiv preprint arXiv:1611.08991*, 2016. 2

[21] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother. Instancecut: from edges to instances with multicut. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7322–7331, 2017. 2

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 1

[23] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *IEEE International Conference on Computer Vision*, pages 4438–4446, 2017. 2, 6

[24] 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. 2

[25] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 936–944, 2017. 6

[26] 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, 2014. 1, 4

[27] 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, 2016. 1

[28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1

[29] L. Neumann, A. Zisserman, and A. Vedaldi. Relaxed softmax: Efficient confidence auto-calibration for safe pedestrian detection. 2018. 3

[30] 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. 2

[31] P. O. Pinheiro, R. Collobert, and P. Dollár. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pages 1990–1998, 2015. 2

[32] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016. 1

[33] 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. 1, 2, 4

[34] P. Tang, C. Wang, X. Wang, W. Liu, W. Zeng, and J. Wang. Object detection in videos by high quality object linking. *arXiv preprint arXiv:1801.09823*, 2018. 1

[35] P. Tang, X. Wang, Z. Huang, X. Bai, and W. Liu. Deep patch learning for weakly supervised object classification and discovery. *Pattern Recognition*, 71:446–459, 2017. 1

[36] L. Tychsen-Smith and L. Petersson. Improving object localization with fitness nms and bounded iou loss. *arXiv preprint arXiv:1711.00164*, 2017. 3

[37] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2881–2890, 2017. 1