

Boundary-preserving Mask R-CNN

Tianheng Cheng¹, Xinggang Wang^{†1}, Lichao Huang², and Wenyu Liu¹

¹ Huazhong University of Science and Technology
`{thch,xgwang,liuwy}@hust.edu.cn`

² Horizon Robotics Inc.
`lichao.huang@horizon.ai`

Abstract. Tremendous efforts have been made to improve mask localization accuracy in instance segmentation. Modern instance segmentation methods relying on fully convolutional networks perform pixel-wise classification, which ignores object boundaries and shapes, leading coarse and indistinct mask prediction results and imprecise localization. To remedy these problems, we propose a conceptually simple yet effective Boundary-preserving Mask R-CNN (BMask R-CNN) to leverage object boundary information to improve mask localization accuracy. BMask R-CNN contains a boundary-preserving mask head in which object boundary and mask are mutually learned via feature fusion blocks. As a result, the predicted masks are better aligned with object boundaries. Without bells and whistles, BMask R-CNN outperforms Mask R-CNN by a considerable margin on the COCO dataset; in the Cityscapes dataset, there are more accurate boundary groundtruths available, so that BMask R-CNN obtains remarkable improvements over Mask R-CNN. Besides, it is not surprising to observe that BMask R-CNN obtains more obvious improvement when the evaluation criterion requires better localization (*e.g.*, AP₇₅) as shown in Fig. 1. Code and models are available at <https://github.com/hustvl/BMaskR-CNN>.

Keywords: instance segmentation, object detection, boundary-preserving, boundary detection

1 Introduction

Instance segmentation, a fundamental but challenging task in computer vision, aims to assign a pixel-level mask to localize and categorize each object in images, driving numerous vision applications such as autonomous driving, robotics and image editing. With the rapid development of deep convolutional neural networks (DCNN), various methods based on DCNN were proposed for instance segmentation. Prevalent methods for instance segmentation are based on object detection, which provides box-level localization information for instance-level segmentation, among which Mask R-CNN [21] is the most successful one. It extends Faster R-CNN [44] by adding a simple fully convolutional network (FCN)

[†]Xinggang Wang is the corresponding author.

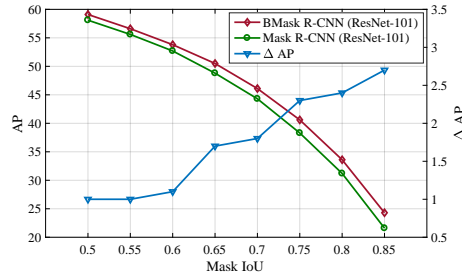


Fig. 1: AP curves of Mask R-CNN and BMask R-CNN under different mask IoU thresholds on the COCO *val2017* set. The blue line shows the AP gains of BMask R-CNN over Mask R-CNN.

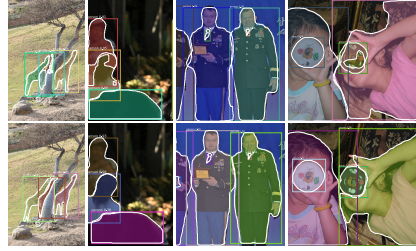


Fig. 2: **First row:** Selected cases of coarse boundaries appeared in the instance segmentation results of Mask R-CNN. **Second row:** Our proposed method can predict more precise boundaries.

to predict the mask of each detected instance. Due to the great effectiveness and flexibility, Mask R-CNN serves as a state-of-the-art baseline and has facilitated most recent instance segmentation research, such as [24,37,7,5,28].

In the Mask R-CNN framework, state-of-the-art instance segmentation networks [21,24,37] obtain instance masks by performing pixel-level classification via FCN. It treats all pixels in the proposal equally and ignores the object shape and boundary information. However, the pixels near boundaries are hard to be classified. Evidently, it is hard for pixel-level classifier to guarantee precise masks. We find that fine boundaries can provide better localization performance and make the object masks more distinct and clear. As illustrated in Fig. 2, Mask R-CNN (the first row) without consideration about boundaries is prone to output coarse and indistinct segmentation results with unreasonable overlaps between objects in comparison with the one that involves boundaries (the second row).

To address this issue, we leverage instance boundary information to enhance the mask prediction. Instance boundary is a dual representation of instance mask and it can guide the mask prediction network to output masks that are well-aligned with their groundtruths. Thus, the masks are more distinct and give more precise object location. Based on this motivation, we propose a conceptually simple and novel Boundary-preserving Mask R-CNN (BMask R-CNN) that unifies instance-level mask prediction and boundary prediction in one network.

Specifically, based on Mask R-CNN, we replace the original mask head with the proposed boundary-preserving mask head which contains two sub-networks for jointly learning object masks and boundaries. We insert two feature fusion blocks to strengthen the connection between boundary feature learning and mask feature learning. At last, mask prediction is guided by boundary features which contain abundant shape and localization information. The main purpose of learning boundaries is to capture features for precise object localization. Nevertheless, learning boundaries is non-trivial, because boundary groundtruths that gener-

ated from sparse annotated polygons (*i.e.*, in the COCO dataset [35]) are noisy and boundary classification has less training pixels than that for mask classification. To solve this problem, we further dive into the optimization for boundary learning by performing studies about boundary loss and exploit a boundary classification loss by combining binary cross-entropy loss and the dice loss [39].

