# LPSNet: A lightweight solution for fast panoptic segmentation

Weixiang Hong, Qingpei Guo, Wei Zhang, Jingdong Chen, Wei Chu
Ant Financial Services Group
{hwx229374, qingpei.gqp, ivy.zw, jingdongchen.cjd, weichu.cw}@antgroup.com

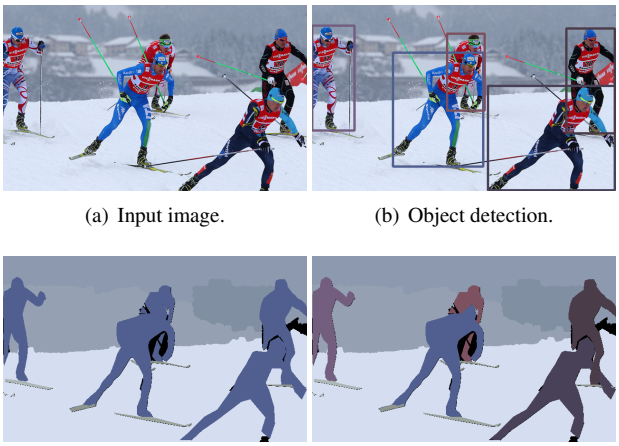
## Abstract

*Panoptic segmentation is a challenging task aiming to simultaneously segment objects (things) at instance level and background contents (stuff) at semantic level. Existing methods mostly utilize a two-stage detection network to attain instance segmentation results, and a fully convolutional network to produce a semantic segmentation prediction. Post-processing or additional modules are required to handle the conflicts between the outputs from these two nets, which makes such methods suffer from low efficiency, heavy memory consumption and complicated implementation. To simplify the pipeline and decrease computation/memory cost, we propose an one-stage approach called Lightweight Panoptic Segmentation Network (LPSNet), which does not involve a proposal, anchor or mask head. Instead, we predict a bounding box and semantic category at each pixel upon the feature map produced by an augmented feature pyramid, and design a parameter-free head to merge the per-pixel bounding box and semantic prediction into panoptic segmentation output. Our LPSNet is not only efficient in computation and memory, but also accurate in panoptic segmentation. Comprehensive experiments on COCO, Cityscapes and Mapillary Vistas datasets demonstrate the promising effectiveness and efficiency of the proposed LPSNet.*

## 1. Introduction

Panoptic Segmentation (PS) is a challenging task aiming to assign each pixel a semantic category and segment each object in the input image [11]. Specifically, the goal of PS is to segment countable objects (things) at instance level and parse amorphous image regions (stuff) at semantic level. Therefore, compared with semantic segmentation or instance segmentation, PS provides more comprehensive scene information and can be broadly used in autonomous driving and scene parsing.

A straightforward solution to tackle PS is to merge the instance segmentation and semantic segmentation predictions, as is done by most existing methods [11, 3, 10, 35, 34]. To achieve good panoptic quality, these methods are often built upon large networks and complex pipelines, without concerning efficiency and computational resources. For example, Mask RCNN [7] has been widely used or integrated to attain promising accuracy, despite that Mask RCNN is heavy in resources consumption and slow in inference. Moreover, the utilization of Mask RCNN naturally introduces conflicts with the output from semantic segmentation branch, hence heuristic or complex post-processing like the pixel rank module [20] are required to deal with the outputs of both branches and obtain unified panoptic segmentation results. Nevertheless, we argue that an efficient solution for PS is not only desired for practical usage like autonomous driving, but also of great importance for potential performance gain by saving memory for larger image and batch size.



(a) Input image. (b) Object detection. (c) Semantic segmentation. (d) Panoptic segmentation.

Figure 1. (Better viewed in color). Lightweight Panoptic Segmentation Network (LPSNet) predicts bounding boxes and semantic segmentation, and merge them into panoptic segmentation output with a parameter-free panoptic head.

In this paper, we propose Lightweight Panoptic Segmentation Network (LPSNet) to tackle the drawback mentioned above. By introducing a parameter-free panoptic head, our LPSNet decomposes panoptic segmentation into two inde-

pendent sub-tasks, *i.e.*, object detection and semantic segmentation, as shown in Figure 1. Thanks to fully convolutional network [22] and the recent one-stage anchor-free detector [13, 12, 31], both sub-tasks can be solved in one pass, in a fully-convolutional style. Therefore, compared with existing two-stage PS methods like UPSNet[34] and AUNet [16], our LPSNet can be around two times faster with less memory consumption. Moreover, by slightly increasing train/test image size, our LPSNet achieves superior accuracy while still maintaining the advantages in memory and computation.

