Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation

Bowen Cheng[1,2], Maxwell D. Collins[2], Yukun Zhu[2], Ting Liu[2],
Thomas S. Huang[1], Hartwig Adam[2], Liang-Chieh Chen[2]

[1]UIUC [2]Google Research

Abstract

*In this work, we introduce Panoptic-DeepLab, a simple, strong, and fast system for panoptic segmentation, aiming to establish a solid baseline for bottom-up methods that can achieve comparable performance of two-stage methods while yielding fast inference speed. In particular, Panoptic-DeepLab adopts the dual-ASPP and dual-decoder structures specific to semantic, and instance segmentation, respectively. The semantic segmentation branch is the same as the typical design of any semantic segmentation model (*e.g.*, DeepLab), while the instance segmentation branch is class-agnostic, involving a simple instance center regression. As a result, our single Panoptic-DeepLab simultaneously ranks first at all three Cityscapes benchmarks, setting the new state-of-art of 84.2% mIoU, 39.0% AP, and 65.5% PQ on test set. Additionally, equipped with MobileNetV3, Panoptic-DeepLab runs nearly in real-time with a single* 1025×2049 *image (15.8 frames per second), while achieving a competitive performance on Cityscapes (54.1 PQ% on test set). On Mapillary Vistas test set, our ensemble of six models attains 42.7% PQ, outperforming the challenge winner in 2018 by a healthy margin of 1.5%. Finally, our Panoptic-DeepLab also performs on par with several top-down approaches on the challenging COCO dataset. For the first time, we demonstrate a bottom-up approach could deliver state-of-the-art results on panoptic segmentation.*

1. Introduction

Panoptic segmentation, unifying semantic segmentation and instance segmentation, has received a lot of attention thanks to the recently proposed panoptic quality metric [34] and associated recognition challenges [46, 16, 53]. The goal of panoptic segmentation is to assign a unique value, encoding both semantic label and instance id, to every pixel in an image. It requires identifying the class and extent of each individual 'thing' in the image, and labelling all pixels that belong to each 'stuff' class.

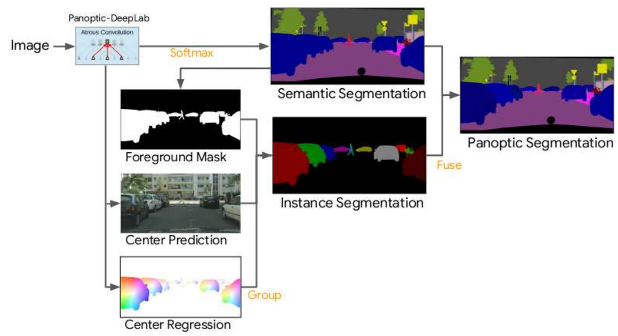


Figure 1. Our Panoptic-DeepLab predicts three outputs: semantic segmentation, instance center prediction and instance center regression. Class-agnostic instance segmentation, obtained by grouping predicted foreground pixels to their closest predicted instance centers, is then fused with semantic segmentation by majority-vote rule to generate final panoptic segmentation.

The task of panoptic segmentation introduces challenges that preceding methods are unsuited to solve. Models typically used in the separate instance and semantic segmentation literature have diverged, and fundamentally different approaches dominate in each setting. For panoptic segmentation, the top-down methods [74, 33, 40, 43, 60], attaching another semantic segmentation branch to Mask R-CNN [25], generate overlapping instance masks as well as duplicate pixel-wise semantic predictions. To settle the conflict, the commonly employed heuristic resolves overlapping instance masks by their predicted confidence scores [34], or even by the pairwise relationship between categories [43] (*e.g.*, *ties* should be always in front of *person*). Additionally, the discrepancy between semantic and instance segmentation results are sorted out by favoring the instance predictions. Though effective, it may be hard to implement the hand-crafted heuristics in a fast and parallel fashion. Another effective way is to develop advanced modules to fuse semantic and instance segmentation results [43, 40, 74]. However, these top-down methods are usually slow in speed, resulted from the multiple sequential processes in the pipeline.

On the other hand, bottom-up methods naturally resolve the conflict by predicting non-overlapping segments. Only few works [75, 22] adopt the bottom-up approach, which typically starts with a semantic segmentation prediction followed by grouping operations to generate instance masks. Tackling panoptic segmentation in such a sequential order allows a simple and fast scheme, such as majority vote [75], to merge semantic and instance segmentation results. Although obtaining promising fast inference speed, bottom-up approaches still demonstrate inferior performance compared to top-down ones prevailing in public benchmarks [46, 16, 53].

The difficulties faced by top-down methods, and the dearth of previous investigations into complementary approaches motivate us to establish a simple, strong, and fast bottom-up baseline for panoptic segmentation. Our proposed **Panoptic-DeepLab** (Fig. 1) requires only three loss functions during training, and introduces extra marginal parameters as well as additional slight computation overhead when building on top of a modern semantic segmentation model. The design of the proposed Panoptic-DeepLab is conceptually simple, adopting dual-ASPP and dual-decoder modules specific to semantic segmentation and instance segmentation, respectively. The semantic segmentation branch follows the typical design of any semantic segmentation model (*e.g*., DeepLab [11]), while the instance segmentation branch involves a simple instance center regression [4, 30], where the model learns to predict instance centers as well as the offset from each pixel to its corresponding center, resulting in an extremely simple grouping operation by assigning pixels to their closest predicted center. Additionally, with fast GPU implementation of the merging operation, Panoptic-DeepLab delivers near real-time end-to-end panoptic segmentation prediction.

We conduct experiments on several popular panoptic segmentation datasets. On Cityscapes test set [16], a *single* Panoptic-DeepLab model (without fine-tuning on different tasks) achieves state-of-the-art performance of 65.5% PQ, 39.0% AP, and 84.2% mIoU, simultaneously ranking first on *all* three Cityscapes tasks when comparing with published works. On Mapillary Vistas [53], our best *single* model attains 40.6% PQ on val set, while employing an ensemble of 6 models reaches a performance of 42.2% PQ on val set and 42.7% PQ on test set, outperforming the winner of Mapillary Vistas Panoptic Segmentation Challenge in 2018 by a healthy margin of 1.5% PQ. For the first time, we show a bottom-up approach could deliver state-of-the-art panoptic segmentation results on both Cityscapes and Mapillary Vistas. On COCO [46] test-dev set, our Panoptic-DeepLab also demonstrates state-of-the-art results, performing on par with several top-down approaches. Finally, we provide extensive experimental results and disclose every detail in our system. We hope our Panoptic-DeepLab could serve as a solid baseline to facilitate the research on panoptic segmentation, especially from the bottom-up perspective.

2. Related Works

We categorize current panoptic segmentation methods [34] into two groups: top-down and bottom-up approaches.