We perform extensive experiments to evaluate the performance of BMask R-CNN. On the challenging COCO dataset [35], BMask R-CNN achieves considerably significant improvements compared with Mask R-CNN regardless of the backbones. Note that our BMask R-CNN provides larger gains if it requires more precise mask localization, as shown in Fig. 1. On the fine-annotated Cityscapes dataset [12], BMask R-CNN brings larger improvements with better mask annotations.

The main contributions of this paper can be summarized as follows.

- We present a novel Boundary-preserving Mask R-CNN (BMask R-CNN), which is the first work that explicitly exploits object boundary information to improve mask-level localization accuracy in the state-of-the-art Mask R-CNN framework.
- BMask R-CNN is conceptually simple yet effective. Without bells and whistles, BMask R-CNN outperforms Mask R-CNN by 1.7% AP and 2.2% AP on the COCO val set and the Cityscapes test set respectively. Further, BMask R-CNN obtains higher AP gains when the mask IoU threshold becomes higher, as shown in Fig. 1.
- We perform ablation studies on the components of BMask R-CNN, *e.g.*, feature fusion blocks, boundary features, boundary losses and the Sobel mask head, which are helpful to interpret how BMask R-CNN works and provide some thoughts for further research on instance segmentation.

2 Related Work

Instance Segmentation: Existing methods can be divided into two categories, *i.e.* detection-based methods and segmentation-based methods. Detection-based methods employ object detectors [18,44,15,34] to generate region proposals and then predict their masks after RoI pooling/align [18,21]. Based on CNN, [42,13,43] predict masks for object proposals. FCIS [33] extends InstanceFCN [13] by exploiting position-sensitive inside/outside score maps and fully convolutional networks for instance segmentation. BAIS [20] uses boundary-based distance transform to predict mask pixels that are beyond bounding boxes. Mask R-CNN [21] extends Faster R-CNN [44] by adding a mask prediction branch in parallel with the existing box regression and classification branches, demonstrating competitive performance on both object detection and instance segmentation. PANet [37] based on Mask R-CNN introduces the bottom-up path augmentation for FPN [34] to enhance information flow and adaptive feature pooling for better mask features. Mask scoring R-CNN [24] addresses the misalignment between mask quality and mask score in Mask R-CNN by explicitly learning the quality of predicted masks. [7] further improves Cascade Mask R-CNN [5] by interweaving

box and mask branches in a multi-stage cascade manner and providing spatial context through semantic segmentation. Huang *et al.* apply a criss-cross attention module [25] to capture the full-image contextual information for instance segmentation. [30] draws on the idea of rendering and adaptively selects key points to recover fine details for high-quality image segmentation.

Segmentation-based methods first exploit pixel-level segmentation over the image and then group the pixels together for each object. InstanceCut [29] adopts boundaries to partition semantic segmentation into instance-level segmentation. SGN [36] groups pixels along rows and columns by line segments. [47] utilizes predicted instance centers and pixel-wise directions to group instances. Recently, several methods [4,17] take the advantage of deep metric learning to learnt the embedding to group pixels to for instance segmentation.

Boundary, Edge and Segmentation: Deep fully convolutional neural networks has achieved great progress in edge detection. Xie *et al.* propose the fully convolutional holistically-nested edge detector HED [49] which performs in an image-to-image manner and end-to-end training. CASENet [52] presents a novel challenging task semantic boundary detection, aiming to detect category-aware boundaries. [53,1] investigate the label misalignment problem caused by noisy labels in semantic boundary detection. [50] proposes geometric aware loss function for object skeleton detection in nature images. In semantic segmentation, Chen *et al.* [10] propose fully connected conditional random field (CRF) [32] to capture spatial details and refine boundaries. Recent semantic segmentation methods [8,51,45,3,23] leverage predicted boundaries or edges to facilitate semantic segmentation. [11,54] refine segmentation results with direction fields learned from predicted boundaries. Zimmermann *et al.* propose edge agreement head [55] to focus on boundaries of instances with an auxiliary edge loss. Different from these previous methods, BMask R-CNN explicitly predicts instance-level boundaries, from which we obtain instance shape information for better mask localization. Compared to semantic segmentation, boundaries in instance segmentation have dual relations to the masks. Therefore, we build fusion blocks to mutually learn boundary and mask features and improve the representations for mask localization and lead the mask prediction focus more on the boundaries.

3 Boundary-preserving Mask R-CNN

3.1 Motivation

In Mask R-CNN, instance segmentation is performed based on pixel-level predictions. To learn a translation invariant predictor, predictions are made based on the local information. Though the local features extracted using deep network have large receptive fields, the shape information of object is ignored. Thus, the predicted masks often contain coarse and indistinct as well as some false positive predictions. For better understanding this problem, we analyze and visualize some raw mask prediction from Mask R-CNN with ResNet-50 [22] and FPN. As



Fig. 3: Visualization of some predicted masks (in the bottom row) of Mask R-CNN *vs.* their groundtruths (in the top row).