Comprehensive experiments on benchmarks COCO [19], Cityscapes [2] and Mapillary Vistas [24] datasets evidently demonstrate that LPSNet achieves competitive performances with excellent efficiency. We summarize our main contributions as follows:

- We present a novel panoptic segmentation approach LPSNet, which is different from existing methods and produces panoptic segmentation in one pass. Our LPSNet does not involve anchor, proposal or mask head, thus is efficient in computation, memory and hyper-parameters usage. For examples, anchor settings such as scale, aspect ratio are sensitive to different applications and datasets. The parameters of proposal ground truth generation, extraction and selecting strategy requires sophistical tuning. For the mask head, the weight of the loss function is subject to careful trial and error. In contrast, our LPSNet is easy to train and more generic to different scenarios.

- We decompose the PS task as object detection and semantic segmentation with a parameter-free PS head. The head takes detection boxes, object center offset prediction and semantic segmentation as inputs to obtain PS results, while existing approaches usually work on instance segmentation and semantic segmentation results. Our panoptic head is portable to other networks with detection and semantic segmentation branches.

- Additionally, overlapping or deformable objects often cause severe false positive in most one-stage detection methods like FCOS [31]. In our approach, we harness mask information to determine whether a pixel is positive and central or not, thus provide more accurate learning targets and boost the final performances.

## 2. Related Work

**Object Detection** Existing object detectors can be roughly grouped into two categories: two-stage framework which includes a region proposal processing step and one-stage framework which is proposal-free. With the emergence of powerful one-stage detectors [18, 31], the traditional view that two-stage is superior to one-stage on accuracy may be inaccurate. Focal loss [18] solves the problem of imbalance between positive and negative examples and also between hard and easy examples, which significantly promote one-stage detector performance. From the perspective of anchor, object detectors can also be categorized into anchor-based methods [28, 7] or anchor-free ones [13, 31]. Most existing object detectors are based on pre-defined anchors, which are considered to be essential for accuracy. However, there are several drawbacks for anchor-based methods. For example, hyper-parameters of anchor settings need carefully tuned. The computation and storage costs related to anchor boxes are also heavy [31]. Recently anchor-free methods attracts a lot of interest, of which FCOS achieves state-of-the-art performance among one-stage detectors.

**Panoptic Segmentation** Panoptic Segmentation is a joint task to segment both thing and stuff, which is known as scene understanding or image parsing in earlier work. The task was reformulated in [11] as a well defined PS task with panoptic quality (PQ) metric. Typically, PS is solved by semantic segmentation [9] and instance segmentation [7] by two separate networks [11, 3] or one network [10] with different heads to each sub-task. The results of semantic segmentation and instance segmentation are then fused using heuristics or specially designed head.

The baseline of PS proposed in [11] using Mask R-CNN to get instance segmentation and PSPNet to get semantic results. The works [10] propose a single network with Mask R-CNN [7] style instance segmentation head and semantic segmentation head, followed by heuristics to merge two kinds of outputs. There are two kinds of conflicts when conducting merge, *i.e.*, one pixel belonging to multiple instance masks or belonging to a instance mask and stuff segmentation result simultaneously. TASCNet [14] constructs a binary mask predicting things *i.e.*, stuff for each pixel to get consistent results of instance segmentation and stuff segmentation. For overlapping instances generated by MASK R-CNN, the strategy of high confidence and small object first is always used to conduct heuristic fusion. OANet [20] proposes spatial ranking module to deal with the occlusion problem between predicted instances. UPSNet [34] proposes a lightweight and parameter-free panoptic head which predicts the final results via pixel-wise classification to solve the conflicts and facilitate end-to-end training.All these methods are based on two-stage detector integrated with a Mask R-CNN like mask-head to get instance masks, which cost computational and memory resources heavily and are inefficient.

Deeperlab [35] and AdaptIS [29] tackles PS in one stage by generating class-agnostic instance masks based on key points. Specifically, Deeperlab utilizes several key point heatmaps to predict instances. AdaptIS harnesses an additional process to train high quality points. Perhaps [4] and