Top-down: Most state-of-the-art methods tackle panoptic segmentation from the top-down or proposal-based perspective. These methods are often referred to as two-stage methods because they require an additional stage to generate proposals. Specifically, Mask R-CNN [25] is commonly deployed to extract *overlapping instances*, followed by some post-processing methods to resolve mask overlaps. The remaining regions are then filled by a light-weight stuff segmentation branch. For example, TASCNet [40] learns a binary mask to enforce the consistency between 'thing' and 'stuff' predictions. Liu *et al*. [52] propose the Spatial Ranking module to resolve the overlapping instance masks. AUNet [43] introduces attention modules to guide the fusion between 'thing' and 'stuff' segmentation. Panoptic FPN [33] endows Mask R-CNN [25] with a semantic segmentation branch. UPSNet [74] develops a parameter-free panoptic head which resolves the conflicts in 'thing'-'stuff' fusion by predicting an extra unknown class. Porzi *et al*. [60] integrate the multi-scale features from FPN [45] with a light-weight DeepLab-inspired module [9]. AdaptIS [66] generates instance masks with point proposals.

Bottom-up: On the other hand, there are few bottom-up or proposal-free methods for panoptic segmentation. These works typically get the semantic segmentation prediction before detecting instances by grouping 'thing' pixels into clusters. The first bottom-up approach, DeeperLab [75], adopts bounding box corners as well as object centers for class-agnostic instance segmentation, coupled with DeepLab semantic segmentation outputs [8, 10]. Recently, SSAP [22] proposes to group pixels based on a pixel-pair affinity pyramid [51] with an efficient graph partition method [31]. Unfortunately, given its simplicity (*i.e*., a single pass of the system for prediction), bottom-up approaches perform inferiorly to top-down methods at almost *all* public benchmarks. In this work, we aim to push the envelope of bottom-up approaches. We note that there are several instance segmentation works [77, 68, 76, 2, 48, 35, 56, 20, 18, 44, 37, 30, 51, 54, 6], which could be potentially extended to bottom-up panoptic segmentation. Additionally, our method bears a similarity to Hough-Voting-based methods [4, 39, 21, 5] and recent works by Kendall *et al*. [30], Uhrig *et al*. [69] and Neven *et al*. [54] in the sense that our class-agnostic instance segmentation is obtained by regressing foreground pixels to their centers. However, our method

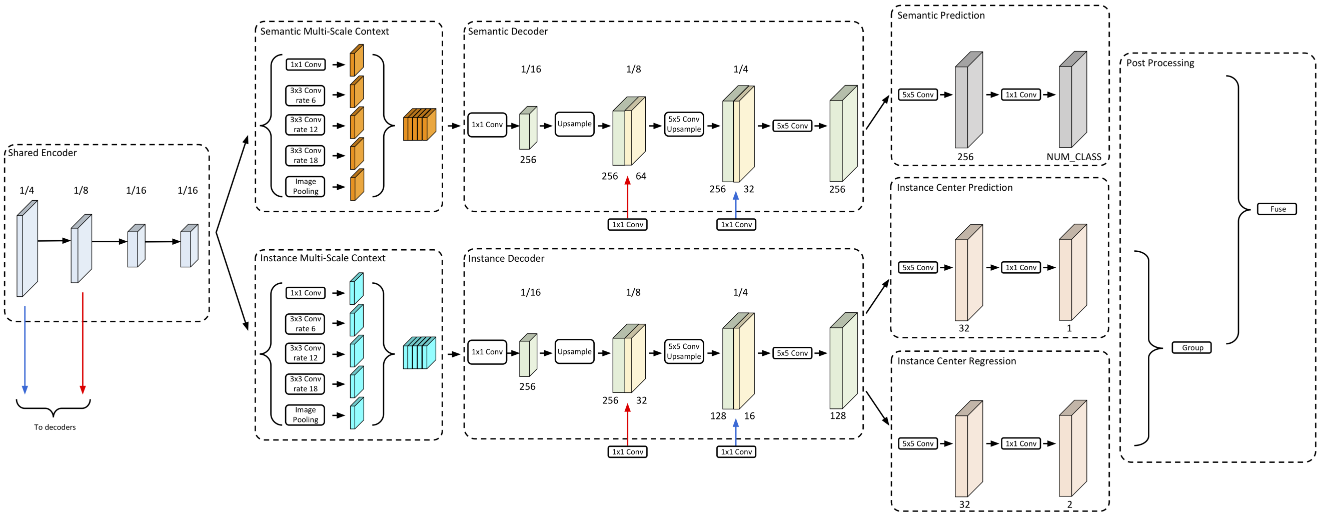


Figure 2. Our Panoptic-DeepLab adopts dual-context and dual-decoder modules for semantic segmentation and instance segmentation predictions. We apply atrous convolution in the last block of a network backbone to extract denser feature map. The Atrous Spatial Pyramid Pooling (ASPP) is employed in the context module as well as a light-weight decoder module consisting of a single convolution during each upsampling stage. The instance segmentation prediction is obtained by predicting the object centers and regressing every foreground pixel (*i.e.*, pixels with predicted 'thing' class) to their corresponding center. The predicted semantic segmentation and class-agnostic instance segmentation are then fused to generate the final panoptic segmentation result by the "majority vote" proposed by DeeperLab.

is even simpler than theirs: we directly predict the instance center locations and group pixels to their closest predicted centers. As a result, our method does not require the clustering method OPTICS [1] used in [30], or the advanced clustering loss function proposed in [54]. Finally, our model employs the parallel multi-head prediction framework similar to [68, 36, 55].

Keypoint representation: Recently, keypoint representations have been used for instance segmentation and object detection. Newell *et al.* [56] group pixels by embedding vectors. PersonLab [58] generates person segmentation masks and groups them into instances by learning offset to their detected keypoints. CornerNet [38] detects objects by predicting paired corners and group corners based on [56]. ExtremeNet [79] groups 'extreme points' [57] according to the relation to a center point. Zhou *et al.* [78] and Duan *et al.* [19] exploit instance centers for object detection. Following the same direction, we represent each instance by its center and take a step further by showing that such a simple representation is able to achieve state-of-the-art panoptic segmentation results on several challenging datasets. Different from keypoint-based detection, our Panoptic-DeepLab only requires class-agnostic object center prediction.

3. Panoptic-DeepLab

As shown in Fig. 2, our proposed Panoptic-DeepLab is deployed in a bottom-up and single-shot manner.

3.1. Architecture

Panoptic-DeepLab consists of four components: (1) an encoder backbone shared for both semantic segmentation and instance segmentation, (2) decoupled ASPP modules and (3) decoupled decoder modules specific to each task, and (4) task-specific prediction heads.