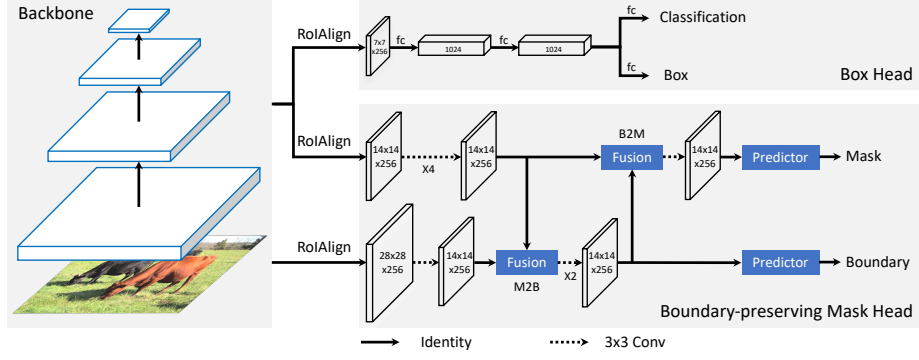


Fig. 4: The Overall architecture of **Boundary-preserving Mask R-CNN (BMask R-CNN)**. The dotted arrow denotes 3×3 convolution and the solid arrow denotes identity connection unless specified annotation in boundary-preserving mask head. “ $\times 4/\times 2$ ” denotes a stack of four/two consecutive convs. The predictor contains a 2×2 deconvolution and a class-specific 1×1 convolution as the output layer for both boundary and mask prediction.

shown in Fig. 3, some mask predictions are rough and imprecise. Obviously, employing object boundaries will be helpful to address this issue by providing better localization and guidance. Therefore, we propose a Boundary-preserving Mask R-CNN to exploit boundary information to guide more precise mask prediction.

3.2 Boundary-preserving Mask Head

BMask R-CNN improves the mask head in Mask R-CNN with boundary features and boundary prediction, as illustrated in Fig. 4. The new mask head termed as boundary-preserving mask head performs RoIAlign [21] to acquire RoI features for both boundary and mask prediction.

Boundary-preserving mask head jointly learns object boundaries and masks in an end-to-end manner. Note that object boundary and object mask have a close relation and we can easily convert either one to another. Features from the mask sub-network can provide high-level semantic information for learning boundaries. After obtaining boundaries, the shape information and localization information in boundary features can guide more precise mask predictions.

RoI Feature Extraction: We define \mathcal{R}_m and \mathcal{R}_b as Region of Interest (RoI) features for mask prediction and boundary prediction respectively. Following [34], \mathcal{R}_m is extracted from the specific feature pyramid level ($P2 \sim P5$) according to the scale of the proposal, while \mathcal{R}_b is obtained from the finest-resolution feature level $P2$, containing abundant spatial information. To preserve spatial information better for boundary prediction, the resolution of \mathcal{R}_b is set to be larger than that of \mathcal{R}_m when performing RoIAlign. Then, it is downsampled by a strided 3×3 convolution and the output feature is denoted as $\widetilde{\mathcal{R}}_b$. $\widetilde{\mathcal{R}}_b$ has the same resolution as \mathcal{R}_m and is used for feature fusion.

The feature fusion scheme in BMask R-CNN is illustrated in Fig. 4. Mask RoI features \mathcal{R}_m is fed into 4 consecutive 3×3 convolutions and the output feature is denoted as \mathcal{F}_m . Boundary features $\widetilde{\mathcal{R}}_b$ is fused with \mathcal{F}_m and then fed into two consecutive 3×3 convolutions.

Mask \rightarrow Boundary (M2B) Fusion: Mask features \mathcal{F}_m contain rich high-level information, *i.e.*, the pixel-wise object category information, which is beneficial to predict object boundaries. Hence, we propose a simple fusion block to integrate boundary features and mask features for boundary prediction. The fusion block can be formulated as follows.

$$\mathcal{F}_b = f(\mathcal{F}_m) + \widetilde{\mathcal{R}}_b, \quad (1)$$

where \mathcal{F}_b denotes the boundary features and f means a 1×1 convolution.

Boundary \rightarrow Mask (B2M) Fusion: We fuse the final boundary features with mask features; thus, boundary information can be used to enrich mask features and guide precise mask prediction. The fusion block is the same as that of M2B.

3.3 Learning and Optimization

Following the common practice in edge detection [49,52], we regard boundary prediction as a pixel-level classification problem. The learned boundary features are fused with mask features to provide shape information for mask prediction.

Boundary Groundtruths: We use the Laplacian operator to generate soft boundaries from the binary mask groundtruths. The Laplacian operator is a second-order gradient operator and can produce thin boundaries. The produced boundaries are converted into binary maps by a threshold 0 as the final groundtruths.

Boundary Loss: Most boundary or edge detection methods [49,52,1] take the advantage of weighted cross-entropy to alleviate the class-imbalance problem in edge/boundary prediction. However, weighted binary cross-entropy leads to thick and coarse boundaries [16]. Following [16], we use dice loss [39] and binary cross-entropy to optimize the boundary learning. Dice loss measures the

overlap between predictions and groundtruths and is insensitive to the number of foreground/background pixels, thus alleviating the class-imbalance problem. Our boundary loss \mathcal{L}_b is formulated as follows.

$$\mathcal{L}_b(p_b, y_b) = \mathcal{L}_{Dice}(p_b, y_b) + \lambda \mathcal{L}_{BCE}(p_b, y_b), \quad (2)$$

where $p_b \in \mathbb{R}^{H \times W}$ denotes the predicted boundary for a particular category and $y_b \in \mathbb{R}^{H \times W}$ denotes the corresponding boundary groundtruth. H and W are height and width of the predicted boundary map respectively. λ is a hyper-parameter to adjust the weight of dice loss (We set $\lambda = 1$ in all experiments). Dice loss is given as follows.

$$\mathcal{L}_{Dice}(p_b, y_b) = 1 - \frac{2 \sum_i^{H \times W} p_b^i y_b^i + \epsilon}{\sum_i^{H \times W} (p_b^i)^2 + \sum_i^{H \times W} (y_b^i)^2 + \epsilon}, \quad (3)$$

where i denotes the i -th pixel and ϵ is a smooth term to avoid zero division (We set $\epsilon = 1$). In ablation experiments, we will analyze and evaluate different loss functions with quantitative results and qualitative results.