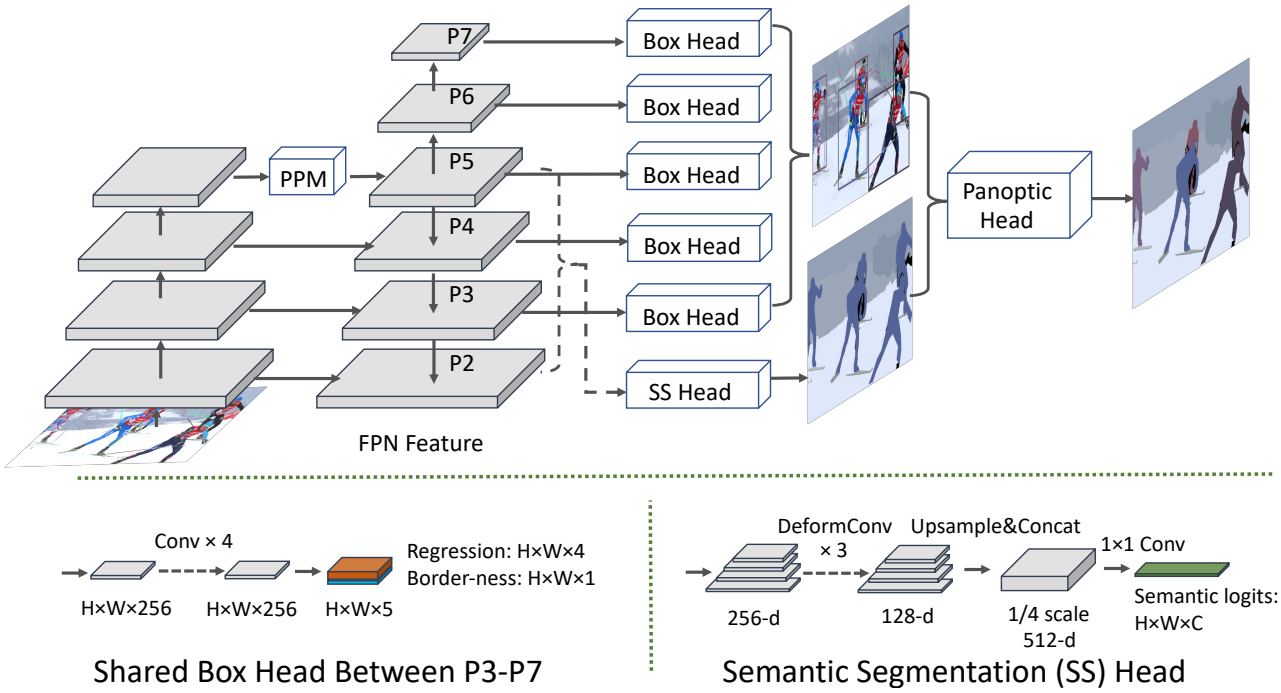



Figure 2. Details of the proposed Single-Stage Panoptic Segmentation approach. The input image is feed to a modified FPN network for computing the feature map. In detail, we increase the empirical receptive field of FPN by adding a Pyramid Pooling Module (PPM), and extend the pyramid of typical FPN by 2 more levels using 2-stride convolution. Our box head takes P3-P7 feature as input, and predict 5 values for each point in feature map, *i.e.*, box and border-ness score. Our Semantic Segmentation (SS) Head take P2-P5 as input and predict per-pixel semantic logit.

[23] are the closest work to ours. [4] uses attention module to assist instance segmentation. [23] uses anchors and centers for obtaining high quality detection boxes. Compared to them, our method is simple in pipeline without involving attention, anchors or centers, while achieving superior performances in public benchmarks.

## 3. Method

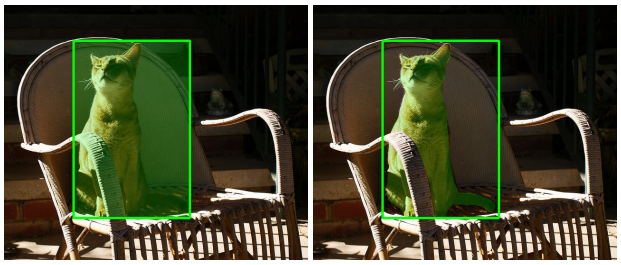
In order to produce panoptic segmentation in one pass, without involving proposal, anchor or mask head, we first introduce a non-parametric panoptic head to decompose panoptic segmentation into semantic segmentation and object detection. Given the fact that semantic segmentation has been tackled in one pass by fully convolutional network (FCN) [22], we propose to formulate object detection in a per-pixel prediction fashion as in semantic segmentation. In detail, our model consists of a shared convolutional feature extraction backbone and three heads on top of it. The three heads are designed for different purposes, but they are either fully-convolutional or non-parametric. The overall model architecture is shown in Figure 2. In this section, we start with explaining the backbone of our lightweight panoptic segmentation network. Then, the detailed architecture of each component is elaborated.

### 3.1. Backbone

We build our model based on a deep residual network (ResNet) [8] with feature pyramid [17]. To better exploit global context information, we use a Pyramid Pooling Module (PPM) over the feature map of the last layer in the 5-th stage of ResNet ('res5'), reduce its dimension to 256 and add back to FPN before producing P5 feature map. PPM is originally proposed in [36] and proven to be highly compatible with FPN in [33]. Similar to [31], we downsample P5 twice to produce P6 and P7 by two convolution layers with stride 2.