Basic architecture: The encoder backbone is adapted from an ImageNet-pretrained neural network paired with atrous convolution for extracting denser feature maps in its last block. Motivated by [14, 13, 54], we employ separate ASPP and decoder modules for semantic segmentation and instance segmentation, respectively, based on the hypothesis that those two branches requires different contextual and decoding information, which is empirically verified in the following section. Our light-weight decoder module follows DeepLabV3+ [11] with two modifications: (1) we introduce an additional low-level feature with output stride 8 to the decoder, thus the spatial resolution is gradually recovered by a factor of 2, and (2) in each upsampling stage we apply a *single* 5×5 depthwise-separable convolution [28].

Semantic segmentation head: We employ the weighted bootstrapped cross entropy loss, proposed in [75], for semantic segmentation, predicting both 'thing' and 'stuff' classes. The loss improves over bootstrapped cross entropy loss [72, 7, 59] by weighting each pixel differently.

Class-agnostic instance segmentation head: Motivated by Hough Voting [4, 30], we represent each object instance by its center of mass. For every foreground pixel (*i.e.*, pixel whose class is a 'thing'), we further predict the offset to its corresponding mass center. During training, groundtruth instance centers are encoded by a 2-D Gaussian with standard deviation of 8 pixels [67]. In particular, we adopt the Mean Squared Error (MSE) loss to minimize the distance between predicted heatmaps and 2D Gaussian-encoded groundtruth heatmaps. We use L_1 loss for the offset prediction, which is only activated at pixels belonging to object instances. During inference, predicted foreground pixels (obtained by filtering out background 'stuff' regions from semantic segmentation prediction) are grouped to their closest predicted mass center, forming our class-agnostic instance segmentation results, as detailed below.

3.2. Panoptic Segmentation

During inference, we use an extremely simple grouping operation to obtain instance masks, and a highly efficient majority voting algorithm to merge semantic and instance segmentation into final panoptic segmentation.

Simple instance representation: We simply represent each object by its center of mass, $\{\mathcal{C}_n : (i_n, j_n)\}$. To obtain the center point prediction, we first perform a keypoint-based non-maximum suppression (NMS) on the instance center heatmap prediction, essentially equivalent to applying max pooling on the heatmap prediction and keeping locations whose values do not change before and after max pooling. Finally, a hard threshold is used to filter out predictions with low confidence, and only locations with top-k highest confidence scores are kept. In experiments, we use max-pooling with kernel size 7, threshold 0.1, and $k = 200$.

Simple instance grouping: To obtain the instance id for each pixel, we use a simple instance center regression. For example, consider a predicted 'thing' pixel at location (i, j), we predict an offset vector $\mathcal{O}(i, j)$ to its instance center. $\mathcal{O}(i, j)$ is a vector with two elements, representing the offset in horizontal and vertical directions, respectively. The instance id for the pixel is thus the index of the closest instance center after moving the pixel location (i, j) by the offset $\mathcal{O}(i, j)$. That is,

$$\hat{k}_{i,j} = \underset{k}{\operatorname{argmin}}\, ||\mathcal{C}_k - ((i, j) + \mathcal{O}(i, j))||^2$$

where $\hat{k}_{i,j}$ is the predicted instance id for pixel at (i, j).

We use semantic segmentation prediction to filter out 'stuff' pixels whose instance id are always set to 0.

Efficient merging: Given the predicted semantic segmentation and class-agnostic instance segmentation results, we adopt a fast and parallelizable method to merge the results, following the "majority vote" principle proposed in DeeperLab [75]. In particular, the semantic label of a predicted instance mask is inferred by the majority vote of the corresponding predicted semantic labels. This operation is essentially accumulating the class label histograms, and thus is efficiently implemented in GPU, which takes only 3 ms when operating on a 1025×2049 input.

3.3. Instance Segmentation

Panoptic-DeepLab can also generate instance segmentation predictions as a by-product. To properly evaluate the instance segmentation results, one needs to associate a confidence score with each predicted instance mask. Previous bottom-up instance segmentation methods use some heuristics to obtain the confidence scores. For example, DWT [2] and SSAP [22] use an average of semantic segmentation scores for some easy classes and use random scores for other harder classes. Additionally, they remove masks whose areas are below a certain threshold for each class. On the other hand, our Panoptic-DeepLab does not adopt any heuristic or post processing for instance segmentation. Motivated by YOLO [62], we compute the class-specific confidence score for each instance mask as

$$Score(Objectness) \times Score(Class)$$

where $Score(Objectness)$ is unnormalized objectness score obtained from the class-agnostic center point heatmap, and $Score(Class)$ is obtained from the average of semantic segmentation predictions within the predicted mask region.

4. Experiments

Cityscapes [16]: The dataset consists of 2975, 500, and 1525 traffic-related images for training, validation, and testing, respectively. It contains 8 'thing' and 11 'stuff' classes.

Mapillary Vistas [53]: A large-scale traffic-related dataset, containing 18K, 2K, and 5K images for training, validation and testing, respectively. It contains 37 'thing' classes and 28 'stuff' classes in a variety of image resolutions, ranging from 1024×768 to more than 4000×6000

COCO [46]: There are 118K, 5K, and 20K images for training, validation, and testing, respectively. The dataset consists of 80 'thing' and 53 'stuff' classes.

Experimental setup: We report mean IoU, average precision (AP), and panoptic quality (PQ) to evaluate the semantic, instance, and panoptic segmentation results.

All our models are trained using TensorFlow on 32 TPUs. We adopt a similar training protocol as in [11]. In particular, we use the 'poly' learning rate policy [50] with an initial learning rate of 0.001, fine-tune the batch normalization [29] parameters, perform random scale data augmentation during training, and optimize with Adam [32] *without weight decay*. On Cityscapes, our best setting is obtained by training with whole image (*i.e.*, crop size equal to 1025×2049) with batch size 32. On Mapillary Vistas, we resize the images to 2177 pixels at the longest side to handle the large input variations, and randomly crop 1025×1025 patches during training with batch size 64. On COCO, we resize the images to 1025 pixels at the longest side and train our models with crop size

Adam	MSE	De. x2	ASPP x2	L-Crop	$C_{Sem} = 256$	$C_{Ins} = 256$	Sem. Only	PQ (%)	AP (%)	mIoU (%)	Params (M)	M-Adds (B)
								60.3	32.7	78.2	41.85	496.84
✓								61.0	34.3	79.4	41.85	496.84
✓	✓							61.8	33.8	78.6	41.85	496.84
✓	✓	✓						60.8	32.7	79.0	41.93	501.88
✓	✓	✓	✓					62.5	33.9	78.7	43.37	517.17
✓	✓	✓	✓	✓				62.7	34.5	79.6	43.37	517.17
✓	✓	✓	✓	✓	✓			**63.0**	**35.3**	**80.5**	46.72	547.49
✓	✓	✓	✓	✓	✓	✓		62.1	35.1	80.3	46.88	573.86
✓				✓	✓		✓	-	-	80.3	43.60	518.84

Table 1. Ablation studies on Cityscapes *val* set. **Adam**: Adam optimizer. **MSE**: MSE loss for instance center. **De. x2**: Dual decoder. **ASPP x2**: Dual ASPP. **L-Crop**: Large crop size. $\mathbf{C_{Sem}} = 256$: 256 (instead of 128) channels in semantic segmentation branch. $\mathbf{C_{Ins}} = 256$: 256 (instead of 128) channels in instance segmentation branch. **Sem. Only**: Only semantic segmentation. M-Adds are measured w.r.t. a 1025×2049 input.

