International Journal of Computer Vision (2022) 130:2571–2589
https://doi.org/10.1007/s11263-022-01662-0

Improving Image Segmentation with Boundary Patch Refinement

Xiaolin Hu[1,2] · Chufeng Tang[1] · Hang Chen[1] · Xiao Li[1] · Jianmin Li[1] · Zhaoxiang Zhang[3]

Received: 4 August 2021 / Accepted: 18 July 2022 / Published online: 12 August 2022

Abstract
Tremendous efforts have been made on image segmentation but the mask quality is still not satisfactory. The boundaries of predicted masks are usually imprecise due to the low spatial resolution of feature maps and the imbalance problem caused by the extremely low proportion of boundary pixels. To address these issues, we propose a conceptually simple yet effective post-processing refinement framework, termed BPR, to improve the boundary quality of the prediction of any image segmentation model. Following the idea of looking closer to segment boundaries better, we extract and refine a series of small boundary patches along the predicted boundaries. The refinement is accomplished by a boundary patch refinement network at the higher resolution. The trained BPR model can be easily transferred to refine the results of other models as well. Extensive experiments show that the proposed BPR framework yields significant improvements on the semantic, instance, and panoptic segmentation tasks over a variety of baselines on the Cityscapes dataset.

Keywords Image Segmentation · Boundary Refinement · Instance Segmentation · Semantic Segmentation · Panoptic Segmentation

1 Introduction

Image segmentation, which aims to recognize an image at the pixel level, is a fundamental yet challenging problem in computer vision. Image segmentation can be formulated as different tasks (Minaee et al., 2021a). Semantic segmentation (Fig. 1b) aims to assign each pixel a category label (*e.g.*, for uncountable stuff classes). Instance segmentation (Fig. 1c) aims to assign each object a pixel-wise instance mask and a category label (*e.g.*, for countable things). Panoptic segmentation (Fig. 1d) unifies semantic and instance segmentation, which aims to assign each pixel a semantic label and an unique instance identity (for both stuff and things).

Numerous algorithms have been developed for image segmentation in the literature. However, the quality of segmentation results is still not satisfactory. One of the most important problems is the imprecise segmentation around category or instance boundaries. Taking instance segmentation as an example (Fig. 2 left), the predicted instance masks of the prevailing Mask R-CNN (He et al., 2017) method are coarse and not well-aligned with the real object boundaries. Empirically, correcting the error pixels near boundaries can improve the mask quality a lot. We conducted an analysis for instance segmentation (Table 1). A large gain (9.4/14.2/17.8 in AP) were obtained by simply replacing the predictions with ground-truth labels for pixels within a certain Euclidean distance (1px/2px/3px) to the predicted instance boundaries, especially for small objects. Note that, in the last row, the mask AP is not 100% since only the segmentation errors were corrected, while the classification and detection errors remained. Similarly, we believe that correcting error pixels for semantic segmentation and panoptic segmentation could also improve the performance, though the performance gain may not be as large as we observed for instance segmentation since their evaluation metrics (e.g., mIoU and PQ) are relatively less sensitive to boundary quality.

There are two critical issues leading to low-quality boundary segmentation. (1) The low spatial resolution of the output,

Communicated by O. Veksler.

✉ Xiaolin Hu
xlhu@mail.tsinghua.edu.cn

[1] The State Key Laboratory of Intelligent Technology and Systems, BNRist, THU-Bosch JCML Center, Institute for Artificial Intelligence, Department of Computer Science and Technology, Tsinghua University, Beijing, China

[2] Chinese Institute for Brain Research (CIBR), Beijing, China

[3] The Institute of Automation, Chinese Academy of Sciences, Centre for Artificial Intelligence and Robotics, HKISI-CAS, University of Chinese Academy of Sciences, Beijing, China

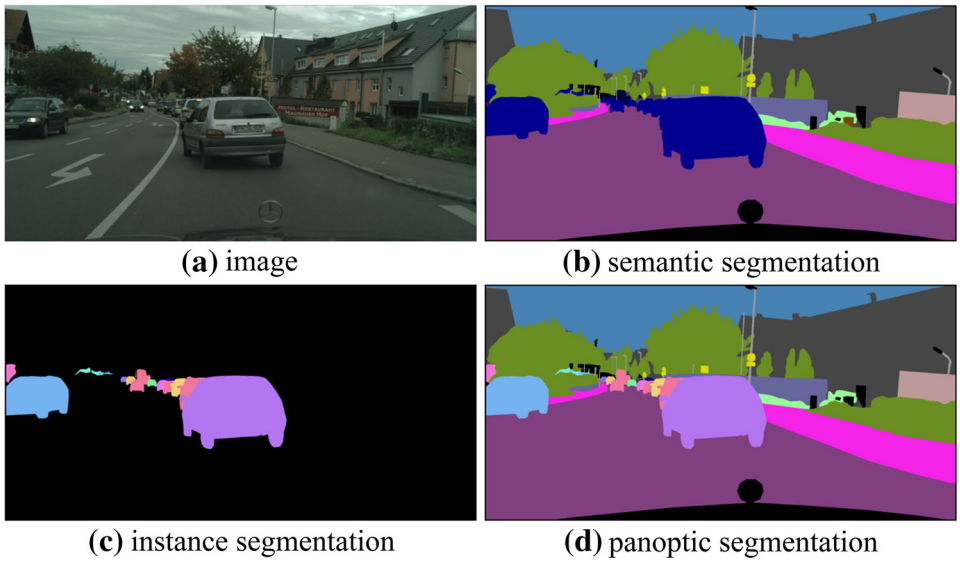

Fig. 1 Illustration of different image segmentation tasks. **a** An image from Cityscapes. **b** Semantic segmentation: assign each pixel a category label. The same color indicates the same category. **c** Instance segmentation: assign each instance a category label and a mask. The same color indicates the same instance. **d** Panoptic segmentation: assign each pixel a category label and a instance label. It is a combination of semantic segmentation and instance segmentation. Best viewed in color (Color figure online)

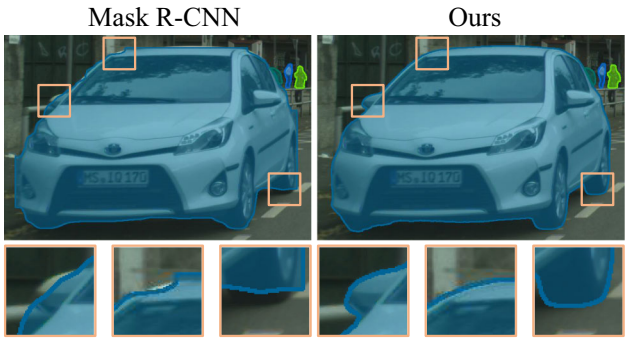

Fig. 2 Left: Instance segmentation results (top) and the zoomed-in boundary patches (bottom) of Mask R-CNN. Right: After the refinement of our BPR model, the instance mask aligns better with object boundaries (Color figure online)

e.g., 28×28 in Mask R-CNN (He et al., 2017) or at most 1/4 of the input resolution in some image segmentation models (Tian et al., 2020; Wang et al., 2020d), makes finer details around object boundaries disappear. The predicted boundaries are always coarse and imprecise (see Fig. 2). (2) Pixels around object boundaries only make up a small fraction of the whole image (less than 1% (Kirillov et al., 2017)), and are inherently hard to classify. Treating all pixels equally may leads to an optimization bias towards smooth interior areas, while underestimating the boundary pixels.

As a long-standing challenge in image segmentation tasks, many studies have attempted to improve the boundary quality (see Sect. 2.2), while the above issues are still not well solved. For example, BMask R-CNN (Chen et al., 2020b), Gated-SCNN (Takikawa et al., 2019), Focal-BG (Wang et al., 2019) and HMEDN (Zhou et al., 2019) employ an additional branch to enhance the boundary awareness of mask features, which can fix the optimization bias to some extent, but the low resolution issue remains unsolved. PolyTransform (Liang et al., 2020) and SegFix (Yuan et al., 2020b) act as a post-processing scheme to improve the boundary quality. PolyTransform (Liang et al., 2020) employs a deforming network with the cropped instance patch to predict the offsets of polygon vertices, while suffers from a large computational overhead. SegFix (Yuan et al., 2020b) replaces the coarse predictions of boundary pixels with interior predictions, but it relies on precise boundary predictions. The instance boundary prediction task may share a similar complexity with instance segmentation (Xie et al., 2020; Xu et al., 2019).

Consider the human annotation behavior for image segmentation. The annotators usually first localize and categorize each area or object in the given image, and then explicitly or implicitly segment some coarse masks at a low resolution. Afterwards, to obtain a high-quality mask, the annotators need to repeatedly zoom in the local boundary regions and explore the sharper boundary segmentation at higher resolution. Intuitively, high-level semantics are required to localize and roughly segment objects, while low-level details (*e.g.*, color consistency and contrast) are more critical for segmenting the local boundary regions.

Motivated by the human segmentation behavior, we propose a conceptually simple yet effective post-processing framework to improve the boundary quality through a *crop-then-refine* strategy. Specifically, given a coarse mask produced by any image segmentation model, we first extract a series of small image patches along the predicted boundaries. After concatenated with the corresponding mask patches, the boundary patches are fed into a refinement network, which performs binary segmentation to refine the coarse boundaries. The refined mask patches are then reassembled into a compact and high-quality mask, shown in Fig. 2 (right). We termed the proposed framework as **BPR** (**B**oundary

Patch **R**efinement). The proposed framework can alleviate the aforementioned issues and improve the mask quality without any modification or fine-tuning to the segmentation models. Since we only crop around object or category boundaries, the patches are allowed to be processed with much higher resolution than previous methods, so that low-level details can be retained better. Concurrently, the fraction of boundary pixels in the small patch is naturally increased, which can alleviate the optimization bias. In addition, the trained BPR model becomes model-agnostic, which can be easily transferred to refine the results of other segmentation models as well, without the need for re-training.

We applied the proposed BPR framework to the three segmentation tasks mentioned above and achieved consistent improvements over a variety of baseline methods on the Cityscapes dataset. Specifically, we improved 6.1% AP over the Mask R-CNN (He et al., 2017) instance segmentation baseline, 2.6% mIoU over the HRNet (Wang et al., 2020b) semantic segmentation baseline, and 2.7% PQ over the UPSNet (Xiong et al., 2019) panoptic segmentation baseline on the Cityscapes val set. Visualization results show that our BPR model produced precise and clear boundaries.

