PRN: Panoptic Refinement Network

Bo Sun[1] Jason Kuen[1] Zhe Lin[1] Philippos Mordohai[2] Simon Chen[1]
[1]Adobe Inc. [2]Stevens Institute of Technology
{bosu, kuen, zlin, sichen}@adobe.com philippos.mordohai@stevens.edu

Abstract

Panoptic segmentation is the task of uniquely assigning every pixel in an image to either a semantic label or an individual object instance, generating a coherent and complete scene description. Many current panoptic segmentation methods, however, predict masks of semantic classes and object instances in separate branches, yielding inconsistent predictions. Moreover, because state-of-the-art panoptic segmentation models rely on box proposals, the instance masks predicted are often of low-resolution. To overcome these limitations, we propose the Panoptic Refinement Network (PRN), which takes masks from base panoptic segmentation models and refines them jointly to produce coherent results. PRN extends the offset map-based architecture of Panoptic-Deeplab with several novel ideas including a foreground mask and instance bounding box offsets, as well as coordinate convolutions for improved spatial prediction. Experimental results on COCO and Cityscapes show that PRN can significantly improve already accurate results from a variety of panoptic segmentation networks.

1. Introduction

Panoptic segmentation addresses semantic and instance segmentation in a unified way, aiming to assign each pixel to one of the background classes (i.e. stuff) or one of the object instances (i.e. things) [22]. Facilitated by the introduction of several open-source datasets (e.g. Cityscapes [11], COCO [33], Mapillary Vistas [37]), panoptic segmentation has quickly become a popular research topic leading to significant progress [8, 9, 17, 21, 24, 26, 27, 34, 45, 47, 49, 51] since its introduction.

Despite this progress, panoptic segmentation results still suffer from a variety of artifacts. Some of these artifacts are due to the difficulty of the problem and are caused by occlusion, visual similarity between instances, etc. Other artifacts, however, are caused by limitations of the panoptic segmentation models used. We distinguish limitations that cause inaccurate boundaries between instances and stuff, or between different instances, mainly due to components of panoptic segmentation networks operating at low resolution, and limitations that cause inconsistencies due to imperfect merging of semantic and instance predictions, which are typically made by different branches of the network.

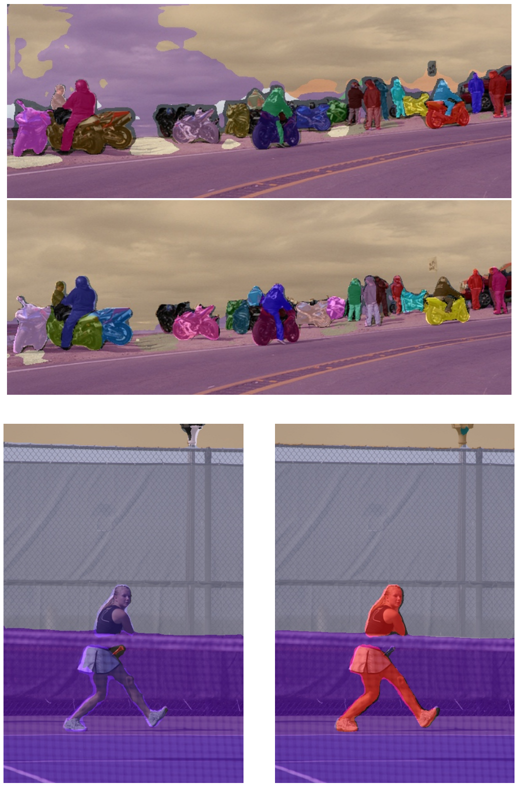

Figure 1. Top row: input panoptic segmentation by MS-PanopticFPN. Second row: PRN makes large corrections to the sky and ground, and smaller corrections to all instances. Bottom row: PRN is able to recover the tennis player (right) who was missing in the input (left). Note that the color of an instance masks represents the index of the instance, and not its class label.

For example, the above issues manifest themselves in the results of Panoptic FPN [21], a groundbreaking panoptic segmentation method. Panoptic FPN is a two-stage approach that relies on Mask R-CNN [15] to extract region of interest (RoI) features and generate low-resolution (e.g., 14×14 or 28×28) instance mask proposals. Many subsequent panoptic segmentation methods [8, 27, 26, 34, 38, 40, 51] also rely on low-resolution RoI-based mask prediction. Such low-resolution masks cannot capture the fine details of object boundaries precisely and fail to achieve high-quality segmentation results. Furthermore, it is common for existing methods to train independent instance segmentation and semantic segmentation branches to predict instance and stuff masks separately. This typically calls for heuristic-driven postprocessing [21] to resolve the conflicts among instance and stuff masks in the panoptic segmentation map, potentially producing unsatisfactory outcomes.

To tackle the above weaknesses of current panoptic segmentation methods, we propose to refine their low-resolution mask predictions by developing a new dedicated mask refinement network for panoptic segmentation. Our design is inspired by the observation that single-shot panoptic segmentation networks, such as Panoptic-DeepLab [9], are able to predict high-resolution masks for instances and stuff, but suffer from low recognition accuracy compared to two-stage methods that leverage object proposals. *We propose to achieve the best of both worlds by re-purposing an architecture similar to Panoptic-DeepLab to refine an initial panoptic segmentation of an image, instead of using it to segment the image from scratch.* This allows us to benefit from the correctly recognized, but imprecisely segmented, outputs from a two-stage panoptic segmentation network in a framework that can obtain precise segmentation boundaries, as well as learn to correct systematic errors of the base panoptic segmentation network, as shown in Fig. 1. As discussed in Section 2, semantic segmentation refinement methods fall short in panoptic segmentation refinement. SegFix [55], which is arguably the state of the art, still cannot generate missing masks, and is outperformed by PRN in our experiments.