### 3.2. Box Head

A shared box head is applied to different feature levels, since shared box heads lead to efficient parameterization and improved performance [31]. As shown in Figure 2, the input and output shape of each box head are $\mathrm{H} \times \mathrm{W} \times 256$ and $\mathrm{H} \times \mathrm{W} \times 5$ respectively, where $\mathrm{H}$ and $\mathrm{W}$ vary with the level of the feature. We utilize a 4-layer sub-network to process the input feature map. A point is considered as a positive sample if it is part of ground-truth mask of things. For each positive sample, we regress a 5-dimensional vector $[l, t, r, b, p]$, where $l, t, r, b$ are the distances from the location to left, top, right and bottom of the bounding box. Note



(a) FCOS [31]. (b) Ours.

Figure 3. Definitions of positive samples. (a). Each point within the bounding box is considered as positive sample in FCOS [31], therefore, the background on the top-right area, as well as the pixels from the chair, will cause false positive and degrade detection performance. (b). Our definition of positive sample utilizes the information from the mask and effectively filters out the false positive pixels.


(a) Center-ness [31]. (b) Border-ness.

Figure 4. **Center-ness *v.s.* border-ness.** Our proposed border-ness assigns more weights to the pixels on object, while center-ness [31] may mistakenly concentrates on the area outside the object. Our border-ness demonstrates improved performance than center-ness in Section 4.4.

that our definition of positive sample is different from [31], where every point within any ground-truth box is treated as a positive sample. As illustrated in Figure 3, our definition can significantly reduce false positive samples, which benefits the detection performances.

The 5-th dimension $p$ to regress is called border-ness, which is introduced to surpress low-quality detected bounding boxes without introducing any hyper-parameters. Let $l^*, t^*, r^*, b^*$ be the distance to the four border of mask as shown in Figure 4, the border-ness target is defined as:

$$\text{border-ness}^* = \sqrt{\frac{\min(l^*, r^*)}{\max(l^*, r^*)} \times \frac{\min(t^*, b^*)}{\max(t^*, b^*)}} \quad (1)$$

The border-ness ranges from 0 to 1 and is thus trained with binary cross entropy (BCE) loss. We employ sqrt to slow down the decay of the border-ness. In inference stage, we multiply the predicted border-ness with the confidence score of the detected bounding boxes so as to depress the bounding boxes far from the center of an object. We expect the following non-maximum suppression (NMS) can filter out these low-quality bounding boxes, and boost the final performances.

### 3.3. Semantic Segmentation Head

The goal of the semantic segmentation head is to parse all semantic classes without discriminating instances. There are plenty of works on designing high-performance segmentation head based on powerful backbone such as ResNet-101 [37], HRNet [30], but rare studies have been done based on FPN. Inspired by the recent work [33, 34], we build our semantic head as a deformable convolution based subnetwork. We feed the 256-channel P2, P3, P4 and P5 feature maps of FPN to our semantic segmentation head. These feature maps are first processed by our deformable convolution network independently, then upsampled to 1/4 scale of the input image by bilinear interpolation. Finally, we concatenate them and apply $1\times1$ convolutions with softmax to predict the semantic class. The architecture is shown in bottom right of Figure 2. We use the pixel-wise cross entropy loss to supervise our semantic segmentation head.

### 3.4. Panoptic Head

The goal of the panoptic head is to produce panoptic segmentation based on bounding boxes and per-pixel semantic segmentation. We design it as a non-parametric module so it is portable to other networks, we note that there are also panoptic head with learned parameters [20]. In details, we iteratively "paste" each box to the semantic segmentation prediction by confidence from high to low, so that the highly confident box may "occupy" the pixels and hide false positive area from low-confident predictions. For a given bounding box of class $C$, the pixels within it are either of class $C$ or not, and we consider those pixels of class $C$ as the mask of this box. If the intersection between the current mask and those already existing is larger than a threshold (0.3 in our experiments), we discard this mask. Otherwise we keep the non-intersecting part.

A special case is that two (or more) boxes of the same class are overlapping, and some pixels in the overlapping area also have the same class. In this case, we compute the border-ness of those pixels to all competing boxes, and assign the pixel to the box whose computed border-ness is the closest to the predicted border-ness. The accurate assignment of these pixels is beneficial to better visualization, but we found in experimentation that it does not matter a lot to performances.