Method	Extra Data	Flip	MS	PQ (%)	AP (%)	mIoU (%)
w/o Extra Data						
TASCNet [40]				55.9	-	-
Panoptic FPN [33]				58.1	33.0	75.7
AUNet [43]				59.0	34.4	75.6
UPSNet [74]				59.3	33.3	75.2
UPSNet [74]		✓	✓	60.1	33.3	76.8
Seamless [60]				60.3	33.6	77.5
AdaptIS [66]		✓		62.0	36.3	79.2
DeeperLab [75]				56.5	-	-
SSAP [22]		✓	✓	61.1	37.3	-
Panoptic-DeepLab				63.0	35.3	80.5
Panoptic-DeepLab		✓		63.4	36.1	80.9
Panoptic-DeepLab		✓	✓	**64.1**	**38.5**	**81.5**
w/ Extra Data						
TASCNet [40]	COCO			59.3	37.6	78.1
TASCNet [40]	COCO	✓	✓	60.4	39.1	78.7
UPSNet [74]	COCO			60.5	37.8	77.8
UPSNet [74]	COCO	✓	✓	61.8	39.0	79.2
Seamless [60]	MV			65.0	-	80.7
Panoptic-DeepLab	MV			65.3	38.8	82.5
Panoptic-DeepLab	MV	✓		65.6	39.4	82.6
Panoptic-DeepLab	MV	✓	✓	**67.0**	**42.5**	**83.1**

Table 2. Cityscapes *val* set. **Flip:** Adding left-right flipped inputs. **MS:** Multiscale inputs. **MV:** Mapillary Vistas.

Method	Extra Data	PQ (%)	AP (%)	mIoU (%)
Semantic Segmentation				
GFF-Net [42]		-	-	82.3
Zhu *et al*. [80]	C, V, MV	-	-	83.5
Hyundai Mobis AD Lab	C, MV	-	-	83.8
Instance Segmentation				
AdaptIS [66]		-	32.5	-
UPSNet [74]	COCO	-	33.0	-
PANet [49]	COCO	-	36.4	-
Sogou_MM	COCO	-	37.2	-
iFLYTEK-CV	COCO	-	38.0	-
NJUST	COCO	-	38.9	-
AInnoSegmentation	COCO	-	**39.5**	-
Panoptic Segmentation				
SSAP [22]		58.9	32.7	-
TASCNet [40]	COCO	60.7	-	-
Seamless [60]	MV	62.6	-	-
Panoptic-DeepLab		62.3	34.6	79.4
Panoptic-DeepLab	MV	**65.5**	39.0	**84.2**

Table 3. Cityscapes *test* set. **C:** Cityscapes coarse annotation. **V:** Cityscapes video. **MV:** Mapillary Vistas.

1025×1025 with batch size 64. We set training iterations to 60K, 150K, and 200K for Cityscapes, Mapillary Vistas, and COCO, respectively. During evaluation, due to the sensitivity of PQ [74, 40, 60], we re-assign to 'VOID' label all 'stuff' segments whose areas are smaller than a threshold. The thresholds on Cityscapes, Mapillary Vistas, and COCO are 2048, 4096, and 4096, respectively. Additionally, we adopt multi-scale inference (scales equal to $\{0.5, 0.75, 1, 1.25, 1.5, 1.75, 2\}$ for Cityscapes and Mapillary Vistas and $\{0.5, 0.75, 1, 1.25, 1.5\}$ for COCO) and left-right flipped inputs, to further improve the performance. For all the reported results, unless specified, Xception-71 [15, 61, 11] is employed as the backbone.

Panoptic-DeepLab is trained with three loss functions: weighted bootstrapped cross entropy loss for semantic segmentation head ($\mathcal{L}_{sem}$) [75]; MSE loss for center heatmap head ($\mathcal{L}_{heatmap}$) [67]; and L1 loss for center offset head ($\mathcal{L}_{offset}$) [58]. The final loss $\mathcal{L}$ is computed as follows.

$$\mathcal{L} = \lambda_{sem}\mathcal{L}_{sem} + \lambda_{heatmap}\mathcal{L}_{heatmap} + \lambda_{offset}\mathcal{L}_{offset}$$

Specifically, we set $\lambda_{sem} = 3$ for pixels belonging to instances with an area smaller than 64×64 and $\lambda_{sem} = 1$ everywhere else, following DeeperLab [75]. To make sure the losses are in the similar magnitude, we set $\lambda_{heatmap} = 200$ and $\lambda_{offset} = 0.01$.

4.1. Ablation Studies

We conduct ablation studies on Cityscapes validation set, as shown in Tab. 1. Replacing SGD momentum optimizer with Adam optimizer yields 0.7% PQ improvement. Instead of using the sigmoid cross entropy loss for training the heatmap (*i.e.*, instance center prediction), it brings 0.8% PQ improvement by applying the Mean Squared Error (MSE) loss to minimize the distance between the predicted heatmap and the 2D Gaussian-encoded groundtruth heatmap. It is more effective to adopt both dual-decoder and dual-ASPP, which gives us 0.7% PQ improvement while maintaining similar AP and mIoU. Employing a large crop size 1025×2049 (instead of 513×1025) during training further improves the AP and mIoU by 0.6% and 0.9% respectively. Finally, increasing the feature channels from 128 to 256 in the semantic segmentation branch achieves our best result of 63.0% PQ, 35.3% AP, and 80.5% mIoU.

Multi-task learning: For reference, we train a Semantic-DeepLab under the same setting as the best Panoptic-DeepLab (last row of Tab. 1), showing that multi-task learning does not bring extra gain to mIoU. Note that Panoptic-DeepLab adds marginal parameters and small computation overhead over Semantic-DeepLab.

4.2. Cityscapes

Val set: In Tab. 2, we report our Cityscapes validation set results. When using only Cityscapes *fine* annotations, our best Panoptic-DeepLab, with multi-scale inputs and left-right flips, outperforms the best bottom-up approach, SSAP, by 3.0% PQ and 1.2% AP, and is better than the best proposal-based approach, AdaptIS, by 2.1% PQ, 2.2% AP, and 2.3% mIoU. When using extra data, our best Panoptic-DeepLab outperforms UPSNet by 5.2% PQ, 3.5% AP, and 3.9% mIoU, and Seamless by 2.0% PQ and 2.4% mIoU. Note that we do not exploit any other data, such as COCO, Cityscapes *coarse* annotations, depth, or video.