To achieve these goals, we extended the Panoptic-DeepLab architecture with mechanisms to predict a foreground mask and bounding box offsets at each pixel. The foreground mask is class-agnostic, which allows the network to predict it at high-resolution thus enabling more precise interaction between thing and stuff masks. The bounding box offsets, meanwhile, play an important role in grouping pixels into instances. They are aided by the use of CoordConv [35] in the encoder and decoder.

In summary, we present the Panoptic Refinement Network (PRN) which is a general, effective, and efficient refinement method that can be trained to improve the results of any base panoptic segmentation network. It is the first approach to tackle the segmentation quality limitations of existing two-stage panoptic segmentation methods, while preserving, or improving, their strong classification performance. The contributions of this paper are as follows:

- A panoptic refinement network that improves boundary consistency across instances and stuff, reduces artifacts due to low-resolution instance masks, and is able to insert and delete instance masks.
- Novelties in the architecture of PRN, such as foreground mask estimation, coordinate convolution and per-pixel instance bounding box prediction that enable the above corrections and are generally applicable.
- Extensive experiments and ablation studies assessing PRN's effectiveness on improving the output of three diverse based panoptic segmentation algorithms.

2. Related Work

In this section, we focus on methods employing deep networks, acknowledging that earlier, conventional approaches have also been published [43, 44, 53]. There are several approaches [8, 21, 27, 26, 39, 40, 51] adopting two-stage, or top-down, architectures inspired by Mask R-CNN [15]. Kirillov et al. [21] endow Mask R-CNN with a semantic segmentation branch using a shared Feature Pyramid Network backbone. Li et al. [27] present a unified framework for instance and stuff segmentation with object-level and pixel-level attention. Porzi et al. [39] employ a crop-aware bounding box regression loss to handle objects at a wide range of scales in high-resolutions images, extending their previous work [38]. The Adaptive Instance Selection (AdaptIS) network [40] performs class-agnostic instance segmentation based on point proposals, while the exemplar-based open-set panoptic segmentation network (EOPSN) [18] can segment known and unknown objects. BANet [8] is based on a bidirectional learning pipeline that enables feature-level interaction between instance and semantic segmentation. Similarly, the Bidirectional Graph Reasoning Network [49] is a graph convolutional network for bidirectional feature fusion at the proposal and class level.

A limitation of the above methods is that they do not optimize a panoptic loss function, but intermediate outputs that are fused heuristically. Panoptic losses were introduced to address this. Liu et al. [34] propose an end-to-end occlusion aware network for panoptic segmentation, which also predicts the ordering of instances. UPSNet [51] relies on deformable convolution for semantic segmentation and Mask R-CNN-style instance segmentation to solve both sub-problems simultaneously. SOGNet [52] models overlap relations among instances by introducing the scene overlap graph. Li et al. [26] present an end-to-end network that does not rely on heuristic post-processing and thus unifies the training and inference pipelines. To address occlusion in instance segmentation, Lazarow et al. [24] model the binary relationship between overlapping instance masks.

We now turn our attention to single-shot, or bottom-up, methods that do not require object proposals. PRN can be applied on the outputs of these methods as well, as shown in Section 4. Cheng et al. [9] introduce Panoptic-DeepLab that employs a class-agnostic instance segmentation branch with instance center regression coupled with DeepLab [5] semantic segmentation outputs. SSAP [12] is a single-shot instance segmentation approach based on a pixel-pair affinity pyramid, which computes the probability that two pixels belong to the same instance in a hierarchical manner. Category- and instance-aware pixel embedding (CIAE) [13] learns an embedding of pixelwise features that encodes both semantic classification and instance distinction information. Pixel Consensus Voting [45] uses a generalized Hough transform for instance segmentation and a unified architecture that jointly models things and stuff. Similarly, Li et al. [28] represent and predict things and stuff in a fully convolutional manner, while Kerola et al. [19] propose Hierarchical Lovász Embeddings for the same purpose.

Axial-DeepLab [47] is a fully attentional network with novel position-sensitive axial-attention layers that combine self-attention for non-local interactions with positional sensitivity. The subsequent MAX-DeepLab [46] integrates a transformer and a CNN in a dual-path architecture, and directly predicts a set of object and stuff masks with a mask transformer. DEtection TRansformer (DETR) [4] introduces a set-based global loss that forces unique predictions via bipartite matching, and a transformer encoder-decoder architecture. DETR is one of the base networks in our experiments. It has been extended [29, 54] as transformer technology rapidly evolves.

Other aspects of panoptic segmentation have also been investigated. Real-time panoptic segmentation [17, 48] is valuable for robotics and autonomous driving. Hou et al. [17] present a new single-shot panoptic segmentation network that leverages dense detections and a global self-attention mechanism to achieve high frame rates with a small loss in accuracy. The Auto-Panoptic method [50] applies Network Architecture Search (NAS) on the components of panoptic segmentation.

Also related to our work are semantic [14, 5, 30] and instance segmentation [56, 10] refinement methods. The former, however, cannot handle boundaries between instances of the same type, while the latter refine one instance at a time. SegFix [55] is a recent, model-agnostic post-processing scheme that improves segmentation boundaries generated by existing methods. The key idea is that, since the label predictions for interior pixels are more reliable, they can be used to correct errors near boundaries. Unlike PRN, SegFix does not need to be trained for each baseline method, but PRN can recover masks that have been entirely missed, and delete large erroneous segments. SegFix is included in our experiments. The Panoptic, Instance, and Semantic Relations (PISR) model [2] captures the relations among semantic classes and instances, and is able to enhance the performance of existing panoptic segmentation systems. PISR was published too recently to allow detailed comparisons, but it seems to achieve similar improvements to PRN on common inputs.