Table 1. **Panoptic segmentation results on COCO val.** The number in bracket stands for the length of shorter size of our train/test images. Other methods are trained/tested with images with shorter size length as 800. Superscripts Th and St stand for thing and stuff. '-' means inapplicable. We note that the running time includes the post-processing.

| Models | PQ | SQ | RQ | $PQ^{Th}$ | $PQ^{St}$ | box mAP | mask mAP | mIoU | time | memory | FLOPS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| JSIS-Net [3] | 26.9 | 72.4 | 35.7 | 29.3 | 23.3 | - | - | - | - | - | - |
| Panoptic FPN [10] | 33.3 | - | - | 45.9 | 28.7 | - | - | 41.0 | - | - | - |
| OANet [20] | 39.0 | 77.1 | - | 43.5 | 24.9 | - | - | - | - | - | - |
| SSAP [6] | 36.5 | **80.7** | 44.8 | 40.1 | 32.0 | - | - | - | - | - | - |
| Axial-DeepLab-L [32] | **43.9** | - | - | **48.6** | **36.8** | - | - | - | - | - | - |
| UPSNet [34] | 42.3 | 78.0 | 52.4 | 48.5 | 33.4 | 37.8 | **34.3** | 54.3 | 202 | 10.4G | 259G |
| LPSNet (800) | 39.1 | 75.2 | 51.2 | 43.9 | 30.1 | 39.2 | 26.8 | 54.5 | 108 | 4.4G | 139G |
| LPSNet (882) | 41.9 | 77.2 | 51.5 | 46.3 | 32.5 | **39.4** | 29.6 | 56.1 | 127 | 4.9G | 153G |
| LPSNet (964) | **42.4** | 79.2 | **52.7** | **48.0** | 35.8 | **39.5** | 31.5 | **59.8** | 146 | 5.5G | 167G |

### 3.5. Implementation Details

We implement our model using PyTorch [26].

**Training:** We train our model using 8 GPUs, with 2 images per GPU for each mini-batch. We use ground-truth per-pixel labels to construct the targets of bounding boxes and border-ness. Our LPSNet contains 3 sub-tasks in total: bounding box localization, border-ness regression and semantic segmentation. Different weighting schemes on these multi-task loss functions could lead to very different training results. We empirically found the loss balance strategy, *i.e.*, assuring the scales of all losses are roughly on the same order of magnitude, works well in practice.

**Inference:** In inference stage, we firstly feed the input image through the network to obtain the predicted bounding boxes and per-pixel semantic prediction. The class of each predicted bounding box is inferred with the output of semantic segmentation head following FCOS [31] . Unless specified, our post-processing procedure on bounding boxes is also the same with [31]. We hypothesize that the performance of our box head may be improved further if we carefully tune the hyper-parameters. The predicted bounding boxes and semantic segmentation are fed to panoptic head for obtaining the final panoptic segmentation outputs.

## 4. Experiments

In this section, we present the experimental results on COCO [19], Cityscapes [2] and Mapillary Vistas [25] datasets.

**COCO** [19] is the most suitable and challenging one for the new panoptic segmentation task, for the detailed annotations and high data complexity. It consists of 115k images for training and 5k images for validation, as well as 20k images for test-dev and 20k images for test-challenge. MS-COCO panoptic annotations includes 80 thing categories and 53 stuff categories. We train our models on train set with no extra data and reports results on val set and test-dev set for comparison.

**Cityscapes** [2] dataset contains 2975 images for training, 500 images for validation and 1525 images for testing with fine annotations. It has another 20k coarse annotations for training, which are not used in our experiment. We report our results on val set with 19 semantic label and 8 annotated instance categories.

**Mapillary Vistas** [24] is adopted to further illustrate the effectiveness of the proposed method. In details, Mapillary Vistas is one of the richest, publicly available street-level image datasets today, with 18k/2k/5k images for training, validation and test, respectively. Following the setting of [29], We scale the images by largest side varied from 500 to 2200 pixels and then perform random crop of size $400 \times 800$.

**Experimental Setup** For all three datasets, we report results on the validation set. To evaluate the performance, we adopt panoptic quality (PQ), recognition quality (RQ) and semantic quality (SQ) [11] as the metrics. For the auxiliary tasks of panoptic segmentation, *i.e.*, object detection, instance segmentation and semantic segmentation, we measure their performances by bounding box mAP, mask mAP and mean IoU. We also report inference FLOPS, inference time and memory consumption. At last, we conduct ablation studies to investigate the effectiveness of different components of proposed method.