Test set: On the test set, we additionally employ the trick proposed in [11] that applies atrous convolution in the last two blocks within the backbone, with rate 2 and 4 respectively, during inference. This trick brings an extra 0.4% AP and 0.2% mIoU on *val* set but no improvement over PQ. We do not use this trick for the Mapillary Vistas Challenge. As shown in Tab. 3, our *single* unified Panoptic-DeepLab achieves state-of-the-art results, ranking first at *all* three Cityscapes tasks, when comparing with published works. Our model ranks second in the instance segmentation track when also taking into account unpublished entries.

4.3. Mapillary Vistas

Val set: In Tab. 4, we report Mapillary Vistas *val* set results. Our best *single* Panoptic-DeepLab model, with multi-scale inputs and left-right flips, outperforms the bottom-up approach, DeeperLab, by 8.3% PQ, and the top-down approach, Seamless, by 2.6% PQ. In Tab. 5, we report our results with three families of network backbones. We observe that naïve HRNet-W48 slightly under-performs Xception-71. Due to the diverse image resolutions in Mapillary Vistas, we found it important to enrich the context information as well as to keep high-resolution features. Therefore, we propose a simple modification for HRNet [70] and Auto-DeepLab [47]. For modified HRNet, called HRNet+, we keep its ImageNet-pretrained head and further attach dual-ASPP and dual-decoder modules. For modified Auto-DeepLab, called Auto-DeepLab+, we remove the stride in the original 1/32 branch (which improves PQ by 1%). To summarize, using Xception-71 strikes the best accuracy and speed trade-off, while HRNet-W48+ achieves the best PQ of 40.6%. Finally, our ensemble of six models attains a 42.2% PQ, 18.2% AP, and 58.7% mIoU.

Test set: Tab. 6 summarizes our Mapillary Vistas test set results along with other top-performing methods. Our entry [12] with an ensemble of six models attain a performance of 42.7% PQ, outperforming the winner of Mapillary Vistas Panoptic Segmentation Challenge in 2018 by 1.5% PQ.

Method	Flip	MS	PQ (%)	PQ^{Th} (%)	PQ^{St} (%)	AP (%)	mIoU (%)
TASCNet [40]			32.6	31.1	34.4	18.5	-
TASCNet [40]	✓	✓	34.3	**34.8**	33.6	**20.4**	-
AdaptIS [66]	✓		35.9	31.5	41.9	-	-
Seamless [60]			37.7	33.8	42.9	16.4	50.4
DeeperLab [75]			32.0	-	-	-	55.3
Panoptic-DeepLab			37.7	30.4	47.4	14.9	55.4
Panoptic-DeepLab	✓		38.0	30.6	47.9	15.2	55.8
Panoptic-DeepLab	✓	✓	**40.3**	33.5	**49.3**	17.2	**56.8**

Table 4. Mapillary Vistas *val* set. **Flip:** Adding left-right flipped inputs. **MS:** Multiscale inputs.

Backbone	Params (M)	M-Adds (B)	PQ (%)	AP (%)	mIoU (%)
Xception-65	44.31	1054.05	39.2	16.4	56.9
Xception-71	46.73	1264.32	40.3	17.2	56.8
HRNet-W48 [70]	71.66	2304.87	39.3	17.2	55.4
HRNet-W48+	88.87	2208.04	40.6	17.8	57.6
HRNet-W48+ (Atrous)	88.87	2972.02	40.5	17.7	57.4
HRNet-Wider+	60.05	1315.70	40.0	17.0	57.0
HRNet-Wider+ (Atrous)	60.05	1711.69	39.7	16.8	56.5
Auto-DeepLab-L+	41.54	1493.78	39.3	15.8	56.9
Auto-DeepLab-XL+	71.98	2378.17	40.3	16.3	57.1
Auto-DeepLab-XL++	72.16	2386.81	40.3	16.9	57.6
Ensemble (top-6 models)	-	-	42.2	18.2	58.7

Table 5. Mapillary Vistas *val* set with different backbones. **HRNet-W48+:** Modified HRNet-W48 with ImageNet-pretraining head kept. **HRNet-W48+ (Atrous):** Additionally apply atrous convolution with rate 2 in the output stride 32 branch of HRNet. **HRNet-Wider+:** A wider version of HRNet using separable convolution with large channels. The ImageNet-pretraining head is also kept. **HRNet-Wider+ (Atrous):** Additionally apply atrous convolution with rate 2 in the output stride 32 branch. **Auto-DeepLab-L+:** Auto-DeepLab with $F = 48$ and remove the stride in the original output stride 32 path. **Auto-DeepLab-XL+:** Auto-DeepLab with $F = 64$ and remove the stride in the original output stride 32 path. **Auto-DeepLab-XL++:** Additionally exploit low-level features from output stride 8 endpoint in the decoder module. We employ dual-ASPP and dual-decoder modules for all model variants except **HRNet-W48** which follows the original design in [70]. Results are obtained with multi-scale and left-right flipped inputs. M-Adds are measured w.r.t. a 2177×2177 input.

Method	PQ	SQ	RQ	PQ^{Th}	SQ^{Th}	RQ^{Th}	PQ^{St}	SQ^{St}	RQ^{St}
DeeperLab [75]	31.6	75.5	40.1	25.0	73.4	33.1	40.3	78.3	49.3
AdaptIS [66]	36.8	76.0	46.3	33.3	75.2	42.6	41.4	77.1	51.3
TRI-ML (2018: 2^{nd})	38.7	78.1	48.4	39.0	79.7	48.9	38.2	75.9	47.9
Team R4D (2018: 1^{st})	41.2	79.1	50.8	37.9	79.7	47.1	45.6	78.4	55.8
Panoptic-DeepLab	42.7	78.1	52.5	35.9	75.3	46.0	51.6	81.9	61.2

Table 6. Performance on Mapillary Vistas *test* set.

Method	Backbone	Flip	MS	PQ (%)	PQ$^{\text{Th}}$ (%)	PQ$^{\text{St}}$ (%)
AUNet [43]	ResNet-50 [26]			39.6	49.1	25.2
Panoptic-FPN [33]	ResNet-101			40.3	47.5	29.5
AdaptIS [66]	ResNeXt-101 [73]	✓		42.3	49.2	31.8
UPSNet [74]	ResNet-50			42.5	48.5	33.4
Detectron2 [71]	ResNet-101			43.0	-	-
UPSNet [74]	ResNet-50	✓	✓	43.2	49.1	34.1
DeeperLab [75]	Xception-71			33.8	-	-
SSAP [22]	ResNet-101	✓	✓	36.5	-	-
Panoptic-DeepLab	Xception-71			39.7	43.9	33.2
Panoptic-DeepLab	Xception-71	✓		40.2	44.4	33.8
Panoptic-DeepLab	Xception-71	✓	✓	41.2	44.9	35.7

Table 7. COCO *val* set. **Flip:** Adding left-right flipped inputs. **MS:** Multiscale inputs.