3. The Panoptic Refinement Network (PRN)

We propose the Panoptic Refinement Network (PRN), an encoder-decoder which jointly refines the instance and semantic segmentation masks generated by a base panoptic segmentation network. We base the design of PRN on Panoptic-DeepLab [9] because it can predict high-resolution masks for instances and stuff jointly. This is due to its single-shot approach and center-based instance prediction mechanism. Panoptic-DeepLab, however, suffers from poor classification accuracy, according to the Recognition Quality (RQ) metric introduced by Kirillov et al. [22]. This is due to the severe class imbalance in the pixel-wise semantic segmentation training samples. Training tends to be dominated by stuff categories which have larger pixel counts in the images compared to the instance categories. In contrast, two-stage panoptic segmentation methods use a separate head to detect and classify instances, and as a result are less affected by class imbalance.

This observation motivated us to change the role of Panoptic-DeepLab from a conventional approach that tackles panoptic segmentation from scratch, to a panoptic refinement module that takes the well-categorized but coarsely-segmented outputs from a trained two-stage panoptic segmentation network and focuses on refining its low-resolution masks to achieve high-quality segmentation.

However, directly applying the architecture of Panoptic-DeepLab as a refinement module suffers from several limitations. First, to prevent excessive memory consumption, the semantic segmentation branch's multi-class output is lower-resolution and thus produces masks with limited segmentation fidelity. Second, detecting instances with instance center prediction and center offset regression is not sufficiently robust and may incorrectly split an instance into multiple instances. We incorporate several new ideas in PRN to addresses these limitations. In addition to the original prediction branches in Panoptic-DeepLab, we propose a class-agnostic foreground mask prediction branch that operates at the same high resolution as the input. To make instance prediction more robust, PRN's instance branch predicts bounding box offset values at each foreground/instance pixel, which are then used to group instance pixels in postprocessing. Additionally, we improve PRN's capability to regress instance offsets by making the network more coordinate-aware with CoordConv [35].

3.1. Overall Architecture

As shown in Fig. 2, PRN takes the RGB image, instance and semantic segmentation masks from the base network as input. PRN consists of four components: (1) an input module which extracts and concatenates the features from the RGB image, instance and semantic segmentation masks,

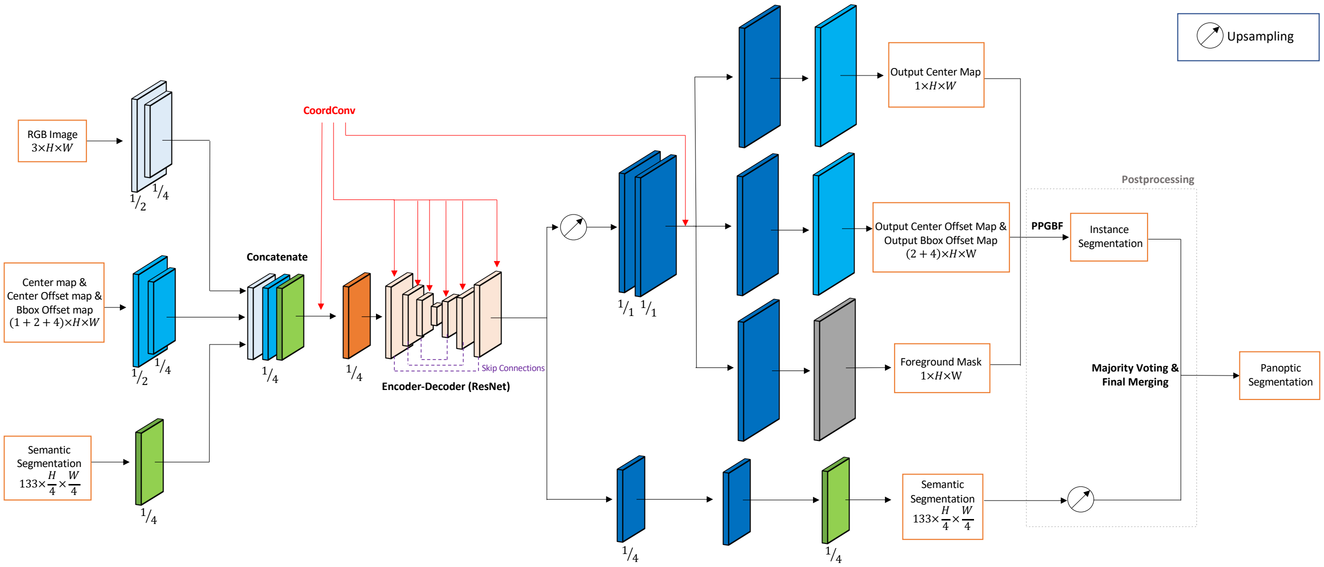

Figure 2. Overview of the architecture of Panoptic Refinement Network (PRN). PPGBF stands for Post-Processing Guided by Bounding Box and Foreground Mask.

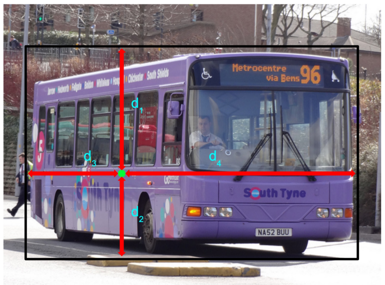

Figure 3. An illustration of the 4D bounding box offsets *in red* for the green pixel of an instance of *Bus*

(2) an encoder-decoder network for joint refinement of instance and semantic segmentation, (3) task-specific prediction branches for instance, semantic and foreground segmentation, (4) a postprocessing module guided by the predicted foreground mask and bounding box at each pixel.

3.2. Input Processing

The input to PRN has three parts: the RGB image, instance maps, and semantic segmentation maps, all from the base panoptic segmentation network. The semantic segmentation maps are often downsampled, typically by a factor of 4 in each dimension. In general, the input branch is adapted according to the output format of the base network.

Instance Maps. The input instance maps have seven channels. Following Panoptic-Deeplab [9], the first three channels represent a 1D center map and a 2D center offset map derived from the output of a base panoptic segmentation network. The center map is a heat map that indicates the probability of each pixel being an instance center. In the center offset map, each pixel contains the 2D offset values that map its location to the center of the instance it belongs to. See Fig. 4 for an example.