We train our network with stochastic gradient descent (SGD). For COCO, we set the initial learning rate as $0.01$ and batch size as $16$. The total number of iterations is $90K$, and the learning rate is reduced by a factor of $10$ at iteration $60K$ and $80K$, respectively. Weight decay and momentum are set as $0.0001$ and $0.9$, respectively. We initialize our backbone networks with the weights pre-trained on ImageNet [5]. For the newly added layers, we initialize them as in [18]. For Cityscapes, we train our method for $12K$ iterations in total, with the learning rate decayed at $9K$ iterations. Loss weights of semantic head are $0.7$ and $1.0$ on COCO and Cityscapes respectively. For Mapillary Vistas, we train for a total of 192k iterations, decreasing the learn-

Table 2. **Panoptic segmentation results on COCO test-dev.** The top 3 rows contain results of top 3 models taken from the official leadboard.

| Models | backbone | PQ | SQ | RQ | $PQ^{Th}$ | $SQ^{Th}$ | $RQ^{Th}$ | $PQ^{St}$ | $SQ^{St}$ | $RQ^{St}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| Megvii | ensemble model | 54.7 | 83.6 | 64.3 | 64.6 | 86.2 | 74.6 | 39.8 | 79.7 | 48.8 |
| Innovation | ensemble model | 53.5 | 83.3 | 63.3 | 61.8 | 84.9 | 72.4 | 41.0 | 80.9 | 49.6 |
| Megvii (Face++) | ensemble model | 53.2 | 83.2 | 62.9 | 62.2 | 85.5 | 72.5 | 39.5 | 79.7 | 48.5 |
| JSIS-Net [3] | ResNet-50 | 27.2 | 71.9 | 35.9 | 29.6 | 71.6 | 39.4 | 23.4 | 72.3 | 30.6 |
| AUNet [16] | ResNeXt-152 | **46.5** | **81.0** | 56.1 | **55.9** | **83.7** | **66.3** | 32.5 | 77.0 | 40.7 |
| UPSNet [34] | ResNet-101-DCN | **46.6** | 80.5 | **56.9** | 53.2 | 81.5 | 64.6 | 36.7 | 78.9 | 45.3 |
| LPSNet (736) | ResNet-101 | 44.2 | 78.3 | 54.2 | 47.7 | 76.2 | 57.8 | 35.4 | 77.3 | 41.8 |
| LPSNet (818) | ResNet-101 | 45.6 | 79.4 | 55.1 | 49.3 | 77.9 | 58.9 | 36.9 | 78.1 | 45.3 |
| LPSNet (900) | ResNet-101 | **46.3** | 80.2 | **57.0** | 50.5 | 78.7 | 62.8 | **38.6** | **79.2** | **46.9** |

Table 3. **Panoptic segmentation results on Cityscapes val.** Superscripts Th and St stand for thing and stuff. '-' means inapplicable.

| Models | PQ | SQ | RQ | $PQ^{Th}$ | $PQ^{St}$ | box mAP | mask mAP | mIoU |
|---|---|---|---|---|---|---|---|---|
| Li *et al.* [15] | 53.8 | - | - | 42.5 | 62.1 | - | 28.6 | 71.6 |
| Panoptic FPN [10] | 58.0 | 79.2 | 71.8 | 52.3 | 62.2 | - | 32.8 | 75.2 |
| TASCNet [14] | 55.9 | - | - | 50.5 | 59.8 | - | - | - |
| SSAP [6] | 58.4 | - | - | 50.6 | - | - | **34.4** | - |
| UPSNet [34] | 59.3 | 79.7 | 73.0 | **54.6** | 62.7 | 36.8 | 33.3 | 75.2 |
| LPSNet (1024) | **59.7** | **79.9** | **73.6** | 54.0 | 63.9 | **38.4** | 32.8 | 78.1 |
| LPSNet (1200) | **60.4** | **80.3** | **74.0** | 54.2 | **64.5** | **38.5** | 33.0 | **78.6** |
| Multi-scale | PQ | SQ | RQ | $PQ^{Th}$ | $PQ^{St}$ | box mAP | mask mAP | mIoU |
| Panoptic FPN [10] | **61.2** | 80.9 | 74.4 | 54.0 | **66.4** | - | **36.4** | 80.9 |
| UPSNet [34] | 60.1 | 80.3 | 73.5 | **55.0** | 63.7 | 37.1 | 33.3 | 76.8 |
| LPSNet (1024) | 60.5 | 80.7 | 73.5 | 53.7 | 64.8 | **38.8** | 33.2 | 80.8 |
| LPSNet (1200) | **61.3** | **81.2** | **74.8** | 54.5 | 65.6 | **38.9** | 33.6 | **81.4** |

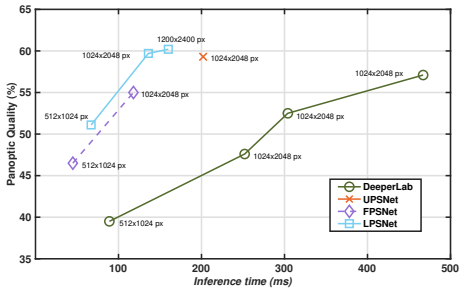



Figure 5. **Panoptic quality *v.s.* inference time.** The results are obtained on Cityscapes dataset with various image sizes. The three variants of DeeperLab at $1024 \times 2048$ are from three different backbones [35]. Compared with FPSNet and UPSNet, our LPSNet achieves promising balance between efficiency and accuracy.

ing rate after 144k and 176k iterations.