Some preliminary results have been presented in a conference paper (Tang et al., 2021), which focuses on instance segmentation only. We provide more results and analysis in this paper, including

(1) more results about the effect of the refinement network and the effect of different training data sources,
(2) the improved results for instance segmentation through the change of refinement network and training data source,
(3) detailed comparison with the other boundary refinement methods with the same baseline model,
(4) comparison with some newly published methods, and more importantly,
(5) extension of this method to boundary refinement for semantic and panoptic segmentation with detailed results and analysis.

2 Related Work

2.1 Image Segmentation

As a post-processing scheme, our proposed BPR framework can be applied to refine the results of any image segmentation model. Since we will demonstrate the effectiveness of our proposed framework on three image segmentation tasks, including instance, semantic, and panoptic segmentation, we here briefly review the methods for solving these tasks.

Table 1 Results of replacing the predictions for pixels within a certain Euclidean distance to the predicted boundaries with their ground-truth labels

Dist.	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
–	36.4	60.8	36.9	11.1	32.4	57.3
1px	45.8	64.8	49.3	21.1	42.6	63.5
2px	50.6	66.5	54.6	26.3	47.0	66.8
3px	54.2	67.5	58.5	30.4	50.7	69.3
∞	70.4	70.4	70.4	41.5	66.7	88.3

∞ means all error pixels were corrected. Experiments were conducted with Mask R-CNN as baseline on Cityscapes val set. The definition of the metrics can be found in Sect. 4.1

(1) *Instance Segmentation* Two-stage instance segmentation methods usually follow the classical *detect-then-segment* strategy. The dominant method is Mask R-CNN (He et al., 2017), which inherits from Faster R-CNN (Ren et al., 2015) to first detect objects then performs binary segmentation within each predicted bounding box. Several extensions of Mask R-CNN have demonstrated better results (Liu et al., 2018; Huang et al., 2019). Some *One-stage* methods (Bolya et al., 2019; Chen et al., 2020a; Ying et al., 2019) also adapt the *detect-then-segment* strategy but replace the detectors with the one-stage alternatives (Lin et al., 2017; Tian et al., 2019). Some recent methods, such as CondInst (Tian et al., 2020) and SOLO (Wang et al., 2020d, c), eliminate the need for detection and segment objects in a location-wise manner.
(2) *Semantic Segmentation* Modern semantic segmentation approaches are pioneered by the fully convolutional network (FCN) (Long et al., 2015). Many methods have been proposed based on this method to improve the segmentation results, such as increasing the resolution of feature maps with dilated/atrous convolutions (Chen et al., 2017, 2018), enriching context information (Yuan et al., 2020a; Fu et al., 2019; Zhao et al., 2017), using an encoder-decoder architecture (Chen et al., 2018; Milletari et al., 2016; Ronneberger et al., 2015), or some refinement schemes (Yuan et al., 2020b; Krähenbühl & Koltun, 2011). See (Minaee et al., 2021b) for a comprehensive review of these approaches.
(3) *Panoptic Segmentation* Kirillov et al. (Kirillov et al., 2019b) first formulated the task of panoptic segmentation. *Top-down* methods usually append a semantic segmentation model or branch to the instance segmentation models (*e.g.*, Mask R-CNN) and further fuse the semantic and instance results. Heuristic post-processing steps (Kirillov et al., 2019b, a; de Geus et al., 2018; Li et al., 2019) or specialized fusion modules (Xiong et al., 2019; Lazarow et al., 2020; Ren et al., 2021; Mohan & Valada, 2020) are used to resolve the inherent overlapping problem. *Bottom-up* methods typically start with

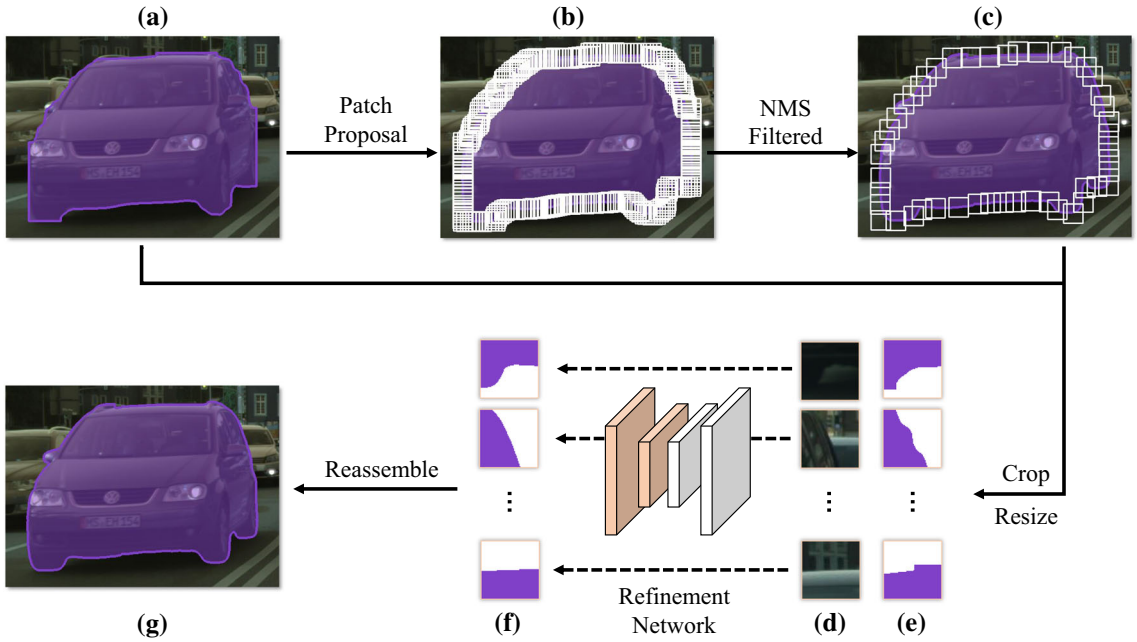

Fig. 3 Overview of the proposed boundary patch refinement framework. Given a coarse mask (**a**) produced by an image segmentation model (taking instance segmentation as an example), we first densely assign a series of squared bounding boxes along the predicted boundaries (**b**), and filter out a subset of boundary patches (**c**) using NMS. The extracted image patches (**d**) and mask patches (**e**) are resized and fed into the boundary patch refinement network. Mask patches after refinement (**f**) are reassembled into a compact and precise segmentation result (**g**). Best viewed in color (Color figure online)

semantic segmentation and then cluster 'thing' pixels into instances. These methods include SSAP (Gao et al., 2019) and DeepLab-based series works (Yang et al., 2019; Chen et al., 2020a; Wang et al., 2020a, 2021).

2.2 Relation to Existing Boundary Refinement Methods

Most recent studies focusing on boundary refinement aim at designing a boundary-aware segmentation model by integrating an extra and specialized module (or loss function) to process boundaries. For example, BMask R-CNN (Chen et al., 2020b), Gated-SCNN (Takikawa et al., 2019), DecoupleSegNets (Li et al., 2020), Focal-BG (Wang et al., 2019) and HMEDN (Zhou et al., 2019) employ an extra branch to enhance the boundary awareness of mask features by estimating boundaries directly. CGBNet (Ding et al., 2020) employs a boundary delineation refinement module to obtain better boundaries in semantic segmentation. PointRend (Kirillov et al., 2020) iteratively samples the feature points with unreliable predictions and refines them with a shared MLP. IABL (Wang et al., 2022) and ContourLoss (Chen et al., 2020) use boundary-aware loss functions to improve the boundary quality. B2Inst (Kim et al., 2021) uses the extra boundary basis, and RefineMask (Zhang et al., 2021) propose a multi-stage boundary refinement module to improve the boundary quality for instance segmentation. These specialized modules usually process boundaries in a low resolution due to the GPU memory constraint, thus the low-resolution issue remains unsolved. By contrast, the proposed BPR method only processes the smaller boundary patches, thus the patches could be up-sampled into larger scales when inputted into the refinement network.

Another line of work attempts to refine boundaries based on the results of existing segmentation models with a post-processing scheme, without any structure modification (extra branch) or fine-tuning strategy (extra loss) to the segmentation models. SegFix (Yuan et al., 2020b) performs refinement by replacing the unreliable predictions of boundary pixels with the predictions of interior pixels. The effectiveness of SegFix highly depends on the accuracy of boundary prediction, while it is challenging to directly estimate precise instance boundaries. PolyTransform (Liang et al., 2020) transforms the contour of an instance into a set of polygon vertices. A Transformer (Vaswani et al., 2017) based network is applied to predict the offsets of vertices towards object boundaries. It achieves superior performance while suffers from a large computational overhead due to the heavy Transformer architecture and the processing of the instance-level patches. The proposed BPR method also worked in a post-processing manner. Different from above mentioned methods, we focus on refining the boundary patches to

improve the mask quality. In addition, all of the above mentioned methods are proposed for instance or semantic segmentation. To the best of our knowledge, no prior work has focused on panoptic segmentation. In this paper, we present a general boundary refinement method for all three image segmentation tasks.

In the field of boundary detection, DeepStrip (Zhou et al., 2020) proposes to convert the boundary regions into a strip image and compute a boundary prediction in the strip domain. Our work is similar to DeepStrip in the high-level spirit of looking closer to the boundary region to refine the predictions. However, these two methods are significantly different in design. Firstly, DeepStrip predicts the boundary pixels directly while BPR learns to predict foreground pixels. BPR is substantially easier to optimize since the proportions of foreground and background pixels are roughly the same in the boundary patches. Secondly, DeepStrip requires an 80×4096 strip image as the input, while BPR processes squared image patches (*e.g.*, 64×64) and requires the corresponding mask patches as the input. Thirdly, the pipeline of the proposed BPR is more concise than DeepStrip, which adopts a series of carefully designed operations and loss functions. By contrast, BPR is simple and straightforward in design.

3 Framework

In this section, we take instance segmentation as an example to explain the details of the proposed boundary refinement framework. We then show that the framework can be extented to refine the results of semantic segmentation and panoptic segmentation. An overview of the proposed framework is illustrated in Fig. 3.

3.1 Boundary Patch Extraction

Given an instance mask produced by an instance segmentation model, we first need to determine which part of the mask should be refined. Based on the findings of previous works (Chen et al., 2020b; Yuan et al., 2020b) and our verification experiments in Table 1, we propose an *sliding-window* algorithm to extract a series of patches along the predicted instance boundaries. Specifically, we densely assign a group of squared bounding boxes where the central areas of the box cover the boundary pixels as shown in Fig. 3b. The obtained boxes contain large overlaps and redundancies, thus we apply a Non-Maximum Suppression (NMS) algorithm to filter out a subset of patches (Fig. 3c). Empirically, we found that larger overlaps boosted performance but incurred higher computational cost. We can adjust the NMS threshold to control the amount of overlap to achieve a better speed/accuracy trade-off. In addition to image patches, we also extract the corresponding binary mask patches from the given coarse instance mask. The concatenated image and mask patches (Fig. 3d and e) are resized and fed into the refinement network.