Motivated by the intuition that pixels of the same instance should be associated with the same bounding box, we design a novel *4D bounding box offset map* which complements the center and center offset maps to further constraint how PRN detects instances. As shown in Fig. 3, the four channels (d_1, d_2, d_3, d_4) correspond to the distances from the pixel to the top, bottom, left and right of the instance's bounding box. The bounding box offset maps make up the last four channels of the input instance maps.

Input Branches. The input image is fed to an RGB-specific input branch consisting of two 5×5, stride-2 convolutional layers to obtain RGB-specific features $V_{\text{rgb}} \in \mathbb{R}^{N_{en} \times \frac{H}{4} \times \frac{W}{4}}$, where N_{en} is the number of input channels to the encoder. The input instance segmentation mask is fed to an instance-specific input branch consisting of 5×5, stride-2 convolutional layers to produce instance-specific features $V_{\text{ins}} \in \mathbb{R}^{N_{en} \times \frac{H}{4} \times \frac{W}{4}}$.

The input semantic segmentation maps has N_{cl} channels, where N_{cl} is the number of semantic (things and stuff) classes. The label probabilities of the classes across all pixel locations are represented by the input semantic segmentation maps. It is fed to a semantic-specific input branch consisting of a 5×5 convolutional layer to generate the semantic-specific features $V_{\text{seg}} \in \mathbb{R}^{N_{en} \times \frac{H}{4} \times \frac{W}{4}}$.

We concatenate the features from all input branches along with a 2D normalized coordinate map C to obtain the feature maps $X \in \mathbb{R}^{(3N_{en}+2) \times \frac{H}{4} \times \frac{W}{4}}$, with $X = \text{Concat}(V_{\text{rgb}}, V_{\text{ins}}, V_{\text{seg}}, C)$. In order to predict the center and bounding box offset values effectively in the instance prediction output branch, PRN must be strongly aware of pixel coordinates. To this end, we leverage CoordConv [35] in panoptic segmentation by adding a 2D normalized coordinate map to X. In addition, CoordConv is applied to subse-

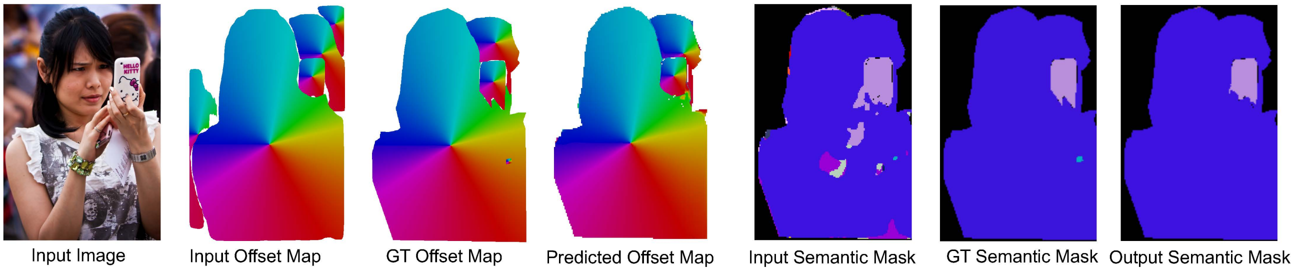

Figure 4. An example of the inputs, intermediate results and outputs of PRN

quent parts of PRN, including its encoder-decoder network and instance prediction branch, to further boost PRN's coordinate awareness.

3.3. Encoder-Decoder

After processing the input, the encoder-decoder generates multi-scale deep features for the output branches. Compared to Panoptic-Deeplab that learns separate decoders for the instance prediction and semantic segmentation output branches, we design an encoder-decoder network with an efficient shared decoder for both branches. We modify ResNet [16] by adding decoder layers to build the encoder-decoder network. First, we remove the first convolutional layer and plug in our input module, which has been described in Section 3.2. Second, we apply CoordConv to each bottleneck block of the encoder and to each layer of the decoder. Third, we feed the encoder's features at $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$ and $\frac{1}{32}$ scales to the decoder layers through skip connections. Other than the above, our encoder-decoder follows the architectural details of standard encoder-decoder networks [1, 6]. The encoder-decoder architecture is similar to the Feature Pyramid Network (FPN) [31], albeit with larger network capacity. We denote its output with $Y \in \mathbb{R}^{C_{de} \times \frac{1}{4} \times \frac{1}{4}}$, where C_{de} is number of the decoder's output channels.

3.4. Prediction Branches

The encoder-decoder's features, Y, are used in PRN's semantic segmentation branch and multiple instance prediction branches: instance center points, center offsets, bounding box offsets, and class-agnostic foreground mask. Due to the smaller output size of the instance prediction branch ($\frac{1}{4}$) of the encoder-decoder compared to the image size, a bilinear upsampling layer is first applied to the incoming features Y, before applying multiple parallel branches of two consecutive 5×5 and 1×1 convolutional layers to predict all instance outputs at the image resolution.

Semantic Segmentation. The semantic segmentation branch uses the same resolution ($\frac{1}{4}$) as the decoder's output and no upsampling is required. We train the semantic segmentation branch using cross entropy as the loss.

Center Points. In PRN, the ground-truth instances are represented by their centers of mass via 2D Gaussians as in Panoptic-Deeplab. The Mean Squared Error (MSE) loss is used to penalize the errors between predictions and the ground-truth in the 2D Gaussian-encoded center heat map. During inference, non-maximum suppression (NMS) is applied to obtain the instance centers.

Center & Bounding Box Offsets. In Panoptic-Deeplab, center offsets are predicted to associate each pixel with its corresponding instance's center point, such that pixels belonging to the same instance can be grouped together. However, such an approach is far from robust and often incorrectly splits an instance into multiple smaller instances, due to the use of a simple centerness-based criterion. To robustify the pixel grouping process in PRN, we propose to apply the representation we have adopted for the input, and additionally predict bounding box offset maps. The predicted offset values are the distances from the current pixel to the four sides of the box bounding the instance it belongs to, similar to FCOS [42] (see Fig. 3). We incorporate bounding box offset prediction in PRN by predicting four additional output channels on top of the two-channel center offset in the offset prediction branch. The offset branch is trained with the L1 loss.