Method	Backbone	Flip	MS	PQ (%)	PQ$^{\text{Th}}$ (%)	PQ$^{\text{St}}$ (%)
TASCNet [40]	ResNet-50			40.7	47.0	31.0
Panoptic-FPN [33]	ResNet-101			40.9	48.3	29.7
AdaptIS [66]	ResNeXt-101	✓		42.8	53.2	36.7
AUNet [43]	ResNeXt-152			46.5	55.8	32.5
UPSNet [74]	DCN-101 [17]	✓	✓	46.6	53.2	36.7
DeeperLab [75]	Xception-71			34.3	37.5	29.6
SSAP [22]	ResNet-101	✓	✓	36.9	40.1	32.0
Panoptic-DeepLab	Xception-71	✓	✓	41.4	45.1	35.9

Table 8. COCO *test-dev* set. **Flip:** Adding left-right flipped inputs. **MS:** Multiscale inputs.

4.4. COCO

Val set: In Tab. 7, we report COCO *val* set result. With a single scale inference, our Panoptic-DeepLab outperforms the previous best bottom-up SSAP by 3.2% PQ and DeeperLab [75] by 5.9% PQ. With multi-scale inference and horizontal flip, Panoptic-DeepLab achieves 41.2% PQ, setting a new state-of-the-art performance for bottom-up methods, and performing comparably with top-down methods.

Test-dev set: In Tab. 8, we report COCO *test-dev* set result. Our Panoptic-DeepLab is 4.5% PQ better than the previous best bottom-up SSAP on COCO and our 41.4% PQ is comparable to most top-down methods without using heavier backbone [73] or deformable convolution [17].

4.5. Runtime

In Tab. 9, we report the end-to-end runtime (*i.e.*, inference time from an input image to final panoptic segmentation, including *all* operations such as merging semantic and instance segmentation) of Panoptic-DeepLab with three different network backbones (MobileNetV3 [27], ResNet-50 [26], and Xception-71 [15, 61]) on all three datasets. The inference speed is measured on a Tesla V100-SXM2 GPU *with batch size of one*. We further plot the speed-accuracy trade-off curve in Fig. 3. Our Panoptic-DeepLab achieves the best trade-off across all three datasets.

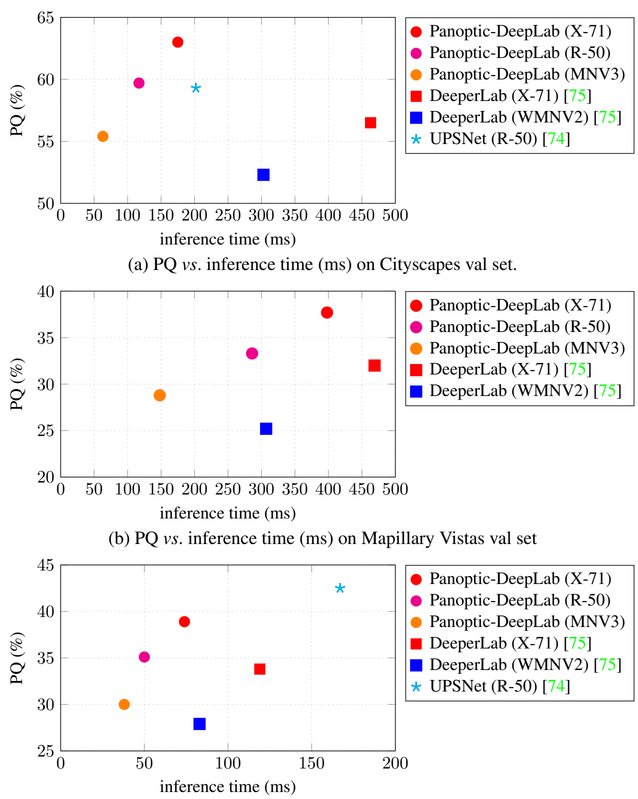
4.6. Discussion

Herein, we list a few interesting aspects in the hope of inspiring future works on bottom-up panoptic segmentation.

Scale variation: Fig. 4 shows visualization of Panoptic-DeepLab. In particular, the cross road (in last 2 rows), with

Method	Backbone	Input Size	PQ [val]	PQ [test]	Speed (ms)	M-Adds (B)
Cityscapes						
DeeperLab [75]	W-MNV2 [64]	1025 × 2049	52.3	-	303	-
DeeperLab [75]	Xception-71	1025 × 2049	56.5	-	463	-
UPSNet [74]	ResNet-50	1024 × 2048	59.3	-	202	-
Panoptic-DeepLab	MNV3	1025 × 2049	55.4	54.1	63	54.17
Panoptic-DeepLab	ResNet-50	1025 × 2049	59.7	58.0	117	381.39
Panoptic-DeepLab	Xception-71	1025 × 2049	63.0	60.7	175	547.49
Mapillary Vistas						
DeeperLab [75]	W-MNV2	1441 × 1441	25.2	25.3	307	-
DeeperLab [75]	Xception-71	1441 × 1441	32.0	31.6	469	-
Panoptic-DeepLab	MNV3	2177 × 2177	28.8	-	148	138.12
Panoptic-DeepLab	ResNet-50	2177 × 2177	33.3	-	286	910.47
Panoptic-DeepLab	Xception-71	2177 × 2177	37.7	-	398	1264.32
COCO						
DeeperLab [75]	W-MNV2	641 × 641	27.9	28.1	83	-
DeeperLab [75]	Xception-71	641 × 641	33.8	34.3	119	-
UPSNet [74]	ResNet-50	800 × 1333	42.5	-	167	-
Panoptic-DeepLab	MNV3	641 × 641	30.0	29.8	38	12.24
Panoptic-DeepLab	ResNet-50	641 × 641	35.1	35.2	50	77.79
Panoptic-DeepLab	Xception-71	641 × 641	38.9	38.8	74	109.21
Panoptic-DeepLab	Xception-71	1025 × 1025	39.7	39.6	132	279.25

Table 9. End-to-end runtime, including merging semantic and instance segmentation. All results are obtained by (1) a single-scale input without flipping, and (2) built-in TensorFlow library without extra inference optimization. **MNV3:** MobileNet-V3. **PQ [val]:** PQ (%) on val set. **PQ [test]:** PQ (%) on test(-dev) set. Note the channels in last block of MNV3 are reduced by a factor of 2 [27].



(a) PQ *vs*. inference time (ms) on Cityscapes val set.