3.2 Boundary Patch Refinement

(1) *Mask Patch* The benefit of the binary mask patch is that it accelerates training convergence and provides location guidance for the instance to be segmented. As discussed in previous works on semantic segmentation (Wang et al., 2020b; Yuan et al., 2020a), context information plays a vital role in pixel-wise classification. Therefore, the cropped image patches are hard to be classified independently due to the limited context information. With the help of location and semantic information provided by the mask patches, the refinement network can eliminate the need for learning instance-level semantics from scratch. Instead, the refinement network only needs to learn how to locate the hard pixels around the decision boundary and push them to the correct side. We believe this goal can be achieved by exploring low-level image properties (*e.g.*, color consistency and contrast) provided in the local and high-resolution image patches. More importantly, the adjacent instances are likely to share an identical boundary patch, while the learning goals are totally different and ambiguous. Together with different mask patches for each instance, these issues can be avoided (see Sect. 4.2 for experimental results).

(2) *Refinement Network* The role of this refinement network is to perform binary segmentation in extracted boundary patches individually. Most prevailing semantic segmentation models can be used here to perform binary segmentation by simply modifying the input channels to 4 (3 for the RGB channels and 1 for the binary mask) and output classes to 2 (distinguish foreground and background). Before inputting into the refinement network, both the image patches and mask patches were upsampled into a larger scale with bilinear interpolation. By increasing the input size, the boundary patches could be processed with much higher resolution than in existing methods to address the low-resolution issue mentioned above.

(3) *Reassembling* The refined boundary patches are reassembled into a compact instance-level mask by replacing their previous predictions. Predictions are unchanged for those pixels without refinement. For the overlapping areas of adjacent patches, the results are aggregated by simply averaging the output logits and applying a threshold of 0.5 to distinguish the foreground and background.

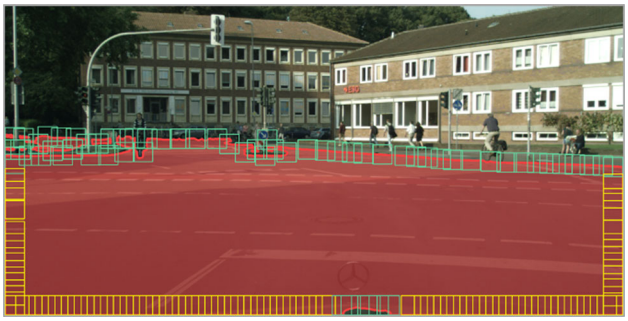

Fig. 4 Patch extraction for semantic masks (taking *road* as an example). Yellow boxes are removed during training for semantic and panoptic segmentation. Best viewed in color (Color figure online)

3.3 Extension to Semantic and Panoptic Segmentation

For semantic segmentation, the model usually outputs a fixed number (*e.g.*, 19 classes in the Cityscapes dataset) of semantic masks and each one corresponds to a semantic category. For panoptic segmentation, the model output is the combination of instance masks and semantic masks. Ideally, treating each semantic mask as a special "instance" mask, the boundary patches can be extracted and processed with the same manner as instance segmentation (see Fig. 4). One problem is that patches extracted near the image border are usually entirely filled by foreground pixels (yellow boxes in Fig. 4). These inferior patches have a negligible contribution for training since no effective boundaries are included. They could degrade the model performance (see Sect. 5.2 for experimental results). We remove these inferior patches from the training patch set when processing the semantic and panoptic segmentation results. For instance segmentation, objects seldomly meet image border thus this issue can be neglected. In addition, for semantic and panoptic segmentation, image patches are more likely to be shared by adjacent semantic masks than for instance segmentation, thus mask patches play a more important role (see Sect. 5.2 for experimental results). Another problem is that one pixel can have multiple predictions for instance segmentation (instances can have overlaps), but every pixel should only be assigned a single category (or a instance) label for semantic (or panoptic) segmentation. Therefore, we need an additional step to ensure the uniqueness of mask predictions after patch reassembling. We take a simple strategy. For pixels with more than one predictions, we keep the semantic (or instance) label with the maximum confidence. With above modifications, the instance, semantic, and panoptic segmentation tasks can be solved in the same framework.

3.4 Learning and Inference

The refinement network is trained based on the boundary patches extracted from training images and results. As a post-processing mechanism, the proposed framework can be applied to refine the results of any image segmentation model, without any modification or fine-tuning to the segmentation models themselves. For instance segmentation, we only extract boundary patches from instances whose predicted masks have an Intersection over Union (IoU) overlap larger than 0.5 with the ground-truth masks during training, while all predicted instances are retained during inference. The model outputs are supervised with the corresponding ground-truth mask patches using the pixel-wise binary cross-entropy loss.

4 Experiments: Instance Segmentation

4.1 Datasets, Metrics, and Implementation Details

(1) *Datasets* We mainly report the results on Cityscapes (Cordts et al., 2016), a real-world dataset with high-quality instance segmentation annotations. We only used the `fine` data, containing 2,975/500/1,525 images for train/val/test, which were collected from 27 cities, with a high resolution of 1024×2048 pixels. There are eight instance categories, including bicycle, bus, person, train, truck, motorcycle, car, and rider.

(2) *Metrics* The COCO-style (Lin et al., 2014) mask AP (averaged over 10 IoU thresholds ranging from 0.5 to 0.95 in the step of 0.05), $AP_{50}/AP_{75}/AP_{90}$ (AP at an IoU of 0.5/0.75/0.9 respectively) and $AP_S/AP_M/AP_L$ (for small/medium/large instances) were reported in most of our experiments. The official Cityscapes-style AP (Cordts et al., 2016) was only used to report the final results for a fair comparison, which was slightly higher than the COCO-style AP. Similar to (Takikawa et al., 2019; Liang et al., 2020; Yuan et al., 2020b), we also used a boundary F-score to evaluate the quality of the predicted boundaries. A mask was considered correct if the boundary was within a certain distance threshold from the ground-truth. We used a threshold of one pixel and only computed for true positives that were determined on the same 10 IoU thresholds ranging from 0.5 to 0.95. The boundary F-score was computed in an instance-wise manner and then averaged over them, termed AF. In addition to the AP and boundary F-score (AF) metrics, we further evaluated the performance with a newly proposed metric designed for measuring the boundary quality, boundary AP (Chen et al., 2021) (AP^b for short), to demonstrate effectiveness of the proposed BPR method for boundary refinement. Boundary AP is

calculated base on boundary IoU, which calculates the IoU for mask pixels within a certain distance from the corresponding ground truth or prediction boundary contours.

(3) *Implementation Details* The `MMSegmentation` (MMSegmentation, 2020) codebase was adopted to implement the boundary patch refinement network. During training, the image patches were augmented by random horizontal flipping and random photometric distortion. The binary mask patches were normalized with the mean and standard deviation both equal to 0.5. We used the SGD optimizer with the initial learning rate 0.01, the momentum 0.9, and the weight decay 0.0005. The learning rate was decayed using the poly learning rate policy with the power of 0.9. The models were trained for 160K iterations with a batch size of 32 on 4 GPUs and syncBN (Zhang et al., 2018). To have an impression about the training speed, we take the default setting adopted in ablation studies (see below) as an example. We extracted 280k/67k patches from the train/val results of Mask R-CNN (adopted from `MMDetection` (Chen et al., 2019)). It took about 10 hours for training on 4 NVIDIA RTX 2080Ti GPUs under this setting.

4.2 Ablation Study

We investigated the effectiveness of the proposed framework through extensive ablation experiments on the configurable design choices. We started the refinement with the results of Mask R-CNN ResNet-FPN-50 baseline trained on Cityscapes `fine` data (with COCO pre-training). We adopted the lightweight HRNetV2-W18-Small (abbreviated as HRNet-W18s in the following) as the refinement network in the default setting, with input size equal to 128×128 pixels. The boundary patches were extracted with patch size equal to 64×64 pixels without padding, and the inference NMS threshold was set to 0.25 by default. Note that the ablation experiments were conducted mainly on instance segmentation, but the results can be generalized to semantic and panoptic segmentation.

(1) Effect of Mask Patches To validate the effect of the mask patches for boundary refinement, we made a comparison by eliminating the mask patches while keeping other settings unchanged. As indicated in Table 2, the model trained with image patches solely yielded terrible results (20.1% AP), even much worse than the segmentation results before refinement (36.4% AP). However, together with mask patches, we achieved a significant improvement (+3.4% in AP, +4.0% in AP^b, +11.9% in AF) by refining the Mask R-CNN segmentation results. A larger gain in AP_{90} (+4.6%), which requires better localization, indicates that the box quality was improved. The remarkable results on the boundary-sensitive AF metric suggest that the AP gains mainly benefited from the improvement of boundary quality. We show some patch-wise examples in Fig. 5. When there was one dominant instance in the image patch (the first row), both models (w/ and w/o mask patches) produced good results. However, when there were multiple instances crowded in the image patch, the model without mask patches (the last column) failed to distinguish which object should be segmented, leading to coarse (the 2nd row) or completely wrong (the 3rd and 4th rows) predictions. In contrast, with the help of mask patches, the model produced high-quality predictions with accurate and clear boundaries (the 3rd column).