Foreground Mask. In Panoptic-Deeplab, the semantic segmentation map acts as a background filter during inference. However, the segmentation map has relatively low resolution due to the computational and memory cost of predicting a dense pixel-wise segmentation map with numerous semantic categories. To this end, we propose a foreground mask prediction branch for PRN that outputs a class-agnostic (*objectness*) foreground mask to replace the semantic segmentation map as a more effective background filter. Given K binary ground-truth instance masks $B = \{B_i | i = \{1, 2, ..., K\}\}$, we compute the target 1D foreground mask using bitwise OR as, $B_1 \vee B_2 \vee ... \vee B_K$. Since the foreground mask is single-channel, memory consumption becomes less of an issue allowing the network to predict the foreground mask at the same high resolution as the input image. This provides higher segmentation fidelity especially to the instance categories, as the boundaries of instances which are in contact with the stuff masks are primarily decided by the foreground mask. Cross entropy loss is used to train the foreground mask branch.

3.5. Postprocessing

During inference, there are two postprocessing steps leading to the final panoptic segmentation map: (1) merging the center and offset maps to form the instance mask, (2) merging the semantic and instance segmentation masks to form the final panoptic segmentation map.

Post-Processing Guided by Bounding Box and Foreground Mask. (PPGBF). We design a novel postprocessing algorithm guided by both the predicted foreground mask and bounding box offset map. First, we perform keypoint-based non-maximum suppression on the instance center heat map to obtain the center point prediction keeping the top-k highest scores that are also above a threshold θ. We set $\theta = 0.2$ and $k = 200$. Second, we assign the pixels on the center offset map to the nearest centers which should have an IoU greater than 0.5 with the instance's bounding box determined by its center point from the center heat map. We remove pixels that cannot be assigned to any center points' bounding boxes with IoU greater than 0.5. Last, we use the predicted foreground mask to filter out bounding boxes with background pixels.

Majority Voting & Final Merging. Given the predicted semantic segmentation and class-agnostic instance segmentation results, we adopt a majority voting technique to obtain the category label of each instance mask. In particular, the semantic label of a predicted instance mask is inferred by the majority of its pixels' predicted labels in the semantic segmentation map. Then, we merge the semantic segmentation and instance segmentation results to obtain the final panoptic segmentation map.

4. Experimental Results

In this section, we describe the base panoptic segmentation models and experimental settings, followed by our results and ablation studies. (See the supplement for details on the licenses of the external software and datasets used.)

4.1. Base Panoptic Segmentation Networks

PRN is trained to refine the results of a base panoptic segmentation network trained on the same dataset. Here, we use DETR [4], Real-time Panoptic [17], and a variant of PanopticFPN [21], dubbed MS-PanopticFPN, as base networks. DETR [4] is a state-of-the-art detection method which performs very well in panoptic segmentation. Real-time Panoptic [17] is a single-shot panoptic segmentation network that leverages dense detections and a global self-attention mechanism to achieve real-time performance and near-SOTA accuracy.

Multi-source Panoptic Feature Pyramid Network (MS-PanopticFPN). We use a panoptic feature pyramid network as an additional base panoptic segmentation network, pretrained on multiple source datasets for better generalization. MS-PanopticFPN comprises detection, instance segmentation and semantic segmentation modules. The detection module is based on ATSS [57], modified to include a hierarchical classification head, with decoupled objectness and classification prediction heads. The detection loss consists of three parts: centerness loss, bounding box regression loss and focal loss [32] for classification. The instance and semantic segmentation modules share parameters with the detection module. The semantic segmentation branch follows Hou et al.'s [17] design, but we rely on dice [41] and focal [32] losses for semantic segmentation. We also employ the instance segmentation branch from CenterMask [25] and use focal loss to train it. (More details on MS-PanopticFPN are presented in the supplement.)

4.2. Datasets, Experimental Setup and Evaluation Metrics.

We evaluate PRN on two datasets:

(1) The **COCO** dataset [33] is a widely used benchmark which was developed for instance segmentation, but stuff annotations were recently added [3]. It contains 118K, 5K, and 20K images for training, validation, and testing, respectively, with 80 thing and 53 stuff classes.

(2) **Cityscapes** [11] is a street-scene dataset containing high-resolution images $(1,024 \times 2,048)$ with pixel-accurate annotations for 8 thing and 11 stuff classes. There are 2975, 500, and 1525 images for training, validation, and testing, respectively.

MS-PanopticFPN Training. We pretrained MS-PanopticFPN on two different datasets: instance segmentation is pretrained on 105 object categories from OpenImages [23] and semantic segmentation on 80 stuff categories from COCO stuff [3]. For the COCO Panoptic dataset [33], we resize the training images so that their shorter side is 640 pixels, their longer side is no more than 1,066 pixels, and also apply random horizontal flipping and GridMask data augmentation [7]. The network is trained for 150K iterations with a batch size of 16 using Stochastic Gradient Descent (SGD) with 0.9 momentum and 0.00001 weight decay. We set the initial learning rate to 0.01 and use the cosine annealing learning rate scheduler [36].

PRN Training. For both datasets, we resize the training images to keep their shorter side at 640 and their longer side at or below 800 pixels, and apply random horizontal flipping and GridMask data augmentation. PRN is trained using the Adam optimizer [20] with 0.9 momentum and 0.0001 weight decay. We set the initial learning rate to 0.001 and use the cosine annealing learning rate scheduler. On COCO, PRN is trained for 150K iterations with a batch size of 16.