## 4.1. COCO

We compare our method with JSIS-Net [3], Panoptic FPN [10], OANet [20], AUNet [16], SSAP [6], Axial-DeepLab-L [32] and UPSNet [34]. For fairness of comparing with other methods, we conduct experiments on the COCO dataset with ResNet-50 as backbone, except that Axial-DeepLab-L [32] developed an improved backbone Axial-ResNet-50 based on ResNet-50. The results under different metrics are shown in Table 1. The mIoU metric is computed over the 133 classes of stuff and thing in the COCO 2018 panoptic segmentation task, which is different from previous 172 classes of COCO-Stuff. Compared with the main competitor UPSNet [34], the PQ of our LPSNet is lower than UPSNet by 3 with the same image size (shorter side as 800), while our method is around 2 times faster than UPSNet with less memory consumption. By slightly increasing train/test image size, our method surpasses UPSNet in PQ while still maintains the advantages in speed and memory. Specifically, with train/test image size as 964, our LPSNet outperforms UPSNet [34] in all metrics except the $PQ^{Th}$ and mask mAP. Note that UPSNet [34] is expected to achieve better PQ by increasing train/test image size, however, when it comes to a larger backbone like ResNet-101-DCN in Table 2, a single image of 768 size with UPSNet will cost around $16G$ memory, which is close to the largest capacity of most existing GPUs like TESLA V100. In contrast, our method is still scalable with ResNet-101 as backbone, as shown next.

We add the comparisons on the test-dev of MS-COCO 2018 in Table 2. Although we just use ResNet-101 as the backbone, we achieve comparable results compared to the recent AUNet [16] that uses ResNeXt-152. We also list the top three results on the leaderboard which uses ensemble and other tricks. It is clear from the table that we are on par with the third best model without bells and wistles.

We show visual examples of panoptic segmentation on this dataset in Figure 6. From the 1-st row of the figure, we observe that our LPSNet achieves better semantic segmentation performance than UPSNet [34], probably due to the utilization of PPM [33]. In the 2-nd row, a bag is missed by UPSNet but detected by our method. Also, our LPSNet excels in handling occlusion in row 3.

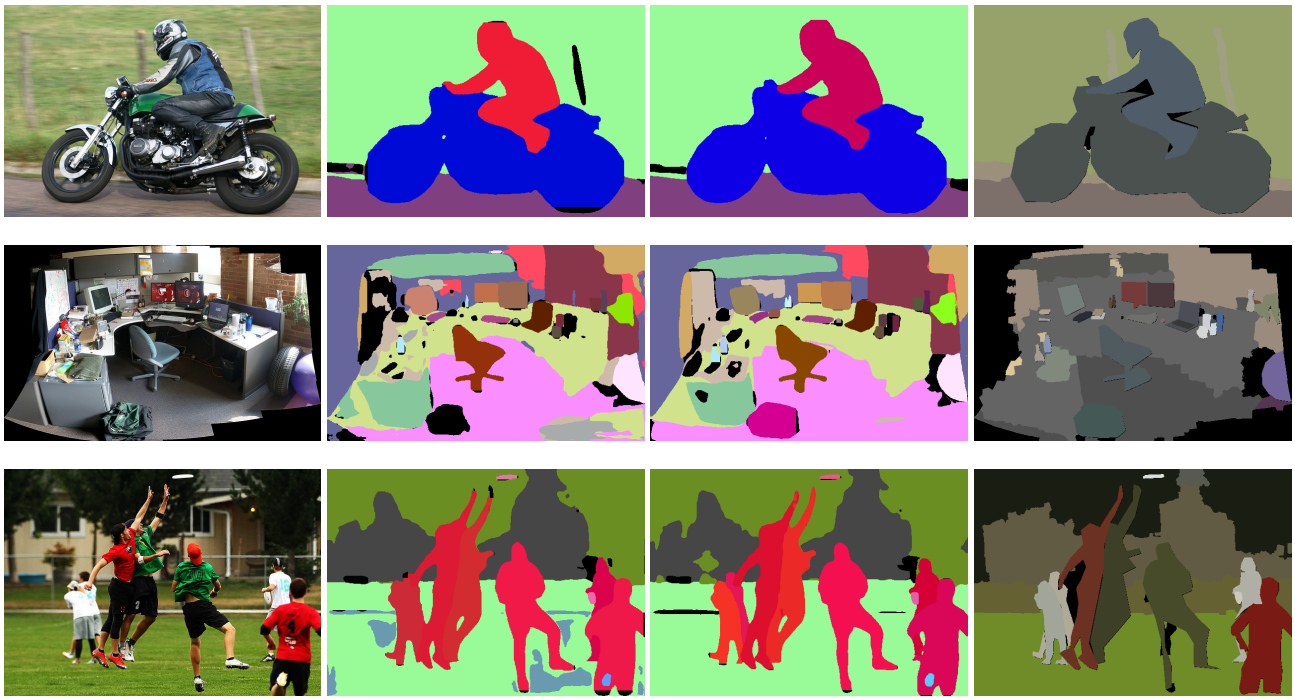


Figure 6. **Visual examples of panoptic segmentation on COCO dataset.** From left to right are input image, UPSNet output, LPSNet output, ground-truth.