Multi-Task Learning: Multi-task learning has been proved effective in many works [21, 14, 45, 27, 40], which achieves better performance for different tasks comparing with separate training. Since boundary and mask are crossed linked by two fusion blocks, jointly training can enhance the feature representation for both boundary and mask. We define a multi-task loss for each sample as follows.

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{box} + \mathcal{L}_{mask} + \mathcal{L}_b, \quad (4)$$

where the classification loss \mathcal{L}_{cls} , regression loss \mathcal{L}_{box} , and Mask loss \mathcal{L}_{mask} are inherited from Mask R-CNN. The boundary loss \mathcal{L}_b has been introduced in detail in Equation (2).

4 Experiments

We perform extensive experiments on the challenging COCO dataset [35] and the Cityscapes dataset [12] to demonstrate the effectiveness of Boundary-preserving Mask R-CNN. To better understanding each component of our method, we provide detailed ablation experiments on COCO.

Dataset and Metrics: COCO contains 115k images for training, 5k images for validation and 20k images for testing. Our models are trained on the training set (*train2017*). We report the results on the validation set (*val2017*) for ablation studies and the results on testing set (*test-dev2017*) to compare with other methods. The Cityscapes dataset is collected in urban scenes which contains 2975 training, 500 validation and 1525 testing images. As for instance segmentation, Cityscapes involves 8 object categories and provides more precise instance-level

segmentation annotations than COCO. We train our models on the training set and report our performance on the validation set and the testing set. For both COCO and Cityscapes, we use the same evaluation metric (*i.e.*, COCO AP), which is the average precision over different IoU thresholds (from 0.5 to 0.95).

Implementation: We adopt Mask R-CNN [38] as our baseline and our method is developed based on it. All hyper-parameters are kept the same. Unless specified, we use ResNet-50 with FPN as our backbone network. We initialize our backbone networks with ImageNet pre-trained weights and freeze all BN [26] layers. The input images are resized such that the shorter side is 800 pixels and the longer is less than 1333 pixels. As for ablation experiments, we adopt 600 pixels for shorter side (the longer is less than 1000 pixels). Following the standard practice, we train all models on 4 NVIDIA GPUs using Synchronized SGD with initial learning rate 0.02 and 16 images per mini-batch for 90,000 iterations and reduce the learning rate by a factor of 0.1 and 0.01 after 60,000 and 80,000 iterations respectively. For larger backbones, we follow the linear scaling rule [19] to adjust the learning schedule when decreasing batch size.

4.1 Overall Results

We first evaluate our BMask R-CNN with different backbones on COCO and compare it with Mask R-CNN. As shown in Table 1, our method outperforms Mask R-CNN by remarkable APs in spite of different backbones. Compared with Mask R-CNN, BMask R-CNN significantly achieves 1.4, 1.7 and 1.5 AP improvements using ResNet-50-FPN, ResNet-101-FPN and HRNetV2-W32-FPN [48] respectively. Exploiting boundary information contributes to more precise mask localization due to the observation that our method yields noteworthy and stable improvements (≈ 2.3 AP) on AP_{75} . AP^b shows AP for bounding box, on which BMask R-CNN very slightly improves over Mask R-CNN.

In Table 2, we compare BMask R-CNN with some state-of-the-art instance segmentation methods. All models are trained on COCO *train2017* and evaluated on COCO *test-dev2017*. Without bells and whistles, BMask R-CNN with ResNet-101-FPN can surpass these methods.

Fig. 1 illustrates the AP curves of BMask R-CNN and Mask R-CNN under different IoU thresholds. Note that our method obtains larger gain when the IoU threshold increases, showing better localization performance of BMask R-CNN.

4.2 Ablation Experiments

In order to comprehend how BMask R-CNN works, we perform exhaustive experiments to analyze the components in BMask R-CNN. Table 3 shows the results of gradually adding components to the Mask R-CNN baseline. Each component of our proposed BMask R-CNN will be investigated in the following sections.

Table 1: Comparison with Mask R-CNN on COCO *val2017*

Method	Backbone	AP	AP ₅₀	AP ₇₅	AP ^b	AP ₅₀ ^b	AP ₇₅ ^b
Mask R-CNN	ResNet-50-FPN	34.2	56.0	36.3	37.8	59.2	41.1
BMask R-CNN	ResNet-50-FPN	35.6	56.3	38.4	37.8	59.0	41.5
Mask R-CNN	ResNet-101-FPN	36.1	58.1	38.3	40.1	61.7	44.0
BMask R-CNN	ResNet-101-FPN	37.8	59.1	40.6	40.4	62.0	44.3
Mask R-CNN	HRNetV2-W32-FPN	36.6	58.7	38.9	40.8	61.9	44.9
BMask R-CNN	HRNetV2-W32-FPN	38.1	59.4	40.7	41.0	61.9	45.1

Table 2: Comparison with state-of-the-art methods for instance segmentation on COCO *test-dev2017*(* denotes our implementation)