(2) Effect of Patch Size We increased the boundary patch size by cropping a larger box with or without padding. Note that the padded areas were only used to enrich the context and not used for reassembling. As the patch size gets larger, the model becomes less focused but can access more context

Table 2 Results with and without mask patch input to the refinement network

w/ mask	AP	AP_{90}	AP^b	AF	AP_S	AP_M	AP_L
–	36.4	11.4	33.9	54.9	11.1	32.4	57.3
✘	20.1	3.8	16.8	57.2	4.0	14.7	36.3
✔	**39.8**	**16.0**	**37.9**	**66.8**	**12.7**	**35.9**	**62.2**

Best results are given in bold
'–' indicates the results of Mask R-CNN before refinement

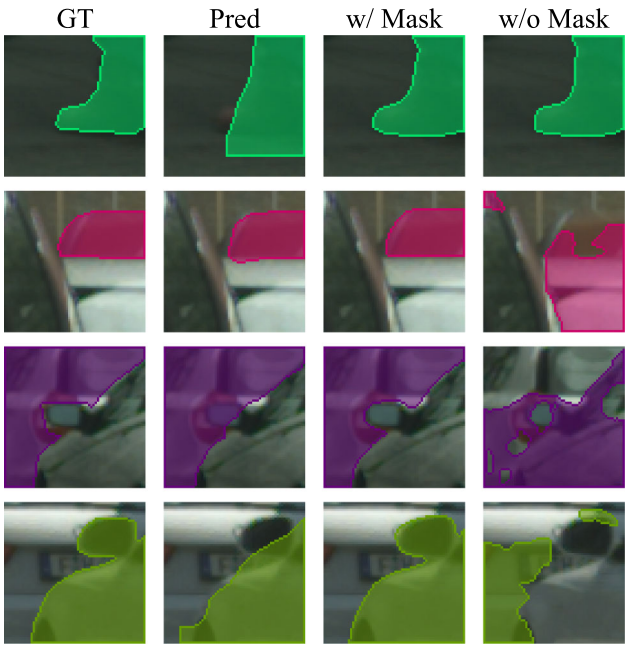

Fig. 5 Boundary patch examples of (from left to right): ground-truth, predictions of Mask R-CNN, results refined by our proposed framework, results refined without the use of mask patch. The mask patch plays a crucial role in our framework, resulting in high-quality boundaries (the 3rd column) (Color figure online)

Table 3 Results with different patch size

Scale/pad	AP	AP_{90}	AP^b	AF	AP_S	AP_M	AP_L
–	36.4	11.4	33.9	54.9	11.1	32.4	57.3
32 / 0	39.4	14.8	37.4	66.8	12.6	35.6	61.4
32 / 5	39.7	15.2	**38.0**	**67.6**	**12.9**	**35.9**	61.6
64 / 0	**39.8**	**16.0**	37.9	66.8	12.7	**35.9**	62.2
64 / 5	39.7	14.9	**38.0**	66.5	12.5	35.8	62.1
96 / 0	39.6	**16.0**	37.5	65.7	12.2	35.4	**62.3**

Best results are given in bold

information. We compared various choices and found that the 64×64 patch without padding worked the best (Table 3). We used this setting in all other experiments.

(3) Effect of Patch Extraction Scheme The most important contribution of this work is the idea of looking closer at instance boundaries to achieve better segmentation results. There are multiple choices about how to extract the boundary patches for refinement. We compared three extraction schemes as shown in Fig. 6. The most straightforward scheme is to pre-define a grid and divide the input image into small patches (Fig. 6b). Then we select patches that cover boundary pixels as boundary patches for refinement. Another scheme is to extract the *instance-level patch* (Fig. 6c) based on the detected bounding box and further re-segment the instance patch, similar to previous studies (Liang et al., 2020; Liu et al., 2020). This scheme can be viewed as an improved Mask R-CNN equipped with a stand-alone mask head. It does not solve the optimization bias issue and the learning process is dominated by interior pixels. Experimental results are listed in Table 4. For the *pre-defined grid* scheme, we varied the patch size and found the results were consistently worse than our proposed "dense sampling + NMS filtering" scheme. The results were improved slightly by enabling padding but still sub-optimal. One of the most important reasons is the imbalanced foreground/background ratio. We observed that some extracted patches were almost entirely filled with either foreground or background pixels (yellow dashed boxes in Fig. 6b). These patches were hard to refine due to the lack of context. In contrast, by restricting the center of patches to cover the boundary pixels (Fig. 6a), the imbalance problem can be alleviated. For the *instance-level patch* scheme, even the patch size was enlarged to 512×512 pixels, the results were still sub-optimal.

(4) Effect of Input Size of the Refinement Network The extracted boundary patches were upsampled into a larger scale before refinement. Table 5 shows the impact of input size. We also report the approximate inference speed of the refinement network, with a fixed batch size of 135 (on average 135 patches per image). As the input size increased, the AP and AF scores increased accordingly, and slightly dropped after 256. This is reasonable as more details are retained with

Table 4 Results with different patch extraction schemes

Scheme	Scale/pad	AP	AP_{90}	AP^b	AF
–	–	36.4	11.4	33.9	54.9
Dense. + NMS	64 / 0	**39.8**	**16.0**	**37.9**	**66.8**
Pre-defined grid	32 / 5	39.3	14.6	37.3	65.8
Pre-defined grid	64 / 0	38.7	14.2	37.0	65.2
Pre-defined grid	64 / 5	39.1	14.8	37.3	65.6
Pre-defined grid	64 / 10	39.2	14.7	37.5	65.6
Pre-defined grid	96 / 5	38.8	14.7	36.8	63.7
Instance-level patch	256 / 0	37.5	12.3	35.3	61.5
instance-level patch	512 / 0	38.7	15.2	36.6	63.8

Best results are given in bold

Table 5 Results with different sizes of input to the refinement network

Size	FPS	AP	AP_{90}	AP^b	AF	AP_S	AP_M	AP_L
–	–	36.4	11.4	33.9	54.9	11.1	32.4	57.3
64	17.5	39.1	15.6	37.1	64.9	11.8	35.1	61.6
128	9.4	39.8	**16.0**	37.9	66.8	12.7	**35.9**	62.2
256	4.1	**40.0**	15.7	**38.0**	**67.0**	**12.8**	**35.9**	**62.5**
512	<2	39.7	15.0	37.9	66.9	12.7	35.7	61.9

Best results are given in bold

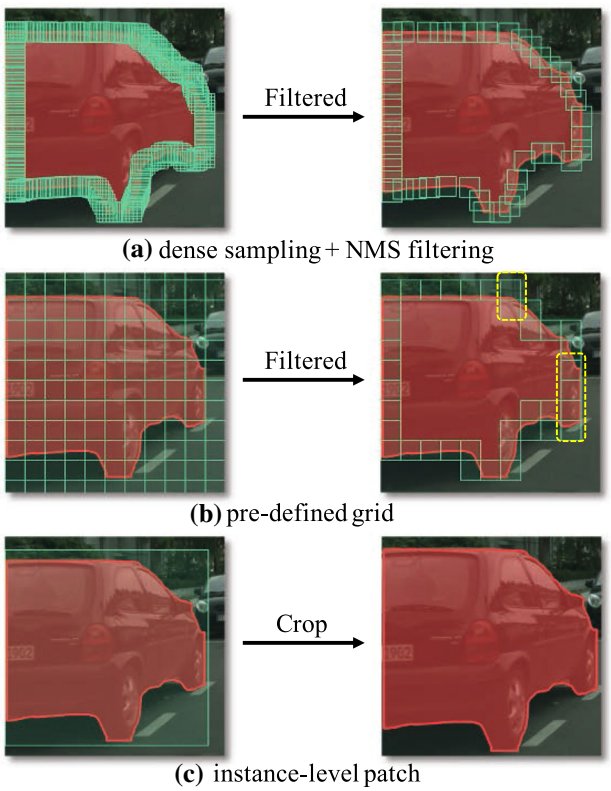

Fig. 6 Illustration of different patch extraction schemes (Color figure online)

Table 6 Results with different refinement networks

Network	FPS	AP	AP_{90}	AP^b	AF
–	–	36.4	11.4	33.9	54.9
FCN-HRNet-W18s	9.4	39.8	16.0	37.9	66.8
FCN-HRNet-W18	5.8	39.8	16.0	37.9	66.8
FCN-HRNet-W48	2.5	40.1	16.2	**38.3**	67.0
FCN-ResNet-18	15.7	39.7	15.6	37.9	66.4
FCN-ResNet-50	4.5	**40.3**	15.9	**38.3**	67.3
DeepLabV3Plus-R50	3.4	40.2	**16.8**	**38.3**	**67.6**
Fast-SCNN	**35.3**	38.6	12.0	36.7	63.2

Best results are given in bold

Table 7 Results with different NMS eliminating thresholds

Thr.	#patch/img	FPS	AP	AP_{90}	AP^b	AF
–	–	–	36.4	11.4	33.9	54.9
0	32	24.0	37.7	12.0	35.2	58.7
0.15	103	12.4	39.6	15.8	37.6	66.0
0.25	135	9.4	39.8	16.0	37.9	66.8
0.35	178	7.3	39.9	15.8	37.9	67.0
0.45	241	5.2	40.0	15.8	38.0	67.0
0.55	332	3.8	**40.1**	**16.2**	38.2	67.1
0.65	485	2.5	**40.1**	15.9	**38.3**	**67.2**

Best results are given in bold
The average number of patches per image is also listed

larger input size, but very large input would diverge the network's focus.

(5) Effect of the Refinement Network Most semantic segmentation model can be used to perform binary segmentation for boundary patches. We compared the performance of different segmentation networks (Table 6). The performance was robust to the choice of refinement network. A stronger backbone usually led to higher performance, but at the expense of lower speed. Even with some lightweight networks (*e.g.*, HRNet-W18s, ResNet-18, Fast-SCNN (Poudel et al., 2019)), we still achieved nontrivial improvements. We adopted FCN-ResNet-50 in the final model. Since the model essentially performs binary segmentation for patches, it can further benefit from the advances in semantic segmentation, such as increasing the resolution of feature maps (Wang et al., 2020b; Chen et al., 2017, 2018).

(6) Effect of NMS Eliminating Threshold We studied the impact of different NMS eliminating thresholds during inference, shown in Table 7. The reported "#patch/img" indicates the total number of patches for all instances in an image. There might be several instances per image, and the number of patches also varied across different instances (*e.g.*, larger instances may produce more boundary patches) in the image. As the threshold got larger, the number of boundary patches increased rapidly. The overlap of adjacent patches provides a chance to correct unreliable predictions of the inferior patches. As shown, the resulting boundary quality was consistently improved with a larger threshold, and saturated around 0.55. We fixed the NMS eliminating threshold to 0.25 during training. During inference, 0.25 was used in ablation experiments (Sect. 4.2) and 0.55 was used in stronger models (Sects. 4.3 and 4.4).

4.3 Transferability

What the BPR model learned is a general ability to correct error pixels around instance boundaries. We can easily transfer this *ability of boundary refinement* to refine the results of any instance segmentation model. After training, the BPR model becomes model-agnostic, similar to SegFix (Yuan et al., 2020b). Specifically, once we get a model trained on the boundary patches extracted from the predictions of Mask R-CNN on Cityscapes, we can make inference to refine the prediction of any model (not only Mask R-CNN) on the same dataset, without retraining the model. We validated the transferability by applying the model trained on Mask R-CNN results to refine the predictions of PointRend (Kirillov et al., 2020) and SegFix (Yuan et al., 2020b). Note that these two methods are also designed to improve boundary quality in segmentation. As shown in Table 8, the transferred model still improved the results of PointRend and SegFix by a large margin, suggesting that our method is compatible with them.