(b) PQ *vs*. inference time (ms) on Mapillary Vistas val set

(c) PQ *vs*. inference time (ms) on COCO val set

Figure 3. PQ *vs*. Seconds. Our Panoptic-DeepLab model variants attain a better speed/accuracy trade-off across challenging datasets. The inference time is measured *end-to-end* from input image to panoptic segmentation output. **X-71:** Xception-71. **R-50:** ResNet-50. **MNV3:** MobileNetV3. Data points from Tab. 9.

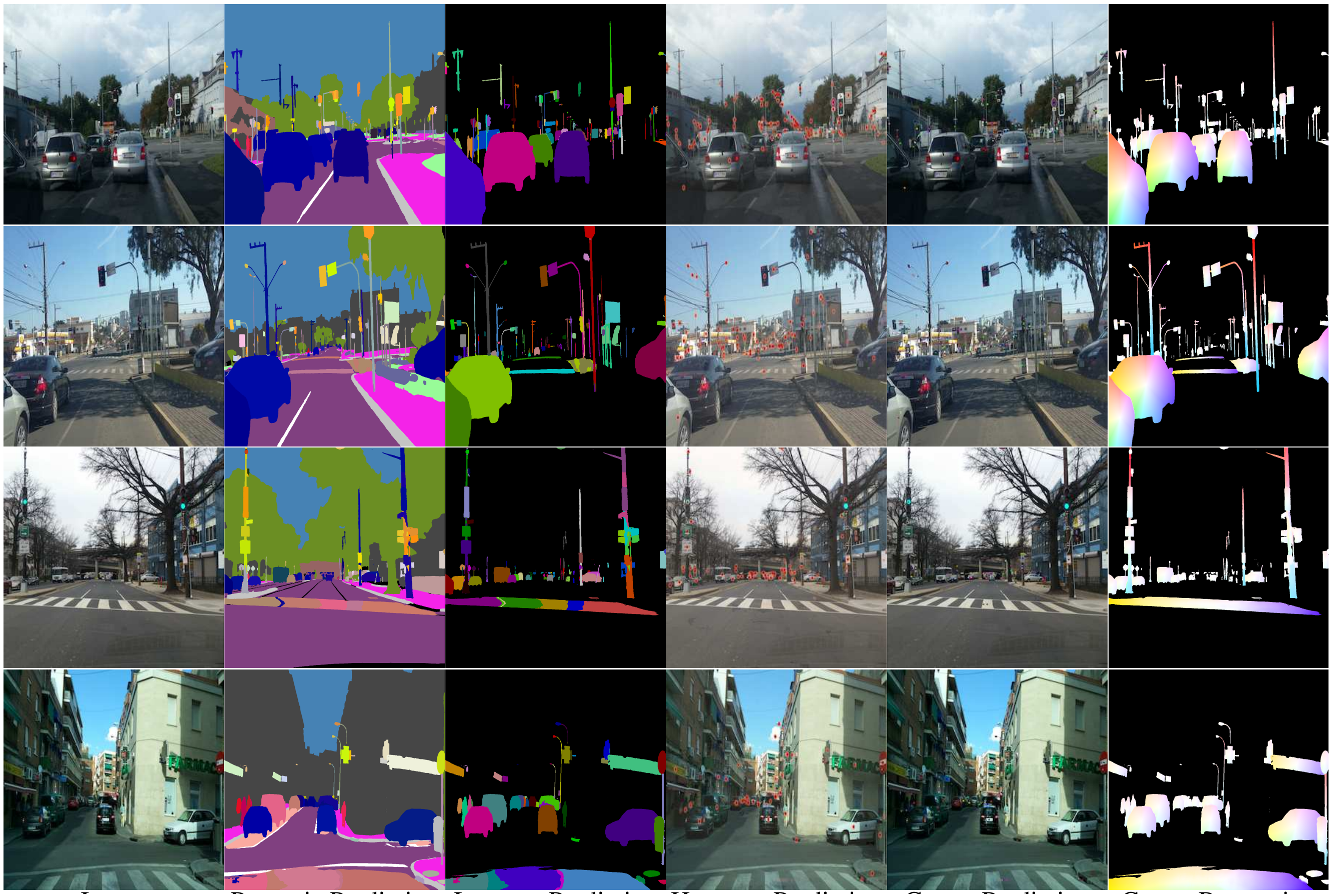


Figure 4. Visualization of Panoptic-DeepLab with Xception-71 on Mapillary Vistas *val* set. Only single scale inference is used and the model achieves 37.7% PQ. We encode 2D offset vectors into RGB values, same as [3]. The cross road in last 2 rows is segmented into multiple instances due to large scale variation. More visualizations are included in supplementary materials.

a large scale variation, is segmented into multiple small instances. On the other hand, top-down methods handle scale variation to some extent by the ROIPooling [23] or ROIAlign [25] operations which normalize regional features to a *canonical* scale [24, 63]. Additionally, incorporating scale-aware information to feature pyramid [45] or image pyramid [65] may improve the performance of bottom-up methods.

PQ^{Thing} *vs*. PQ^{Stuff}: As shown in Tab. 6 and Tab. 8, Panoptic-DeepLab has higher PQ^{Stuff} but lower PQ^{Thing} when compared with other top-down approaches which better handle instances of large scale variation as discussed above. Combining the best from both bottom-up and top-down approaches is thus interesting to explore but beyond the scope of current work.

Panoptic *vs*. instance annotations: Most bottom-up panoptic segmentation methods only exploit the panoptic annotations. We notice there are two types of annotations in the COCO dataset, panoptic annotations and instance annotations. The former do not allow overlapping masks (thus creating occlusions among masks), while the latter allows overlaps, which might make the training target easier to optimize, similar to amodal segmentation [81, 41].

End-to-end training: Current bottom-up panoptic segmentation methods still require some post-processing steps to obtain the final panoptic segmentation, which may make it hard to end-to-end train the whole system.

5. Conclusion

We have presented Panoptic-DeepLab, a simple, strong, and fast baseline for bottom-up panoptic segmentation. Panoptic-DeepLab is simple in design, requiring only three loss functions during training and adds marginal parameters to a modern semantic segmentation model. Panoptic-DeepLab is the first bottom-up and single-shot panoptic segmentation model that attains state-of-the-art performance on several public benchmarks, and delivers near real-time end-to-end inference speed. We hope our simple and effective model could establish a solid baseline and further benefit the research community.