Method	Backbone	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
MNC [14]	ResNet-101	24.6	44.3	24.8	4.7	25.9	43.6
FCIS+++ [33]	ResNet-101	33.6	54.5	-	-	-	-
Mask R-CNN [21]	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN [21]	ResNeXt-101-FPN	37.1	60.0	39.4	16.9	39.9	53.5
MaskLab [9]	ResNet-101-FPN	35.4	57.4	37.4	16.9	38.3	49.2
MaskLab+ [9]	ResNet-101-FPN	37.3	59.8	39.6	19.1	40.5	50.6
Mask Scoring R-CNN [24]	ResNet-50-FPN	35.8	56.5	38.4	16.2	37.4	51.0
Mask Scoring R-CNN [24]	ResNet-101-FPN	37.5	58.7	40.2	17.2	39.5	53.0
CondInst [46]	ResNet-50-FPN	35.4	56.4	37.6	18.4	37.9	46.9
BlendMask [6]	ResNet-50-FPN	34.3	55.4	36.6	14.9	36.4	48.9
PointRend [30]	ResNet-50-FPN	36.3	-	-	-	-	-
Mask R-CNN*	ResNet-50-FPN	34.6	56.5	36.6	15.4	36.3	49.7
BMask R-CNN	ResNet-50-FPN	35.9	57.0	38.6	15.8	37.6	52.2
Mask R-CNN*	ResNet-101-FPN	36.2	58.6	38.4	16.4	38.4	52.1
BMask R-CNN	ResNet-101-FPN	37.7	59.3	40.6	16.8	39.9	54.6
BMask R-CNN w/ Mask Scoring	ResNet-101-FPN	38.7	59.1	41.9	17.4	40.7	55.5

Effects of Boundaries: To validate the effect of boundaries for mask prediction, we use mask targets to replace boundary targets and also evaluate the performance without boundary supervision and loss with the architecture kept the same. Table 4 indicates that boundary supervision with our proposed boundary-preserving mask head improves mask results by 0.8 and 0.7 AP compared with mask supervision and no supervision respectively. Notably, using boundary can improve the mask localization performance (AP₇₅) by a significant margin.

RoI Feature Extraction: Compared with mask prediction, predicting boundaries requires more precise spatial information due to boundaries are spatially sparse. Therefore, we explore several strategies and present two considerations to extract better RoI features for boundaries. The first aspect is the source of RoI features. Lin *et al.* [34] propose that RoI features are extracted from the different levels ($P2 \sim P5$) in FPN depending on the scales of the corresponding

Table 3: Experiment results on COCO *val2017* of adding components to Mask R-CNN. We gradually add boundary supervision (BCE loss), fusions between mask and boundary features, BCE-Dice loss and our RoI feature extraction strategy for boundary features.

Boundary	Fusions	BCE-Dice	RoI Strategy	AP	AP ₅₀	AP ₇₅	AP ^b
-	-	-	-	33.2	54.4	34.9	36.6
✓				33.9	54.8	35.8	36.7
✓	✓			34.2	55.4	36.4	36.8
✓	✓	✓		34.4	55.0	36.6	36.7
✓	✓	✓	✓	34.7	55.1	37.2	36.8

Table 4: Experiment results on COCO *val2017* of changing groundtruth of the boundary head. ✕ denotes no supervision on the boundary head.

Groundtruth	AP	AP ₅₀	AP ₇₅	AP ^b
✕	34.0	55.0	36.0	36.3
mask	33.9	54.3	35.9	36.0
boundary	34.7	55.1	37.2	36.8

proposals. Features of high levels in FPN lacks spatial information which are inappropriate for boundaries. Fig. 5 illustrates different sources for mask features \mathcal{R}_m and boundary features \mathcal{R}_b . Fig. 5(a) shows that boundary features are directly extracted from $P2$ while mask features are from from $P2 \sim P5$ according to the scale of the proposal. Fig. 5(b) shows that both boundary features and mask features are extracted from the same feature level from $P2 \sim P5$. The other aspect is the feature resolution. Higher-resolution features preserve more spatial information which is beneficial to boundaries. Therefore, we explore the effects of 28×28 resolution and 14×14 resolution RoI features for learning boundaries. As Table 5 shows, directly extracting boundary RoI features from $P2$ is more effective with larger resolution. We employ boundary features extracted from $P2$ with 28×28 resolution in other experiments.

Feature Fusion: In Section 3.2, we have emphasized the relation between boundary features and mask features. Fusion blocks in our boundary-preserving mask head build explicit links to enrich both feature representation. Table 6 shows more results: if there is no fusion, it improves Mask R-CNN by 0.5 AP, which is the gain of multi-task learning; with both M2B and B2M fusion blocks, BMask R-CNN has 1.5 AP improvement over Mask R-CNN. We further investigate the influence of adding more subsequent fusion blocks. Keeping the overall computation cost substantially unchanged, adding more B2M or M2B fusions brings negligible improvements.

Table 5: Experiment results on COCO *val2017* for different RoI feature extraction strategies.

Source	Size	AP	AP ₅₀	AP ₇₅	AP ^b
P2 ~ P5	14	34.4	55.1	36.5	36.8
P2	14	34.5	55.0	36.7	36.7
P2 ~ P5	28	34.4	55.1	36.8	36.6
P2	28	34.7	55.1	37.2	36.8

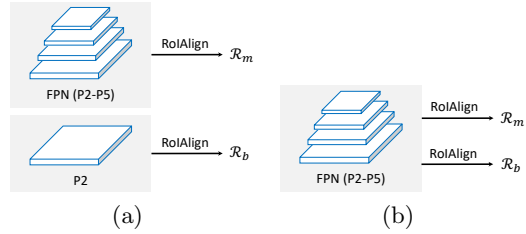


Fig. 5: Different RoI feature extraction strategies for Boundary-preserving Mask Head.