In these experiments, BPR model was trained on Mask R-CNN (w/ COCO pre-training) predictions. At the same time, choosing which model's predictions to train the BPR model is also worth considering. We trained several BPR models on different segmentation results and applied them to refine the result of Mask R-CNN (Table 9). We adopted FCN-ResNet-50 as the refinement network with NMS threshold equal to 0.55, other settings were the same as ablation experiments above. We found that training the BPR model with the predictions of Mask R-CNN without COCO pre-training worked the best. The reason for training the BPR model with the predictions of Mask R-CNN (w/o COCO pre-training) was better than training it with the predictions of Mask R-CNN (w/ COCO pre-training) might be that the boundaries of the weaker Mask R-CNN (w/o COCO pre-training) were usually coarser, thus provide more diverse boundary patches for training the BPR model.

4.4 Overall Results

(1)*Comparison with the State-of-the-art Methods* We adopted the optimal design choices and hyperparameters found in ablation experiments to train a stronger BPR model. Specifically, we adopted FCN-ResNet-50 as our refinement network, with 256×256 input patches resized from 64×64, and

Table 8 Results of transferring BPR to Other Models

	AP	AP_{90}	AP^b	AF
PointRend	35.6	11.3	33.1	58.0
w/ BPR†	**38.6**	**14.3**	**36.4**	**66.5**
Mask R-CNN + SegFix	38.2	12.7	35.6	63.2
w/ BPR†	**40.0**	**15.4**	**38.2**	**67.0**

Best results are given in bold
BPR† was trained on the results of Mask R-CNN

Table 9 Results of BPR models trained on different segmentation results

Source of training data	AP	AP_{90}	AP^b	AF
–	36.4	11.4	33.9	54.9
PointRend	39.4	14.7	37.4	65.7
SegFix	39.5	15.6	37.6	64.7
Mask R-CNN (w/ COCO)	40.5	16.2	38.6	67.7
Mask R-CNN (w/o COCO)	**40.8**	**16.7**	**38.9**	**68.2**

Best results are given in bold
w/ and w/o COCO indicate Mask R-CNN with and without the COCO pre-training

a NMS threshold of 0.55 during inference. The BPR model here was trained on the results of Mask R-CNN (w/o COCO pre-training). The model was evaluated on Cityscapes `val` and `test` sets and compared against some state-of-the-art methods, including DWT (Bai & Urtasun, 2017), SGN (Liu et al., 2017), Mask R-CNN (He et al., 2017), BMask R-CNN (Chen et al., 2020b), AdaptIS (Sofiiuk et al., 2019), PANet (Liu et al., 2018), SSAP (Gao et al., 2019), UPSNet (Xiong et al., 2019), PANet (Liu et al., 2018) (Table 10). We had the following observations. (1) Compared with the Mask R-CNN baseline, we achieved a significant improvement (+4.5% and +4.6% AP on `val` and `test` sets). Our BPR outperformed SegFix (Yuan et al., 2020b) by a large margin, which is also a boundary refinement module applied to the same baseline. Applying our BPR model to the results already refined by SegFix led to even better results (slightly lower than applying BPR only). (2) By applying BPR to the strong PolyTransform (Liang et al., 2020) baseline (1^{st} place at CVPR 2020). Our "PolyTransform + BPR" consistently improved 2.6% AP on the Cityscapes `test` set and also outperformed "PolyTransform + SegFix" (2^{nd} place at ECCV 2020) by a large margin (+1.5%). (3) By applying BPR to the stronger "PolyTrans-

Table 10 Instance segmentation results on Cityscapes val (AP [`val`] column) and test (other columns) sets

	training data	AP [`val`]	AP	AP_{50}	Person	Rider	Car	Truck	Bus	Train	Mcycle	Bicycle
DWT	`fine`	21.2	19.4	35.3	15.5	14.1	31.5	22.5	27.0	22.9	13.9	8.0
SGN	`fine` + `coarse`	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	`fine`	31.5	26.2	49.9	30.5	23.7	46.9	22.8	32.2	18.6	19.1	16.0
BMask R-CNN	`fine`	35.0	29.4	54.7	34.3	25.6	52.6	24.2	35.1	24.5	21.4	17.1
AdaptIS	`fine`	36.3	32.5	52.5	31.4	29.1	50.0	31.6	41.7	39.4	24.7	12.1
PANet	`fine`	36.5	31.8	57.1	36.8	30.4	54.8	27.0	36.3	25.5	22.6	20.8
SSAP	`fine`	37.3	32.7	51.8	35.4	25.5	55.9	33.2	43.9	31.9	19.5	16.2
UPSNet	`fine` + COCO	37.8	33.0	59.7	35.9	27.4	51.9	31.8	43.1	31.4	23.8	19.1
PANet	`fine` + COCO	41.4	36.4	63.1	41.5	33.6	58.2	31.8	45.3	28.7	28.2	24.1
RefineMask	`fine`	37.6	32.0	56.6	37.4	29.3	55.6	26.6	36.5	26.6	23.4	20.8
+ BPR		40.8	35.2	58.0	41.6	33.3	60.5	29.5	40.1	28.0	25.5	23.4
Mask R-CNN	`fine` + COCO	36.4	32.0	58.1	34.8	27.0	49.1	30.1	40.9	30.9	24.1	18.7
Mask R-CNN*		36.8	32.6	59.2	36.7	29.2	52.8	30.0	40.3	27.9	25.0	19.0
+ SegFix§		38.2	33.3	57.8	37.9	30.3	54.1	31.0	40.0	27.9	25.1	20.5
+ BPR‡		41.1	36.9	61.0	42.0	33.3	59.9	32.9	44.4	32.6	28.0	22.3
+ BPR		**41.3**	**37.2**	**61.3**	**42.5**	**33.8**	**60.3**	**33.5**	**44.9**	31.2	**28.3**	**22.8**
+ SegFix + BPR		41.0	36.8	59.8	41.0	32.8	58.7	32.9	43.1	**36.8**	26.5	22.2
PolyTransform	`fine` + COCO	44.6	40.1	65.9	42.4	34.8	58.5	39.8	50.0	41.3	30.9	23.4
+ SegFix		-	41.2	66.1	44.3	35.9	60.5	40.5	51.2	41.6	31.7	24.1
+ BPR‡		**46.9**	42.4	**66.6**	45.6	36.7	62.4	41.2	52.3	43.4	32.7	**25.2**
+ BPR		**46.9**	42.7	66.5	45.9	**37.2**	62.7	**41.6**	52.5	**43.8**	**32.9**	25.1
+ SegFix + BPR		-	**42.8**	66.5	**46.1**	**37.2**	**62.9**	41.5	**52.7**	43.7	32.7	**25.2**

Best results are given in bold
BPR denotes our proposed model, which was trained on the results of Mask R-CNN (w/o COCO). Mask R-CNN* was implemented by us. SegFix§ results were reported in the original paper (Yuan et al., 2020b), which used a slightly different Mask R-CNN baseline (36.5/32.0 in AP val/test). BPR‡ is our conference version (Tang et al., 2021) model

Table 11 Results by applying BPR to Mask2Former (Chen et al., 2022) (measured on Cityscapes `val`), which is a latest query-based method for instance segmentation

	AP	AP_{90}	AP^b	AF
Mask2Former ResNet101	38.5	15.9	36.1	58.8
+ BPR	**41.2**	**19.1**	**38.8**	**67.8**
Mask2Former Swin-L	43.7	16.8	41.1	60.4
+ BPR	**45.6**	**19.3**	**42.9**	**68.4**

Best results are given in bold

form + SegFix" (Yuan et al., 2020b) baseline, we achieved state-of-the-art results on the Cityscapes `test` set with AP of 42.8%. (4) Our BPR improved the results of RefineMask (Zhang et al., 2021) by a large margin (+3.2% AP), which is a newly published method focusing on boundary refinement for instance segmentation. (5) The results of the proposed BPR model were a little bit better than those reported in our conference paper (Tang et al., 2021), due to the change of refinement network (from HRNet-W48 to FCN-ResNet-50) and training data source (from Mask R-CNN w/ COCO pre-training to Mask R-CNN w/o COCO pre-training).

We further applied the proposed BPR method to refine the results of Mask2Former (Chen et al., 2022), which is a recently proposed query-based method for instance segmentation and achieved remarkable performance on the popular benchmarks. As shown in Table 11, the proposed BPR model successfully improved the results of Mask2Former on Cityscapes `val`, even with the powerful Swin-L (Liu et al., 2021) backbone. Note that the best-performing model (45.6%) still lagged behind "PolyTransform + BPR" (46.9%) since COCO pre-training was not used in Mask2Former (Cityscapes `fine` data only).

(2)*Comparison with Similar Methods* Several previous methods also focus on boundary refinement for segmentation, such as BMask R-CNN (Chen et al., 2020b), PointRend (Kirillov et al., 2020), and SegFix (Yuan et al., 2020b). SegFix and our BPR are model-agnostic. BMask R-CNN and PointRend add or replace the head of Mask R-CNN, and the original papers only report the results based on the Mask R-CNN (w/o COCO) baseline. We further compared with these methods under the same baseline. As shown in Table 12, our BPR method remarkably improved the baseline results (+6.1% and +5.8% AP on val and test sets respectively), and outperformed these similar approaches by large margins.

(3)*Qualitative Results* We show some qualitative results on Cityscapes `val` in Fig. 7a. Compared with the coarse predictions of Mask R-CNN, our BPR generated substantially better segmentation results with precise and clear boundaries. It largely alleviated the over-smoothing issues (Kirillov et al., 2020) in previous methods caused by the low resolution feature maps.