On Cityscapes, it is trained for 60K iterations with a batch size of 32. The loss has the five components presented in Section 3.4:

$$\begin{aligned}\mathcal{L}_{PRN} = \lambda_0 \mathcal{L}_{\text{sem}} + \lambda_1 \mathcal{L}_{\text{center\_heatmap}} + \lambda_2 \mathcal{L}_{\text{center\_offset}} \\ + \lambda_3 \mathcal{L}_{\text{box\_offset}} + \lambda_4 \mathcal{L}_{\text{foreground}}.\end{aligned} \tag{1}$$

For all experiments, we set $\lambda_0 = 1$, $\lambda_1 = 200$, $\lambda_2 = 0.02$, $\lambda_3 = 0.02$, $\lambda_4 = 5$.

Method	Backbone	PQ	PQ^{Th}	PQ^{St}
Panoptic FPN [21]	Res50-FPN	39.0	45.9	28.7
Panoptic FPN [21]	Res101-FPN	40.3	47.5	29.5
UPSNet [51]	Res50-FPN	42.5	48.6	33.4
AUNet [27]	Res50-FPN	39.6	49.1	25.2
CIAE [13]	Res50-FPN	40.2	45.3	32.3
OCFusion [24]	Res50	41.3	49.4	29.0
BANet [8]	Res50-FPN	41.1	49.1	29.1
PCV [45]	Res50	37.5	40.0	33.7
RealTimePan [17]	Res50-FPN	37.1	41.0	31.3
BGRNet [49]	Res50-FPN	43.2	49.8	33.4
Unifying [26]	Res50-FPN	43.4	48.6	35.5
Panoptic-Deeplab [9]	Res50	35.1	-	-
AdaptIS [40]	Res50	35.9	40.3	29.3
AdaptIS [40]	Res101	37.0	41.8	29.9
AdaptIS [40]	ResNext101	42.3	49.2	31.8
Axial-DeepLab-L [47]	Axial-Res50-L	43.9	48.6	**36.8**
Auto-Panoptic [50]	Auto	44.8	**51.4**	35.0
HLE [19]	Res50	37.1	41.1	30.9
HLE [19]	Res101	38.1	42.8	31.0
MS-PanopticFPN	Res50-FPN	40.6	46.6	31.6
MS-PanopticFPN & **PRN**	Res50*	44.4	50.9	34.4
DETR [4]	Res50	43.4	48.2	36.3
DETR [4] & **PRN**	Res50*	**45.1**	51.2	36.5

Table 1. Panoptic segmentation results on the COCO validation set. * indicates the backbone used for PRN, not the base network.

Evaluation Metrics. We report results on the validation sets of both datasets using panoptic quality (PQ) [22] as the metric. PQ captures both recognition and segmentation quality (RQ and SQ), and treats both stuff and thing categories in a unified manner. Additionally, we use PQ^{St} and PQ^{Th} to report the performance on stuff and thing categories separately.

4.3. Results on COCO

Table 1 shows quantitative results on the COCO validation set. MS-PanopticFPN achieves comparable results to the top-performing methods. RPN, trained on its results, improves the PQ of MS-PanopticFPN by 3.8%. It also improves its RQ from 51.8% to 54.9%, and its SQ from 78.0% to 79.6%. We then train RPN on the panoptic segmentation results of DETR [4] and improve its PQ by 1.7%. It also improves its RQ from 53.8% to 55.7%, and its SQ from 79.3% to 79.8%.

Method	Backbone	PQ	PQ^{Th}	PQ^{St}
Panoptic FPN [21]	Res50-FPN	57.7	51.6	62.2
Panoptic FPN [21]	Res101-FPN	58.1	52.0	62.5
UPSNet [51]	Res50-FPN	59.3	54.6	62.7
AUNet [27]	Res50-FPN	56.4	52.7	59.0
OCFusion [24]	Res50	59.3	53.5	63.6
PCV [45]	Res50	54.2	47.8	58.9
Unifying [26]	Res50-FPN	61.4	54.7	66.3
Panoptic-Deeplab [9]	Res50	59.7	-	-
AdaptIS [40]	Res50	59.0	55.8	61.3
AdaptIS [40]	Res101	60.6	57.5	62.9
AdaptIS [40]	ResNext101	**62.0**	**58.7**	64.4
Seamless [38]	Res50-FPN	60.2	55.6	63.6
SSAP [12]	Res50-FPN	61.4	54.7	66.3
HLE [19]	Res50	59.8	51.1	66.1
HLE [19]	Res101	60.6	51.4	**67.2**
RealTimePan [17]	Res50-FPN	58.8	52.1	63.7
RealTimePan [17] & **SegFix**	HRNet-W48*	60.5	54.0	64.6
RealTimePan [17] & **PRN**	Res50*	61.9	55.8	64.3

Table 2. Panoptic segmentation results on the Cityscapes validation set. * indicates the backbone used for PRN or SegFix. PRN refines the results of the Real-time Panoptic network [17] and surpasses the performance of SegFix.

Notably, the PQ of PRN with MS-PanopticFPN as the base model is 9.3% better than that of Panoptic-Deeplab [9], even though we use part of Panoptic-Deeplab's instance mask representation (offset/center map). Figure 4 shows an example of PRN input, intermediate results and output. Figure 5 shows qualitative results on COCO dataset. PRN not only refines the boundary of the instance mask, but also *suppresses incorrectly detected instances and discovers missing instance masks*.

4.4. Results on Cityscapes

Quantitative results on the Cityscapes dataset are shown in Table 2. We train PRN on the panoptic segmentation results of Real-time Panoptic [17] and improve its PQ by 3.1%. As on the COCO dataset, PRN's PQ is 2.2% better than that of Panoptic-Deeplab [9], despite the similarities. Refining the results of Real-time Panoptic with PRN ranks a close second in the table, behind AdaptIS [40] with a much larger backbone. (Both PRN-refined results are better than all variants of AdaptIS on COCO.)