Table 6: Experiment results on COCO *val2017* for the impacts of fusion blocks, *i.e.*, mask \rightarrow boundary (M2B) fusion and boundary \rightarrow mask (B2M) fusion.

M2B Fusion	B2M Fusion	AP	AP ₅₀	AP ₇₅	AP ^b
✗	✗	33.7	55.0	35.7	36.8
✓	✗	34.2	54.9	36.5	36.8
✗	✓	33.9	54.7	36.1	36.6
✓	✓	34.7	55.1	37.2	36.8

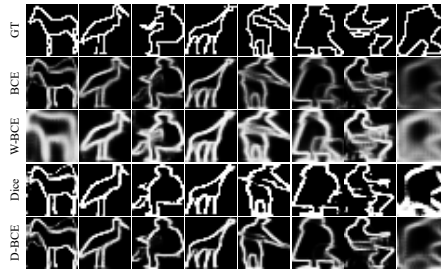
Loss Functions: We evaluate the impacts of different loss functions for optimizing boundary learning. Table 7 shows that the combination of BCE and Dice loss leads to better performance compared with individual BCE or Dice loss. Weighted BCE brings less gain than BCE in boundary prediction.

To investigate how the Dice-BCE combined loss provides such competitive improvements, we present detailed analysis on the visualization results of these experiments. As shown in Fig. 6, different loss functions have different impacts on learning boundaries. BCE loss provides considerably precisely-localized but unclear boundaries due to the class-imbalance problem. Weighted BCE solves this problem by applying balancing weights but this hard balancing leads to thick and coarse boundaries which exceed their corresponding masks. Dice loss also solves the class-imbalance problem without thick boundaries but lacks precise localization. Consequently, combining Dice loss and BCE can provide better-localized boundaries and avoid the class-imbalance problem.

Computation Cost: Compared with Mask R-CNN, our method involves four 3×3 and two 1×1 convolutional layers for boundary prediction and two fusion blocks which increase the computation cost. To clarify the improvements of BMask R-CNN are not from extra computation cost, we form a larger mask head by adding 4 more 3×3 convolutional layers as a comparison. Table 8 shows that BMask R-CNN still achieves a significant gain compared with Mask R-CNN with equal computation cost.

Table 7: Experiment results on COCO *val2017* for evaluating different loss functions.

Loss type	AP	AP ₅₀	AP ₇₅	AP ^b
BCE	34.3	54.9	36.3	36.7
Weighted BCE	34.1	55.0	36.2	36.7
Dice	34.3	55.0	36.5	36.5
Dice-BCE	34.7	55.1	37.2	36.8

Fig. 6: Visualization results for analyzing the impacts of different loss functions. **GT** denotes groundtruth. **W-BCE** and **D-BCE** denote Weighted BCE and Dice-BCE respectively.Table 8: Experiment results on COCO *val2017* for evaluating the impacts of computation cost. MRCNN and LMH denote Mask R-CNN and the larger mask head respectively. FLOPs are counted only for mask head without the final predictors. Inference time is tested on one NVIDIA RTX 2080Ti with the input size 600*1000

Method	FLOPs	Time(ms/img.)	AP	AP ₅₀	AP ₇₅	AP ^b
MRCNN	0.46G	59.0	33.2	54.4	34.9	36.6
MRCNN w/ LMH	0.93G	61.3	33.7	54.6	35.8	36.7
BMask R-CNN	0.95G	63.7	34.7	55.1	37.2	36.8

4.3 Experiments on Cityscapes

To further explore the effects of BMask R-CNN on the fine-annotated Cityscapes dataset, we only use images with **fine** annotations to train and evaluate our models. For fair comparisons, we use ResNet-50-FPN as our backbone and resize images with shorter edge randomly selected from [800, 1024] for training. For inference, input images are kept the original size 1024×2048. Models are trained by SGD on 4 GPUs with mini-batch size 4 for 48,000 iterations. The learning rate is 0.005 at the beginning and reduced to 0.0005 after 36,000 iterations. Other settings are the same with experiments on COCO.

We report the results evaluated on Cityscapes **val** and **test** in Table 9. BMask R-CNN achieves 29.4 AP on **test** and obtains a remarkable 2.2 AP gain compared with the baseline Mask R-CNN. BMask R-CNN outperforms previous methods without extra data.

4.4 Discussions

Coarse boundary annotation vs. precise boundary prediction: When datasets become larger and larger, obtaining precise mask annotations is un-

Table 9: Experiment results on Cityscapes **val** (AP[**val**]) and **test**.(* denotes our implementation)

	AP [val]	AP ₇₅ [val]	AP	AP ₅₀	person	rider	car	truck	bus	train	mcycle	bicycle
BAIS [20]	-	-	17.4	36.7	-	-	-	-	-	-	-	-
DIN [2]	-	-	20.0	38.8	16.5	16.7	25.7	20.6	30.0	23.4	17.1	10.1
SGN [36]	29.2	-	25.0	44.9	21.8	20.1	39.4	24.8	33.2	30.8	17.7	12.4
Mask R-CNN [21]	31.5	-	26.2	49.9	30.5	23.7	46.9	22.8	32.2	18.6	19.1	16.0
BshapeNet [28]	-	-	27.1	50.3	29.6	23.3	46.8	25.8	32.9	24.6	20.3	14.0
BshapeNet+ [28]	-	-	27.3	50.5	30.7	23.4	47.2	26.1	33.3	24.8	21.5	14.1
Neven <i>et al.</i> [41]	-	-	27.6	50.9	34.5	26.1	52.4	21.7	31.2	16.4	20.1	18.9
Mask R-CNN*	32.0	30.1	27.2	53.0	31.4	23.7	49.1	22.9	33.7	21.9	19.4	15.4
BMask R-CNN	35.0	33.6	29.4	54.7	34.3	25.6	52.6	24.2	35.1	24.5	21.4	17.1