Table 12 Comparison with similar methods based on the Mask R-CNN (w/o COCO pre-training) baseline on Cityscapes. $*$ indicates our implementation. The Mask R-CNN baselines in BMask R-CNN[§] (Chen et al., 2020b) and PointRend[§] (Kirillov et al., 2020) are slightly different from our Mask R-CNN*

	AP [`val`]	AP [`test`]
Mask R-CNN*	32.6	29.0
Mask R-CNN* + SegFix*	34.4 (+1.8)	30.9 (+1.9)
BMask R-CNN[§]	35.0 (+2.4)	29.4 (+0.4)
PointRend[§]	35.8 (+3.2)	-
Mask R-CNN* + BPR	**38.7 (+6.1)**	**34.8 (+5.8)**

Best results are given in bold

Table 13 Results on COCO val2017. $AP^\star$ is measured on the higher-quality LVIS (Gupta et al., 2019) annotations. Mask R-CNN ResNeXt-FPN-101 was used as the baseline model

w/ BPR	AP	AP^b	AF	$AP^\star$	$AP^\star_S$	$AP^\star_M$	$AP^\star_L$
	38.4	24.0	54.5	40.4	24.5	48.3	57.2
✔	**39.2**	**26.0**	**58.4**	**42.1**	**24.8**	**50.3**	**60.4**

Best results are given in bold

(4)*Speed* The inference time of our proposed framework is independent of the original instance segmentation models, which consists of three parts: patch extraction, refinement, and reassembling. Note that only the refinement part was considered when we calculated the FPS in Tables 5, 6, and 7. Besides, the FPS was measured in an imprecise manner by fixing the batch size to 135 (average number of patches per image), while the exact number of patches varied from image to image. Here we report the total inference time, which measured by calculating the exact inference time for each image individually and then taking the average. Taking the default setting (HRNet-W18s with input size of 128×128) in ablation experiments as an example, it took about 211ms (52ms, 81ms, 78ms for the above three parts respectively) to process an image (1024×2048) of Cityscapes on a single RTX 2080Ti GPU, which is still much faster than PolyTransform (575ms per image (Liang et al., 2020), measured on a single GTX 1080Ti GPU, which is about 35% slower than our RTX 2080Ti GPU with FP32 training (Li, 2019)). Undoubtedly, the network speed can be further improved with more efficient backbones (*e.g.*, MobileNets), smaller input size (*e.g.*, 32×32 or 64×64), and fewer inference patches (*e.g.*, with lower NMS thresholds or adaptively selecting the most unreliable patches). Note that the BPR models can still achieve a remarkable performance under these lightweight settings (Tables 5,6,7). The patch extraction and reassembling steps can also be accelerated with more CPU cores.

(5)*Limitation Analysis* The performance of our proposed framework relies on the initial masks. Some failure cases are illustrated in Fig. 8. For example, our model failed to produce

Fig. 7 Qualitative results on Cityscapes `val`. Results on ignored regions are not showed. **a** Comparison of Mask R-CNN and the results of our BPR on instance segmentation. **b** Comparison of HRNet and our BPR results on semantic segmentation. **c** Comparison of Panoptic-DeepLab and our BPR results on panoptic segmentation. Our method produced substantially better masks with more precise boundaries than baselines. Best viewed in color (Color figure online)

an optimal mask if the initially predicted boundaries were far from the real object boundaries (the 1st row), but note that we still refined this case to some extent (IoU was improved). In addition, if the initial mask over-segmented the neighboring instance, our model may regard the two instances as a whole and further amplify this error (the 2nd and 3rd rows) since we only process the local boundary regions without a global view. We analyzed the IoU improvements for all predicted instances on Cityscapes val set, shown in Fig. 9. In most cases, our refinement model improved the mask IoU (red dots above the dash line). However, we found that it was hard to refine instance masks with extremely low IoU (*e.g.*, < 0.1) due to the poor quality of initial boundaries. In addition, we observed that the improvement for smaller instances (about 2% in AP_S) was not as high as we got for larger instances (about 5% in AP_L).

(6)*Results on COCO Dataset* To demonstrate the generality of our framework, we also report the results on the COCO dataset (Lin et al., 2014), which contains 80 categories and more images (118k/5k for train/val). It is important to note that the coarse annotations in COCO may not fully reflect the improvements in mask quality (Gupta et al., 2019). Following some previous works (Kirillov et al., 2020; Zhang et al., 2021), we further report the $AP^\star$ measured using the higher quality LVIS (Gupta et al., 2019) annotations. We randomly sampled about 8% of instances for fast training. As shown in Table 13, we improved the powerful Mask R-CNN ResNeXt-FPN-101 baseline by 0.8% AP and 1.7% $AP^\star$ on `val2017`. The AP improvement on COCO dataset was not as high as we got on Cityscapes. The most critical problem is that the coarse polygon-based annotations on COCO dataset yield significantly lower boundary quality (Gupta et al., 2019). Several examples (which are ubiquitous on COCO) are shown in Fig. 10 (top row). The misalignment between annotations and real instance boundaries may greatly increase the optimization difficulty of our refinement model. Especially, the coarse annotations may provide ambiguous optimization objectives for our local boundary patches, thus hampering the model convergence. We observed that some *contour-based* instance segmentation methods (Xie et al., 2020; Xu et al., 2019; Peng et al., 2020), which are sensitive to the quality of boundary annotations, also suffered from this misalignment issue. It seems that the coarse COCO annotations may not be friendly to these methods and it is hard to achieve very high AP scores by using these approaches. In spite of this, we still improved the Mask R-CNN results in some cases, shown in Fig. 10 (the middle and bottom rows). Some results were even better than the annotations.

Table 14 Results of ablation experiments for semantic segmentation

	w/ removing	w/ mask	mIoU	AF
HRNet-W18s	-	-	73.8	53.3
	✔		45.1	35.4
+ BPR		✔	75.4	59.6
	✔	✔	**75.8**	**60.2**

Best results are given in bold

Fig. 8 Illustration of some failure cases on Cityscapes `val` (Color figure online)

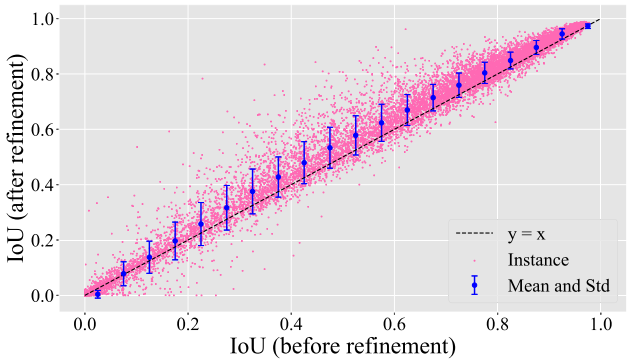

Fig. 9 IoU improvements for all predicted instances on Cityscapes val. A red dot indicates an instance. Dots below the dash line are failure cases (Color figure online)

Table 15 Semantic segmentation results on Cityscapes `val` set. R101 denotes ResNet101

	Road	Sidewalk	Building	Wall	Fence	Pole	Traffic light	Traffic sign	Vegetation	Terrian	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle	mIoU	AF
HRNet-W18s	97.7	82.8	91.7	56.3	56.7	62.8	65.1	75.7	92.0	61.4	93.7	79.5	56.2	93.9	64.1	78.8	63.8	55.9	74.1	73.8	53.3
+ SegFix	97.8	83.5	92.1	57.0	57.3	65.3	68.3	78.4	92.4	62.1	94.3	81.2	57.4	94.6	64.5	79.4	64.3	57.0	75.4	74.9	57.1
+ BPR	**98.0**	**84.2**	**92.7**	**57.6**	**58.0**	**67.9**	**73.6**	**81.2**	**92.8**	**63.1**	**94.7**	**82.9**	**60.2**	**95.0**	**67.4**	**80.1**	**65.5**	**58.9**	**77.6**	**76.4**	**61.8**
PointRend-R101	98.2	85.4	92.8	50.4	61.2	68.2	73.1	79.9	92.6	62.5	94.9	83.1	64.9	95.2	74.1	87.3	78.0	64.3	78.8	78.2	59.4
+ SegFix	**98.3**	86.0	93.2	51.0	61.8	70.4	75.7	82.3	93.0	63.0	95.3	84.6	66.2	95.7	74.7	88.0	78.7	65.5	79.9	79.1	62.6
+ BPR	**98.3**	**86.2**	**93.3**	**51.3**	**62.0**	**71.5**	**76.9**	**83.3**	**93.1**	**63.5**	**95.5**	**85.3**	**66.6**	**95.8**	**75.9**	**88.6**	**80.6**	**66.7**	**80.4**	**79.7**	**65.3**
DeepLabV3-R101	98.3	86.2	93.1	56.0	63.0	67.7	73.7	81.0	92.7	64.1	94.9	83.8	66.9	95.7	86.7	89.8	79.4	70.4	79.8	80.2	60.3
+ SegFix	**98.5**	86.9	93.5	**56.6**	63.6	70.3	76.3	83.5	93.1	64.5	95.4	85.3	68.3	**96.3**	87.2	90.7	80.1	71.5	80.9	81.2	63.9
+ BPR	**98.5**	**87.1**	**93.6**	**56.6**	**63.4**	**71.8**	**76.7**	**84.1**	**93.3**	**65.0**	**95.6**	**86.0**	**68.4**	**96.3**	**87.6**	**91.2**	**82.2**	**71.9**	**81.3**	**81.6**	**66.1**
HRNet-W48	98.5	87.0	93.5	58.5	64.7	71.4	75.6	82.8	93.2	64.8	95.3	84.7	66.9	95.8	82.9	91.5	82.9	69.8	80.1	81.0	62.7
+ SegFix	98.5	87.4	93.7	**59.0**	65.1	72.5	77.0	84.0	**93.4**	65.1	95.4	85.7	67.7	**96.1**	83.1	91.9	83.4	70.8	81.0	81.6	64.6
+ BPR	**98.6**	**87.6**	**93.8**	**59.0**	**65.3**	**72.9**	**77.4**	**84.5**	**93.4**	**65.9**	**95.6**	**86.0**	**68.2**	**96.1**	**83.8**	**92.2**	**84.3**	**71.5**	**81.2**	**82.0**	**66.3**
W48+OCR	98.5	87.4	93.6	59.1	65.1	71.2	75.0	82.4	93.1	66.2	95.3	83.8	64.0	95.8	86.3	91.2	84.4	70.1	79.6	81.2	63.3
+ SegFix	**98.6**	87.9	93.8	**59.6**	65.6	72.4	76.9	84.2	93.3	66.6	95.7	84.8	65.0	**96.3**	86.8	91.7	84.9	71.2	80.6	81.9	65.1
+ BPR	**98.6**	**88.1**	**93.9**	59.4	**65.7**	**72.8**	**77.3**	**84.6**	**93.4**	**66.8**	**95.9**	**85.2**	**65.3**	**96.3**	**87.1**	**91.8**	**85.8**	**72.0**	**80.8**	**82.2**	**67.1**
W48+OCR (MS)	98.6	88.2	93.8	61.5	66.5	72.3	76.8	83.6	93.3	68.2	95.4	85.2	67.5	96.1	88.8	91.9	86.8	72.2	80.9	82.5	66.3
+ SegFix	**98.7**	88.7	93.9	**61.8**	**66.8**	73.0	**78.1**	85.0	93.5	68.4	95.7	86.0	68.0	**96.5**	89.1	92.2	87.4	73.0	**81.6**	83.0	67.5
+ BPR	**98.7**	**88.8**	**94.0**	61.7	66.7	**73.2**	78.0	**85.2**	**93.6**	**68.7**	**95.8**	**86.3**	**68.1**	**96.5**	**89.2**	**92.3**	**87.9**	**73.3**	**81.6**	**83.1**	**68.7**