References

[1] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. In *ACM Sigmod record*, 1999. 3

[2] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017. 2, 4

[3] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 2011. 8

[4] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 1981. 2, 3

[5] Olga Barinova, Victor Lempitsky, and Pushmeet Kholi. On detection of multiple object instances using hough transforms. *IEEE TPAMI*, 2012. 2

[6] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019. 2

[7] Samuel Rota Bulò, Gerhard Neuhold, and Peter Kontschieder. Loss maxpooling for semantic image segmentation. In *CVPR*, 2017. 3

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 2

[9] 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*, 2017. 2

[10] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017. 2

[11] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 2, 3, 4, 5, 6

[12] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab. In *ICCV COCO + Mapillary Joint Recognition Challenge Workshop*, 2019. 6

[13] Bowen Cheng, Yunchao Wei, Honghui Shi, Rogerio Feris, Jinjun Xiong, and Thomas Huang. Decoupled classification refinement: Hard false positive suppression for object detection. *arXiv:1810.04002*, 2018. 3

[14] Bowen Cheng, Yunchao Wei, Honghui Shi, Rogerio Feris, Jinjun Xiong, and Thomas Huang. Revisiting rcnn: On awakening the classification power of faster rcnn. In *ECCV*, 2018. 3

[15] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 5, 7

[16] 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 *CVPR*, 2016. 1, 2, 4

[17] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 7

[18] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv:1708.02551*, 2017. 2

[19] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, 2019. 3

[20] Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P Murphy. Semantic instance segmentation via deep metric learning. *arXiv:1703.10277*, 2017. 2

[21] Juergen Gall and Victor Lempitsky. Class-specific hough forests for object detection. In *CVPR*, 2009. 2

[22] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaiqi Huang. Ssap: Single-shot instance segmentation with affinity pyramid. In *ICCV*, 2019. 2, 4, 5, 7

[23] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 8

[24] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 8

[25] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1, 2, 8

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7

[27] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019. 7

[28] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*, 2017. 3

[29] Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 4

[30] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018. 2, 3

[31] Margret Keuper, Evgeny Levinkov, Nicolas Bonneel, Guillaume Lavoué, Thomas Brox, and Bjorn Andres. Efficient decomposition of image and mesh graphs by lifted multicuts. In *ICCV*, 2015. 2

[32] Diederik P Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *ICLR*, 2015. 4

[33] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 1, 2, 5, 7

[34] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 1, 2

[35] Alexander Kirillov, Evgeny Levinkov, Bjoern Andres, Bogdan Savchynskyy, and Carsten Rother. Instancecut: from edges to instances with multicut. In *CVPR*, 2017. 2

[36] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision

using diverse datasets and limited memory. In *CVPR*, 2017. 3

[37] Victor Kulikov, Victor Yurchenko, and Victor Lempitsky. Instance segmentation by deep coloring. *arXiv:1807.10007*, 2018. 2

[38] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 2018. 3

[39] Bastian Leibe, Ales Leonardis, and Bernt Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on statistical learning in computer vision, ECCV*, 2004. 2

[40] Jie Li, Allan Raventos, Arjun Bhargava, Takaaki Tagawa, and Adrien Gaidon. Learning to fuse things and stuff. *arXiv:1812.01192*, 2018. 1, 2, 5, 6, 7

[41] Ke Li and Jitendra Malik. Amodal instance segmentation. In *ECCV*, 2016. 8

[42] Xiangtai Li, Houlong Zhao, Lei Han, Yunhai Tong, and Kuiyuan Yang. Gff: Gated fully fusion for semantic segmentation. *arXiv:1904.01803*, 2019. 5

[43] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. In *CVPR*, 2019. 1, 2, 5, 7

[44] Xiaodan Liang, Liang Lin, Yunchao Wei, Xiaohui Shen, Jianchao Yang, and Shuicheng Yan. Proposal-free network for instance-level object segmentation. *IEEE TPAMI*, 2018. 2

[45] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2, 8

[46] 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 *ECCV*, 2014. 1, 2, 4

[47] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *CVPR*, 2019. 6

[48] Shu Liu, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. Sgn: Sequential grouping networks for instance segmentation. In *ICCV*, 2017. 2

[49] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018. 5

[50] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv:1506.04579*, 2015. 4

[51] Yiding Liu, Siyu Yang, Bin Li, Wengang Zhou, Jizheng Xu, Houqiang Li, and Yan Lu. Affinity derivation and graph merge for instance segmentation. In *ECCV*, 2018. 2

[52] Huanyu Liu1, Chao Peng, Changqian Yu, Jingbo Wang, Xu Liu, Gang Yu, and Wei Jiang. An end-to-end network for panoptic segmentation. In *CVPR*, 2019. 2

[53] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 1, 2, 4

[54] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *CVPR*, 2019. 2, 3

[55] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Fast scene understanding for autonomous driving. *arXiv:1708.02550*, 2017. 3

[56] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *NeurIPS*, 2017. 2, 3

[57] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *ICCV*, 2017. 3

[58] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *ECCV*, 2018. 3, 5

[59] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *CVPR*, 2017. 3

[60] Lorenzo Porzi, Samuel Rota Bulò, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In *CVPR*, 2019. 1, 2, 5, 6

[61] Haozhi Qi, Zheng Zhang, Bin Xiao, Han Hu, Bowen Cheng, Yichen Wei, and Jifeng Dai. Deformable convolutional networks – coco detection and segmentation challenge 2017 entry. *ICCV COCO Challenge Workshop*, 2017. 5, 7

[62] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 4

[63] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 8

[64] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 7

[65] Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection-snip. *CVPR*, 2018. 8

[66] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. Adaptis: Adaptive instance selection network. In *ICCV*, 2019. 2, 5, 6, 7

[67] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NeurIPS*, 2014. 4, 5

[68] Jonas Uhrig, Marius Cordts, Uwe Franke, and Thomas Brox. Pixel-level encoding and depth layering for instance-level semantic labeling. In *German Conference on Pattern Recognition*, 2016. 2, 3

[69] Jonas Uhrig, Eike Rehder, Björn Fröhlich, Uwe Franke, and Thomas Brox. Box2pix: Single-shot instance segmentation by assigning pixels to object boxes. In *IEEE Intelligent Vehicles Symposium (IV)*, 2018. 2

[70] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep

high-resolution representation learning for visual recognition. *arXiv:1908.07919*, 2019. 6

[71] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 7

[72] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Bridging category-level and instance-level semantic image segmentation. *arXiv:1605.06885*, 2016. 3

[73] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 7

[74] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *CVPR*, 2019. 1, 2, 5, 7

[75] Tien-Ju Yang, Maxwell D Collins, Yukun Zhu, Jyh-Jing Hwang, Ting Liu, Xiao Zhang, Vivienne Sze, George Papandreou, and Liang-Chieh Chen. Deeperlab: Single-shot image parser. *arXiv:1902.05093*, 2019. 2, 3, 4, 5, 6, 7

[76] Ziyu Zhang, Sanja Fidler, and Raquel Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *CVPR*, 2016. 2

[77] Ziyu Zhang, Alexander G Schwing, Sanja Fidler, and Raquel Urtasun. Monocular object instance segmentation and depth ordering with cnns. In *ICCV*, 2015. 2

[78] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv:1904.07850*, 2019. 3

[79] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, 2019. 3

[80] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *CVPR*, 2019. 5

[81] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *CVPR*, 2017. 8