Fig. 7: **(a):** Qualitative comparison between COCO *val* annotations (left) and our instance boundary predictions (right). **(b): Sobel Mask Head:** We use Sobel operator to obtain 2-channel boundary features indicate X direction and Y direction and then apply two 3×3 convolutions to output boundary predictions.

avoidably time-consuming. Though the COCO dataset provides abundant instance-level annotations, the mask and boundary annotations (represented by sparse polygons) are coarse, which limits the performance of our method BMask R-CNN. Nevertheless, BMask R-CNN can output more precise and smooth boundaries with fewer mask overlap between instances; some selected examples are shown in Fig. 7(a).

Sobel mask head: Instead of predicting boundaries using an extra branch, we also design a simple Sobel mask head to predict boundaries from masks, which is a improved version of [55]. As illustrated in Fig. 7(b), it has a Sobel operator [31] and two 3×3 convolutions following the mask predictions. We adopt the same Dice-BCE loss function for training. Using ResNet-50-FPN backbone and keep the rest settings the same, this Sobel mask head method obtains 34.0 AP which improves Mask R-CNN by 0.8 AP but is 0.7 AP worse than our main method.

4.5 Qualitative Results

We provide representative visualization results on COCO to compare our method with Mask R-CNN and further prove the effectiveness of our method. Fig. 8(a) shows the qualitative results on COCO *val*. Mask R-CNN is more prone to generate masks with coarse boundaries which contain much background along with



Fig. 8: **(a)**: Qualitative results on COCO dataset generated by Mask R-CNN and BMask R-CNN with ResNet-101-FPN. MRCNN and BMRCNN denotes Mask R-CNN and BMask R-CNN respectively. **(b)**: Raw mask prediction and boundary prediction from boundary-preserving mask head. GT: the groundtruth segmentation. MRCNN: mask predicted by Mask R-CNN. BMRCNN: mask predicted by BMask R-CNN. Boundary: boundary predicted by BMask R-CNN. Results are obtained with ResNet-101-FPN backbone.

some false positive areas. Our proposed BMask R-CNN can alleviate this issue with the help of preserving boundaries. We further visualize our raw boundary and mask results in Fig. 8(b). It can be easily observed that predicted masks are more clear and highly coincident with their boundaries. Furthermore, utilizing predicted boundaries to refine masks brings minor improvement and the refinement is vulnerable to the noises.

5 Conclusion

We address the issue that coarse boundaries and imprecise localization in instance segmentation and propose a novel Boundary-preserving Mask R-CNN. It incorporates boundary information to guide the mask learning for better boundaries and localization. Our experiments demonstrate that our method achieves remarkable and stable improvements on both COCO and Cityscapes especially in terms of localization performance. Extensive studies and visualization results provide a deep understanding of how our method BMask R-CNN works. Our method could also be plugged into Cascade Mask R-CNN and *etc.* for higher performance. We hope it can be a strong baseline and sheds light on this fundamental research topic.

Acknowledgements

This work was in part supported by NSFC (No. 61733007 and No. 61876212), Zhejiang Lab (No. 2019NB0AB02), and HUST-Horizon Computer Vision Research Center.

References

1. Acuna, D., Kar, A., Fidler, S.: Devil is in the edges: Learning semantic boundaries from noisy annotations. In: CVPR. pp. 11075–11083 (2019)
2. Arnab, A., Torr, P.H.S.: Pixelwise instance segmentation with a dynamically instantiated network. In: CVPR. pp. 879–888 (2017)
3. Bertasius, G., Shi, J., Torresani, L.: Semantic segmentation with boundary neural fields. In: CVPR. pp. 3602–3610 (2016)
4. Brabandere, B.D., Neven, D., Gool, L.V.: Semantic instance segmentation with a discriminative loss function. CoRR **abs/1708.02551** (2017)
5. Cai, Z., Vasconcelos, N.: Cascade R-CNN: high quality object detection and instance segmentation. CoRR **abs/1906.09756** (2019)
6. Chen, H., Sun, K., Tian, Z., Shen, C., Huang, Y., Yan, Y.: Blendmask: Top-down meets bottom-up for instance segmentation. In: CVPR (2020)
7. Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: Hybrid task cascade for instance segmentation. In: CVPR. pp. 4974–4983 (2019)
8. Chen, L., Barron, J.T., Papandreou, G., Murphy, K., Yuille, A.L.: Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In: CVPR. pp. 4545–4554 (2016)
9. Chen, L., Hermans, A., Papandreou, G., Schroff, F., Wang, P., Adam, H.: Masklab: Instance segmentation by refining object detection with semantic and direction features. In: CVPR. pp. 4013–4022 (2018)
10. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2018)
11. Cheng, F., Chen, C., Wang, Y., Shi, H., Cao, Y., Tu, D., Zhang, C., Xu, Y.: Learning directional feature maps for cardiac mri segmentation. In: MICCAI 2020 (2019)
12. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
13. Dai, J., He, K., Li, Y., Ren, S., Sun, J.: Instance-sensitive fully convolutional networks. In: ECCV. pp. 534–549 (2016)
14. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: CVPR. pp. 3150–3158 (2016)
15. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: NIPS. pp. 379–387 (2016)
16. Deng, R., Shen, C., Liu, S., Wang, H., Liu, X.: Learning to predict crisp boundaries. In: ECCV (2018)
17. Fathi, A., Wojna, Z., Rathod, V., Wang, P., Song, H.O., Guadarrama, S., Murphy, K.P.: Semantic instance segmentation via deep metric learning. CoRR **abs/1703.10277** (2017)
18. Girshick, R.B.: Fast R-CNN. In: ICCV. pp. 1440–1448 (2015)
19. Goyal, P., Dollár, P., Girshick, R.B., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., He, K.: Accurate, large minibatch SGD: training imagenet in 1 hour. CoRR **abs/1706.02677** (2017)
20. Hayder, Z., He, X., Salzmann, M.: Boundary-aware instance segmentation. In: CVPR. pp. 587–595 (2017)
21. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: ICCV (2017)