Best results are given in bold
MS denotes multi-scale inference

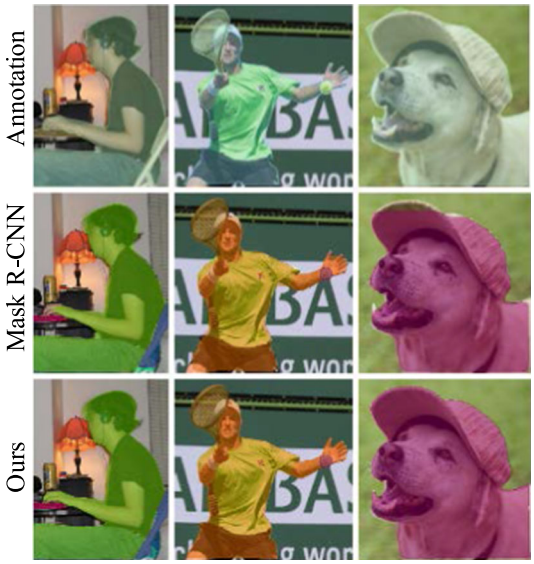

Fig. 10 Illustration of the coarse annotations (top row) on COCO `val2017`. The annotated instance masks are not well-aligned with the real object boundaries. The proposed model (bottom row) generated substantially better masks with more precise boundaries than Mask R-CNN (middle row) (Color figure online)

5 Experiments: Semantic Segmentation

5.1 Datasets, Metrics, and Implementation Details

(1)*Datasets and Metrics* We used the `fine` data of Cityscapes dataset, containing 2,975/500/1,525 images for train/val/test. Different from instance segmentation, 19 categories are involved for semantic segmentation. For evaluation, we report the frequently-used *class-wise mIoU* to measure the semantic segmentation performance, and the *boundary F-score (AF)* to measure the quality of predicted boundaries. AF calculation for semantic segmentation is slightly different from instance segmentation (Sect. 4.1) since there are no instance concept. We adopted exactly the same metric as previous studies (Takikawa et al., 2019; Yuan et al., 2020b; Wang et al., 2022) to calculate boundary F-score for each category, and then averaged over categories.

(2)*Implementation Details* The differences of applying BPR to instance segmentation and semantic segmentation are described in Sect. 3.3. Most configurations were the same as instance segmentation, except the batch size. Since more boundary patches were extracted for semantic segmentation results, the batch size was increased to 144. In this set of experiments, the BPR model trained on the results of HRNet-W18s was transferred to refine the results of other models.

5.2 Ablation Study

We have validated the effectiveness of most configurable design choices in previous experiments (Sect. 4.2). Here we conducted ablation experiments for the special design choices of semantic segmentation. The configurations described in Sect. 4.2 were adopted. We have the following observations. (1) As shown in Table 14 (the last two rows), removing inferior patches extracted near image border improved the performance echoing the analysis in Sect. 3.3 and Fig. 4. (2) The BPR model without mask patch failed to converge (only 45.1% AP). The reason is that image patches were usually shared by adjacent objects but the learning goals were different (see the last two rows of Fig. 5). Location and semantic information provided by the mask patches can avoid this issue.

5.3 Quantitative Results

We applied the BPR model to refine a variety of semantic segmentation results ((Wang et al., 2020b; Yuan et al., 2020a; Long et al., 2015; Chen et al., 2017; Kirillov et al., 2020), adopted from `MMSegmentation`). The BPR model was trained with the same configurations as described in Sect. 4.4. As shown in Table 15, we achieved consistent improvements over different baselines. For example, we improved the HRNet-W18s baseline by **2.6**% mIoU and **8.5**% AF. The significant improvement on the boundary-sensitive AF scores demonstrates the effectiveness on boundary refinement. On the powerful HRNet-W48-OCR baseline, we still improved 1.0% and 0.6% mIoU under the single-scale and multi-scale settings respectively. In addition, we consistently outperformed SegFix (Yuan et al., 2020b) on both mIoU and AF metrics over different baselines.

Note that the overall improvements on semantic segmentation models ($+0.6\% \sim +2.6\%$) are not as high as we got on instance segmentation ($+2.5\% \sim +6.1\%$). Delving into the class-wise mIoU in Table 15, we found that categories (*e.g.*, *traffic light, traffic sign, rider*) with smaller and fragmented labeling areas usually got more significant improvements than categories (*e.g.*, *road, building, vegetation*) with larger and coherent areas. The reason lies on the IoU definition. For smaller regions, boundary pixels make much more contributions in IoU calculation than for larger regions. Thus categories with smaller regions can benefit more from boundary refinement.

5.4 Qualitative Results

We show some qualitative results on Cityscapes `val` in Fig. 7b. Compared with the initial predictions of HRNet, our BPR framework generated better semantic segmentation results with precise boundaries. One limitation is that the

Table 16 Panoptic segmentation results on Cityscapes `val` set

	Standard mask PQ					Boundary PQ				
	PQ	PQ^{Th}	PQ^{St}	SQ	RQ	PQ	PQ^{Th}	PQ^{St}	SQ	RQ
UPSNet-ResNet50	59.1	54.0	62.8	79.6	72.8	33.2	36.7	30.6	63.0	51.6
+ BPR	**61.5**	**55.8**	**65.7**	**81.9**	**74.0**	**39.0**	**41.8**	**36.9**	**66.7**	**57.3**
UPSNet-ResNet101-COCO	61.0	57.5	63.6	80.8	74.3	35.8	42.4	31.1	63.4	55.5
+ BPR	**63.7**	**59.6**	**66.7**	**82.8**	**75.9**	**42.2**	**47.6**	**38.2**	**67.0**	**61.9**
Panoptic-DeepLab-ResNet52	60.3	51.1	67.0	81.5	72.9	39.2	39.1	39.2	65.6	58.8
+ BPR	**61.8**	**53.3**	**68.0**	**82.6**	**74.0**	**42.3**	**43.2**	**41.6**	**67.3**	**61.8**
Panoptic-DeepLab-HRNet48	63.4	56.3	68.6	81.9	76.4	42.7	44.8	41.1	66.5	63.0
+ BPR	**64.7**	**58.8**	**69.1**	**82.6**	**77.3**	**44.7**	**47.9**	**42.4**	**68.0**	**64.7**

Best results are given in bold

performance of refinement relies on the initial predictions, which is similar to instance segmentation. For example, our model failed to refine the poor initial masks illustrated in Fig. 7b (last column, dashed boxes).

6 Experiments: Panoptic Segmentation

6.1 Datasets, Metrics, and Implementation Details

(1)*Datasets and Metrics* We used the `fine` data of Cityscapes dataset. There are 8 'thing' and 11 'stuff' classes. For evaluation, we used the panoptic quality (PQ) metric (Kirillov et al., 2019b) to measure the performance, including the breakdowns of recognition (RQ) *vs.*segmentation (SQ) performance and stuff (PQ^{St}) *vs.*things (PQ^{Th}) performance. In addition to the standard mask PQ (Kirillov et al., 2019b), we further evaluated the performance with a recently proposed metric designed for measuring the boundary quality, boundary PQ (Chen et al., 2021), to demonstrate effectiveness of the proposed method for boundary refinement.

(2)*Implementation Details* The differences of applying BPR to instance segmentation and panoptic segmentation are described in Sect. 3.3. Most configurations were the same as instance segmentation, except the batch size (which was increased to 256). In this set of experiments, BPR trained on the results of UPSNet-ResNet-50 (Xiong et al., 2019) was transferred to refine the results of other models. We omitted ablation experiments here because the conclusions drawn in previous ablation experiments (Sec. 4.2 and Sect. 5.2) were also valid for panoptic segmentation.

6.2 Quantitative Results

We applied the BPR model to refine the results of several typical panoptic segmentation models. The BPR model was trained with the same configurations as in Sect. 4.4. As shown in Table 16, we achieved consistent improvements over different baselines (Xiong et al., 2019; Chen et al., 2020a). For example, we improved the UPSNet-ResNet-101-COCO (Xiong et al., 2019) baseline by **2.7**% PQ and by **6.4**% boundary PQ. The improvements in terms of boundary PQ are more significant than the standard mask PQ, which suggests that the proposed method successfully improved the boundary quality for panoptic segmentation. Notably, for the two UPSNet models, the improvements on thing classes (PQ^{Th}) were larger than on stuff classes (PQ^{St}), while the observation for the two Panoptic-DeepLab models was opposite. Besides, the overall improvements on UPSNet models ($+2.4\% \sim +2.7\%$ on standard PQ) were larger than on Panoptic-DeepLab models ($+1.3\% \sim +1.5\%$ on standard PQ). These differences may be due to distinct design principles for these two methods. UPSNet (Xiong et al., 2019) is a *top-down* method, which first produces well-segmented instance results and then fills in the remaining regions with a semantic head. Panoptic-DeepLab (Chen et al., 2020a) is a *bottom-up* method, which groups instances from the well-segmented semantic results. As a result, UPSNet pays more attention on thing classes (higher PQ^{Th}), but Panoptic-DeepLab pays more attention on stuff classes (higher PQ^{St}).

6.3 Qualitative Results

We show some qualitative results on Cityscapes `val` in Fig. 7c. Compared with the initial predictions of Panoptic-DeepLab, our BPR framework generated better panoptic segmentation results with precise and clear boundaries. Similar limitation was observed for panoptic segmentation. As shown in Fig. 7c (last column, red dashed box), the model failed to refine the poor initial masks.

7 Conclusion

In this paper, we present a conceptually simple yet effective boundary refinement framework to improve the boundary

quality for any image segmentation model. Starting from a coarse mask, we extract and refine a series of boundary patches along the predicted boundaries through an effective boundary refinement network. The proposed BPR framework achieved consistent and impressive improvements on different image segmentation tasks over a variety of baselines. Qualitative results showed that our BPR model produced high-quality masks with precise and clear boundaries.

Post-processing is a *double-edged sword*. Our proposed BPR framework can improve the results of any image segmentation model, but meanwhile increase the total inference time. We will explore the speed-up of BPR in future works.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Nos. 62061136001, 61836014, and U19B2034), the National Science and Technology Major Project (No. 2018ZX01028-102), and THU-Bosch JCML center.

References

Bai, M., & Urtasun, R. (2017). Deep watershed transform for instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 5221–5229).

Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). YOLACT: Real-time instance segmentation. In *IEEE International Conference on Computer Vision (ICCV)*. (pp. 9157–9166).

Chen, H., Sun, K., Tian, Z., Shen, C., Huang, Y., & Yan, Y. (2020a). BlendMask: Top-down meets bottom-up for instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 8573–8581).

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., et al. (2019). MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155.

Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision (ECCV)*. (pp. 801–818).

Chen, Z., Zhou, H., Lai, J., Yang, L., & Xie, X. (2020). Contour-aware loss: Boundary-aware learning for salient object segmentation. *IEEE Transactions on Image Processing (TIP), 30*, 431–443.

Cheng, B., Collins, M. D., Zhu, Y., Liu, T., Huang, T. S., Adam, H., et al. (2020a). Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 12475–12485).

Cheng, B., Girshick, R., Dollár, P., Berg, A. C., & Kirillov, A. (2021). Boundary iou: Improving object-centric image segmentation evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 15334–15342).

Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., & Girdhar, R. (2022). Masked-attention mask transformer for universal image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 1290–1299).

Cheng, T., Wang, X., Huang, L., & Liu, W. (2020b). Boundary-preserving Mask R-CNN. In *European Conference on Computer Vision (ECCV)*. (pp. 660–676).

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., et al. (2016). The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 3213–3223).

de Geus, D., Meletis, P., & Dubbelman, G. (2018). Panoptic segmentation with a joint semantic and instance segmentation network. arXiv preprint arXiv:1809.02110.

Ding, H., Jiang, X., Shuai, B., Liu, A. Q., & Wang, G. (2020). Semantic segmentation with context encoding and multi-path decoding. *IEEE Transactions on Image Processing (TIP), 29*, 3520–3533.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., et al. (2019). Dual attention network for scene segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 3146–3154).

Gao, N., Shan, Y., Wang, Y., Zhao, X., Yu, Y., Yang, M., et al. (2019). SSAP: Single-shot instance segmentation with affinity pyramid. In *IEEE International Conference on Computer Vision (ICCV)*. (pp. 642–651).

Gupta, A., Dollar, P., & Girshick, R. (2019). LVIS: A dataset for large vocabulary instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 5356–5364).

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*. (pp. 2961–2969).

Huang, Z., Huang, L., Gong, Y., Huang, C., & Wang, X. (2019). Mask Scoring R-CNN. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 6409–6418).

Kim, M., Woo, S., Kim, D., & Kweon, I. S. (2021). The devil is in the boundary: Exploiting boundary representation for basis-based instance segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WAVC)*. (pp. 929–938).

Kirillov, A., Girshick, R., He, K., & Dollár, P. (2019a). Panoptic feature pyramid networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 6399–6408).

Kirillov, A., He, K., Girshick, R., Rother, C., & Dollár, P. (2019b). Panoptic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 9404–9413).

Kirillov, A., Levinkov, E., Andres, B., Savchynskyy, B., & Rother, C. (2017). InstanceCut: from edges to instances with multicut. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 5008–5017).

Kirillov, A., Wu, Y., He, K., & Girshick, R. (2020). PointRend: Image segmentation as rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 9799–9808).

Krähenbühl, P., & Koltun, V. (2011). Efficient inference in fully connected crfs with gaussian edge potentials. In *Neural Information Processing Systems (NeurIPS)*. (pp. 109–117).

Lazarow, J., Lee, K., Shi, K., & Tu, Z. (2020). Learning instance occlusion for panoptic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 10720–10729).

Li, C. (2019). NVIDIA RTX 2080 Ti deep learning benchmarks with TensorFlow. https://lambdalabs.com/blog/2080-ti-deep-learning-benchmarks/.

Li, X., Li, X., Zhang, L., Cheng, G., Shi, J., Lin, Z., et al. (2020). Improving semantic segmentation via decoupled body and edge supervision. In *European Conference on Computer Vision (ECCV)*. (pp. 435–452).

Li, Y., Chen, X., Zhu, Z., Xie, L., Huang, G., Du, D., et al. (2019). Attention-guided unified network for panoptic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 7026–7035).

Liang, J., Homayounfar, N., Ma, W.-C., Xiong, Y., Hu, R., & Urtasun, R. (2020). PolyTransform: Deep polygon transformer for instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 9131–9140).

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *IEEE International Conference on Computer Vision (ICCV)*. (pp. 2980–2988).

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*. (pp. 740–755).

Liu, S., Jia, J., Fidler, S., & Urtasun, R. (2017). SGN: Sequential grouping networks for instance segmentation. In *IEEE International Conference on Computer Vision (ICCV)*. (pp. 3496–3504).

Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 8759–8768).

Liu, Y., Song, G., Zang, Y., Gao, Y., Xie, E., Yan, J., et al. (2020). 1st place solutions for openimage2019–object detection and instance segmentation. arXiv preprint arXiv:2003.07557.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE International Conference on Computer Vision (ICCV)*. (pp. 10012–10022).

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 3431–3440).

Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *Proceedings of the Fourth International Conference on 3D Vision (3DV)*. (pp. 565–571).

Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., & Terzopoulos, D. (2021a). Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)*.

Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., & Terzopoulos, D. (2021b). Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)*.

MMSegmentation, C. (2020). MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation.

Mohan, R., & Valada, A. (2020). EfficientPS: Efficient panoptic segmentation. *International Journal on Computer Vision (IJCV), 129*(5), 1551–1579.

Peng, S., Jiang, W., Pi, H., Li, X., Bao, H., & Zhou, X. (2020). Deep snake for real-time instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 8533–8542).

Poudel, R. P., Liwicki, S., & Cipolla, R. (2019). Fast-SCNN: fast semantic segmentation network. arXiv preprint arXiv:1902.04502.

Ren, J., Yu, C., Cai, Z., Zhang, M., Chen, C., Zhao, H., et al. (2021). REFINE: Prediction fusion network for panoptic segmentation. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*. (pp. 2477–2485).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NeurIPS)*. (pp. 91–99).

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention (MICCAI)*. (pp. 234–241).

Sofiiuk, K., Barinova, O., & Konushin, A. (2019). AdaptIS: Adaptive instance selection network. In *IEEE International Conference on Computer Vision (ICCV)*. (pp. 7355–7363).

Takikawa, T., Acuna, D., Jampani, V., & Fidler, S. (2019). Gated-SCNN: Gated shape cnns for semantic segmentation. In *IEEE International Conference on Computer Vision (ICCV)*. (pp. 5229–5238).

Tang, C., Chen, H., Li, X., Li, J., Zhang, Z., & Hu, X. (2021). Look closer to segment better: Boundary patch refinement for instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 13926–13935).

Tian, Z., Shen, C., & Chen, H. (2020). Conditional convolutions for instance segmentation. In *European Conference on Computer Vision (ECCV)*. (pp. 282–298).

Tian, Z., Shen, C., Chen, H., & He, T. (2019). FCOS: Fully convolutional one-stage object detection. In *IEEE International Conference on Computer Vision (ICCV)*. (pp. 9627–9636).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Neural Information Processing Systems (NeurIPS)*. (pp. 5998–6008).

Wang, C., Zhang, Y., Cui, M., Ren, P., Yang, Y., Xie, X., et al. (2022). Active boundary loss for semantic segmentation. In *Association for the Advancement of Artificial Intelligence (AAAI)*. (pp. 2397–2405).

Wang, H., Zhu, Y., Adam, H., Yuille, A., & Chen, L.-C. (2021). Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 5463–5474).

Wang, H., Zhu, Y., Green, B., Adam, H., Yuille, A., & Chen, L.-C. (2020a). Axial-DeepLab: Stand-alone axial-attention for panoptic segmentation. In *European Conference on Computer Vision (ECCV)*. (pp. 108–126).

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., et al. (2020b). Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)*.

Wang, X., Kong, T., Shen, C., Jiang, Y., & Li, L. (2020c). SOLO: Segmenting objects by locations. In *European Conference on Computer Vision (ECCV)*. (pp. 649–665).

Wang, X., Zhang, R., Kong, T., Li, L., & Shen, C. (2020d). SOLOv2: Dynamic and fast instance segmentation. In *Neural Information Processing Systems (NeurIPS)*. (pp. 17721–17732).

Wang, Y., Zhao, X., Hu, X., Li, Y., & Huang, K. (2019). Focal boundary guided salient object detection. *IEEE Transactions on Image Processing (TIP), 28*(6), 2813–2824.

Xie, E., Sun, P., Song, X., Wang, W., Liu, X., Liang, D., et al. (2020). PolarMask: Single shot instance segmentation with polar representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 12193–12202).

Xiong, Y., Liao, R., Zhao, H., Hu, R., Bai, M., Yumer, E., et al. (2019). UPSNet: A unified panoptic segmentation network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 8818–8826).

Xu, W., Wang, H., Qi, F., & Lu, C. (2019). Explicit shape encoding for real-time instance segmentation. In *IEEE International Conference on Computer Vision (ICCV)*. (pp. 5168–5177).

Yang, T.-J., Collins, M. D., Zhu, Y., Hwang, J.-J., Liu, T., Zhang, X., et al. (2019). DeeperLab: Single-shot image parser. arXiv preprint arXiv:1902.05093.

Ying, H., Huang, Z., Liu, S., Shao, T., & Zhou, K. (2019). EmbedMask: Embedding coupling for one-stage instance segmentation. arXiv preprint arXiv:1912.01954.

Yuan, Y., Chen, X., & Wang, J. (2020a). Object-contextual representations for semantic segmentation. In *European Conference on Computer Vision (ECCV)*. (pp. 173–190).

Yuan, Y., Xie, J., Chen, X., & Wang, J. (2020b). SegFix: Model-agnostic boundary refinement for segmentation. In *European Conference on Computer Vision (ECCV)*. (pp. 489–506).

Zhang, G., Lu, X., Tan, J., Li, J., Zhang, Z., Li, Q., et al. (2021). RefineMask: Towards high-quality instance segmentation with fine-grained features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 6861–6869).

Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., et al. (2018). Context encoding for semantic segmentation. In *IEEE Conference*

on Computer Vision and Pattern Recognition (CVPR). (pp. 7151–7160).

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 2881–2890).

Zhou, P., Price, B., Cohen, S., Wilensky, G., & Davis, L. S. (2020). Deepstrip: High-resolution boundary refinement. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (pp. 10558–10567).

Zhou, S., Nie, D., Adeli, E., Yin, J., Lian, J., & Shen, D. (2019). High-resolution encoder-decoder networks for low-contrast medical image segmentation. *IEEE Transactions on Image Processing (TIP), 29*, 461–475.