We also apply SegFix [55] on the same outputs of Real-time Panoptic and obtain lower overall PQ, lower PQ on things, and similar PQ on stuff compared to PRN. This is not surprising since SegFix cannot add or delete masks, but is effective on the refinement of existing boundaries. Figure 6 shows qualitative results of Real-time Panoptic [17], SegFix [55] and PRN on Cityscapes. The limitations of SegFix compared to PRN, being unable to create or delete masks, are visible in these examples.

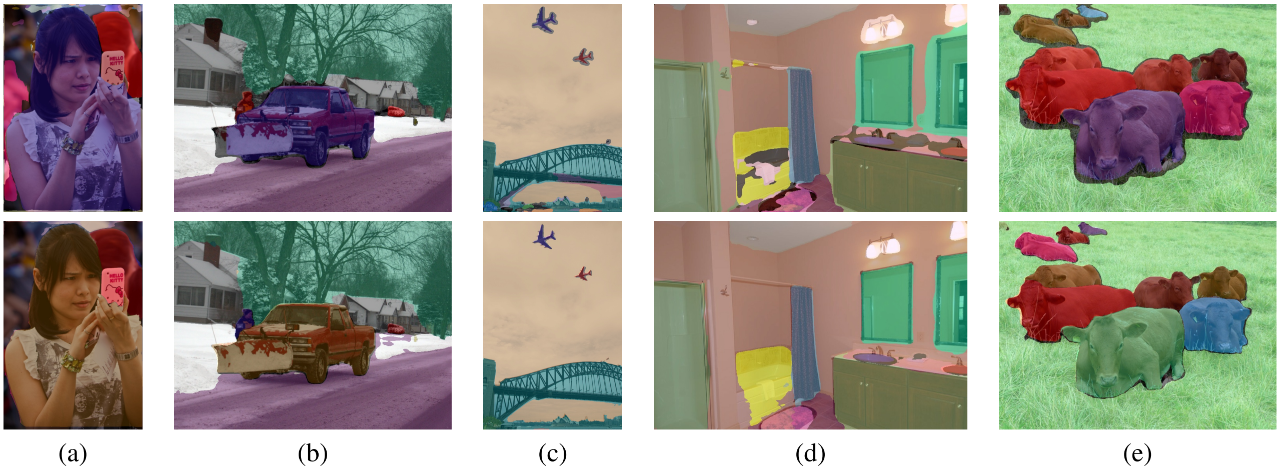

Figure 5. Qualitative results of MS-PanopticFPN (top) and PRN (bottom) on the COCO validation set. (The color of an instance mask represents the index of the instance, not its class label.) Notice: the suppressed instance mask in (a); the detection of the truck in (b), the sky under the bridge and plane boundaries in (c); mask insertion, deletion and refinement in (d); and boundary refinement and instance splitting in (e).

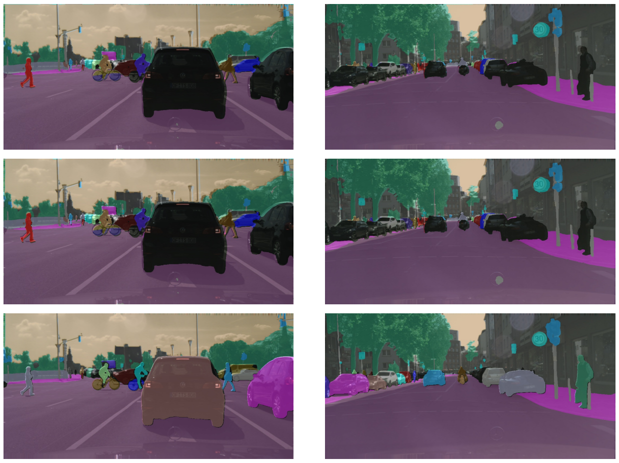

Figure 6. Qualitative results on Cityscapes validation set. Top: Real-Time Panoptic. Middle: SegFix. Bottom: PRN. (The color of an instance mask represents the index of the instance, and not its class label. Black pixels are unlabeled.) PRN recovers the missing car on the left, while SegFix cannot make such a correction. On the right, PRN not only finds the missing cars and person, but also obtains better segmentation masks for the sign and traffic lights.

4.5. Ablation Studies

We conduct ablation studies on the COCO validation set to evaluate the effectiveness of each component of PRN. We summarize them here and provide more details in the supplement. We first compare two ways of obtaining the foreground mask: (1) from the semantic segmentation branch, or (2) the foreground mask branch. The latter is more effective justifying our design choice. We then assess the contribution of CoordConv by applying it: (1) only in encoder layers, (2) only in decoder layers, (3) in both encoder and decoder layers. PQ is improved by an additional 1.4% when CoordConv is used in the encoder, 1.6% when it is used in the decoder, and 1.9% when it is used in both.

We can further improve PQ by 2.5% when we use predicted bounding boxes at each pixel when merging the center and offset maps. PQ is improved by 3.8% when we apply CoordConv in both encoder and decoder layers and use predicted bounding boxes in postprocessing.

4.6. Limitations

The main limitation of PRN at this point is that it must be trained on the results of a specific base panoptic segmentation network. Achieving more general applicability would make it much more useful and convenient.

5. Conclusion

We have presented a novel architecture for refining panoptic segmentation that is able to alleviate the common shortcomings of state-of-the-art panoptic segmentation algorithms. PRN reduces errors caused by inconsistency between instance and stuff segmentation, occlusion among instances of the same type, and low-resolution instances, while being able to recover missing instances, and fix incorrectly merged and split instances. This is accomplished via the introduction of novel elements including a foreground mask, coordinate convolution, and prediction of the bounding box offsets at each pixel. We experimentally validate PRN on challenging panoptic segmentation datasets demonstrating that the results of highly accurate panoptic segmentation networks can be significantly improved. As mentioned above, an interesting future direction is exploring if and how PRN can generalize well on the results of panoptic models different than the one it is trained on, potentially by training it on a variety of base networks.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *TPAMI*, 39(12):2481–2495, 2017.