22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
23. Huang, Q., Xia, C., Zheng, W., Song, Y., Xu, H., Kuo, C.J.: Object boundary guided semantic segmentation. In: ACCV. pp. 197–212 (2016)
24. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X.: Mask scoring R-CNN. In: CVPR. pp. 6409–6418 (2019)
25. Huang, Z., Wang, X., Wei, Y., Huang, L., Shi, H., Liu, W., Huang, T.S.: Cnet: Criss-cross attention for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* pp. 1–1 (2020). <https://doi.org/10.1109/TPAMI.2020.3007032>
26. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML. pp. 448–456 (2015)
27. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: CVPR. pp. 7482–7491 (2018)
28. Kim, H.Y., Kang, B.R.: Instance segmentation and object detection with bounding shape masks. *CoRR* **abs/1810.10327** (2018)
29. Kirillov, A., Levinkov, E., Andres, B., Savchynskyy, B., Rother, C.: Instancecut: From edges to instances with multicut. In: CVPR. pp. 7322–7331 (2017)
30. Kirillov, A., Wu, Y., He, K., Girshick, R.: Pointrend: Image segmentation as rendering. In: CVPR (2020)
31. Kittler, J.: On the accuracy of the sobel edge detector. *Image and Vision Computing* **1**(1), 37–42 (1983)
32. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: NIPS. pp. 109–117 (2011)
33. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: CVPR. pp. 4438–4446 (2017)
34. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR (2017)
35. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: ECCV (2014)
36. Liu, S., Jia, J., Fidler, S., Urtasun, R.: SGN: sequential grouping networks for instance segmentation. In: ICCV. pp. 3516–3524 (2017)
37. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: CVPR. pp. 8759–8768 (2018)
38. Massa, F., Girshick, R.: maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. <https://github.com/facebookresearch/maskrcnn-benchmark> (2018)
39. Milletari, F., Navab, N., Ahmadi, S.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 3DV. pp. 565–571 (2016)
40. Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: CVPR. pp. 3994–4003 (2016)
41. Neven, D., Brabandere, B.D., Proesmans, M., Gool, L.V.: Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In: CVPR. pp. 8837–8845 (2019)
42. Pinheiro, P.H.O., Collobert, R., Dollár, P.: Learning to segment object candidates. In: NIPS. pp. 1990–1998 (2015)
43. Pinheiro, P.O., Lin, T., Collobert, R., Dollár, P.: Learning to refine object segments. In: ECCV. pp. 75–91. Springer (2016), https://doi.org/10.1007/978-3-319-46448-0_5

44. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1137–1149 (2017)
45. Takikawa, T., Acuna, D., Jampani, V., Fidler, S.: Gated-scnn: Gated shape cnns for semantic segmentation. *ICCV* (2019)
46. Tian, Z., Shen, C., Chen, H.: Conditional convolutions for instance segmentation. *ArXiv abs/2003.05664* (2020)
47. Uhrig, J., Cordts, M., Franke, U., Brox, T.: Pixel-level encoding and depth layering for instance-level semantic labeling. In: *GCPR*. pp. 14–25 (2016)
48. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., Xiao, B.: Deep high-resolution representation learning for visual recognition. *CoRR abs/1908.07919* (2019)
49. Xie, S., Tu, Z.: Holistically-nested edge detection. *International Journal of Computer Vision* **125**(1-3), 3–18 (2017)
50. Xu, W., Parmar, G., Tu, Z.: Learning geometry-aware skeleton detection. In: *BMVC* (2019)
51. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Learning a discriminative feature network for semantic segmentation. In: *CVPR*. pp. 1857–1866 (2018)
52. Yu, Z., Feng, C., Liu, M., Ramalingam, S.: Casenet: Deep category-aware semantic edge detection. In: *CVPR*. pp. 1761–1770 (2017)
53. Yu, Z., Liu, W., Zou, Y., Feng, C., Ramalingam, S., Kumar, B.V.K.V., Kautz, J.: Simultaneous edge alignment and learning. In: *ECCV*. pp. 400–417 (2018)
54. Yuan, Y., Xie, J., Chen, X., Wang, J.: Segfix: Model-agnostic boundary refinement for segmentation (2020)
55. Zimmermann, R.S., Siems, J.N.: Faster training of mask R-CNN by focusing on instance boundaries. *CoRR abs/1809.07069* (2018)