[2] Shubhankar Borse, Hyojin Park, Hong Cai, Debasmit Das, Risheek Garrepalli, and Fatih Porikli. Panoptic, instance and semantic relations: A relational context encoder to enhance panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1269–1279, 2022.

[3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2018.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229, 2020.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE TPAMI*, 40(4):834–848, 2017.

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018.

[7] Pengguang Chen. Gridmask data augmentation. *arXiv preprint arXiv:2001.04086*, 2020.

[8] Yifeng Chen, Guangchen Lin, Songyuan Li, Omar Bourahla, Yiming Wu, Fangfang Wang, Junyi Feng, Mingliang Xu, and Xi Li. Banet: Bidirectional aggregation network with occlusion handling for panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3793–3802, 2020.

[9] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12475–12485, 2020.

[10] Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8890–8899, 2020.

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

[12] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaiqi Huang. SSAP: Single-shot instance segmentation with affinity pyramid. In *International Conference on Computer Vision*, pages 642–651, 2019.

[13] Naiyu Gao, Yanhu Shan, Xin Zhao, and Kaiqi Huang. Learning category-and instance-aware pixel embedding for fast panoptic segmentation. *IEEE Transactions on Image Processing*, 30:6013–6023, 2021.

[14] Golnaz Ghiasi and Charless C Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *ECCV*, pages 519–534. Springer, 2016.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[17] Rui Hou, Jie Li, Arjun Bhargava, Allan Raventos, Vitor Guizilini, Chao Fang, Jerome Lynch, and Adrien Gaidon. Real-time panoptic segmentation from dense detections. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8523–8532, 2020.

[18] Jaedong Hwang, Seoung Wug Oh, Joon-Young Lee, and Bohyung Han. Exemplar-based open-set panoptic segmentation network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1175–1184, 2021.

[19] Tommi Kerola, Jie Li, Atsushi Kanehira, Yasunori Kudo, Alexis Vallet, and Adrien Gaidon. Hierarchical lovász embeddings for proposal-free panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14413–14423, 2021.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[21] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019.

[22] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019.

[23] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

[24] Justin Lazarow, Kwonjoon Lee, Kunyu Shi, and Zhuowen Tu. Learning instance occlusion for panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10720–10729, 2020.

[25] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13906–13915, 2020.

[26] Qizhu Li, Xiaojuan Qi, and Philip HS Torr. Unifying training and inference for panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13328, 2020.

[27] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided

unified network for panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7026–7035, 2019.

[28] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Fully convolutional networks for panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 214–223, 2021.

[29] Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, Ping Luo, and Tong Lu. Panoptic segformer: Delving deeper into panoptic segmentation with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1280–1289, 2022.

[30] Guosheng Lin, Fayao Liu, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for dense prediction. *IEEE TPAMI*, 42(5):1228–1242, 2019.

[31] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *International Conference on Computer Vision*, pages 2980–2988, 2017.

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[34] Huanyu Liu, Chao Peng, Changqian Yu, Jingbo Wang, Xu Liu, Gang Yu, and Wei Jiang. An end-to-end network for panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6172–6181, 2019.

[35] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *NeurIPS*, 2018.

[36] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[37] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The Mapillary Vistas dataset for semantic understanding of street scenes. In *International Conference on Computer Vision*, pages 4990–4999, 2017.

[38] Lorenzo Porzi, Samuel Rota Bulo, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8277–8286, 2019.

[39] Lorenzo Porzi, Samuel Rota Bulo, and Peter Kontschieder. Improving panoptic segmentation at all scales. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7302–7311, 2021.

[40] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. Adaptis: Adaptive instance selection network. In *International Conference on Computer Vision*, pages 7355–7363, 2019.

[41] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248, 2017.

[42] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully Convolutional One-Stage Object Detection. In *International Conference on Computer Vision*, 2019.

[43] Joseph Tighe, Marc Niethammer, and Svetlana Lazebnik. Scene parsing with object instances and occlusion ordering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2014.

[44] Zhuowen Tu, Xiangrong Chen, Alan L Yuille, and Song-Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of Computer Vision*, 63(2):113–140, 2005.

[45] Haochen Wang, Ruotian Luo, Michael Maire, and Greg Shakhnarovich. Pixel consensus voting for panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9464–9473, 2020.

[46] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. MaX-DeepLab: End-to-End Panoptic Segmentation With Mask Transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5463–5474, 2021.

[47] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation. In *European Conference on Computer Vision*, 2020.

[48] Mark Weber, Jonathon Luiten, and Bastian Leibe. Single-shot panoptic segmentation. In *IROS*, 2020.

[49] Yangxin Wu, Gengwei Zhang, Yiming Gao, Xiajun Deng, Ke Gong, Xiaodan Liang, and Liang Lin. Bidirectional graph reasoning network for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9080–9089, 2020.

[50] Yangxin Wu, Gengwei Zhang, Hang Xu, Xiaodan Liang, and Liang Lin. Auto-panoptic: Cooperative multi-component architecture search for panoptic segmentation. In *Advances in Neural Information Processing Systems*, 2020.

[51] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8818–8826, 2019.

[52] Yibo Yang, Hongyang Li, Xia Li, Qijie Zhao, Jianlong Wu, and Zhouchen Lin. SOGNet: Scene Overlap Graph Network for Panoptic Segmentation. In *AAAI*, 2020.

[53] Jian Yao, Sanja Fidler, and Raquel Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 702–709, 2012.

[54] Qihang Yu, Huiyu Wang, Dahun Kim, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille,

and Liang-Chieh Chen. Cmt-deeplab: Clustering mask transformers for panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2560–2570, 2022.

[55] Yuhui Yuan, Jingyi Xie, Xilin Chen, and Jingdong Wang. SegFix: Model-agnostic boundary refinement for segmentation. In *European Conference on Computer Vision*, pages 489–506, 2020.

[56] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. CANet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5217–5226, 2019.

[57] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9759–9768, 2020.