### 4.2. Cityscapes

We compare our method with Li *et al.* [15], Panoptic FPN [10], TASCNet [14], FPSNet [4], DeeperLab [35], SSAP [6] and UPSNet [34]. Note that the method in [15] uses a ResNet-101 as the backbone, whereas all other reported methods use ResNet-50. Most existing works reported the panoptic quality on CityScapes val set, thus it is convenient and fair to compare on val set.

The results are reported in Table 3. Even with the same train/test image size as 1024, our method achieves competitive PQ at 59.7. This may possibly be caused by the fact that the stuff area in Cityscapes are usually larger than that of COCO. Although multi-scale testing significantly improves both Panoptic FPN [11] and our LPSNet, ours is still slightly better with train/test image size as 1200. As illustrated in Figure 5, our method achieves promising balance between panoptic quality and inference time. In case of train/test image size as $1024 \times 2048$ and $1200 \times 2400$, our method achieves the best PQ among all compared method. Compared with FPSNet [4], our method is higher in PQ by around 10, at the cost of around 20ms slower in inference.

### 4.3. Mapillary Vistas

We compare our method with TASCNet [14], SeamlessSeg [27], AdaptIS [29] and Axial-DeepLab-L [32] on Mapillary Vistas dataset. With ResNet-50 as backbone, all methods are compared in Table 4. We obtain +0.3% and +4.5% PQ score over SeamlessSeg [27] and AdaptIS [29] , respectively. The consistent advantages on state-of-the-art methods validate the merits of our LPSNet.

Table 4. **Panoptic segmentation results on Mapillary Vistas val.** Superscripts Th and St stand for thing and stuff. '-' means inapplicable.

| Methods | Backbone | PQ | $PQ^{Th}$ | $PQ^{St}$ | mIoU |
|---|---|---|---|---|---|
| TASCNet [14] | ResNet-50 | 32.6 | 31.1 | 34.4 | - |
| AdaptIS [29] | | 32.0 | 26.6 | 39.1 | - |
| DeeperLab [35] | | 32.0 | - | - | **55.3** |
| SeamlessSeg [27] | | 36.2 | **33.6** | 40.0 | 45.8 |
| Axial-DeepLab-L [32] | Axial-ResNet-50 | 41.1 | 33.4 | 51.3 | 58.4 |
| LPSNet | ResNet-50 | **36.5** | 33.2 | **41.0** | 48.8 |

### 4.4. Ablation Study

We investigate the effectiveness of different components of the proposed LPSNet. The performance achieved by different variants and settings are reported in the following.

**The effectiveness of PPM** To verify the importance of Pyramid Pooling Module (PPM) in enhancing the feature maps of FPN, we compare it with a simple baseline "Global Average Pooling" proposed in [34]. We also try to train our network without the PPM. We note that this degrades our backbone similar to Panoptic FPN [10].

Since the effect of PPM to box mAP is experimentally observed to be minor, here we only report mIoU and PQ of different variants in Table 5 left. The PPM outperforms "Global Average Pooling" by more than 3% in mean IoU. When the PPM is removed, the IoU significantly drops

Table 5. **Ablation studies on PPM and conflicted pixels assignment.** The experiments are conducted on COCO val set with ResNet-50 as backbone.

| Models | mIoU | PQ | Strategies | PQ | $PQ^{Th}$ | $PQ^{St}$ |
|---|---|---|---|---|---|---|
| Global Average Pooling | 51.2 | 36.5 | Smallest Area | 38.9 | 43.5 | 30.0 |
| Without PPM | 50.9 | 35.7 | Highest Confidence | 39.0 | 43.6 | 30.0 |
| With PPM | 54.5 | 39.1 | Closest Border-ness | 39.1 | 43.9 | 30.1 |

around 4%. These observations validate that PPM can effectively enhance the feature map of FPN to boost semantic segmentation.

**Closest Border-ness Assignment** In our proposed network, the closest border-ness rule is mainly designed to improve id assignment of intra-class overlaps, for the purpose of better visualization. Here we give quantitative evaluations of how the closest border-ness assignment affects the final PQ on COCO val set. As shown in Table 5 right, we adopt three different strategies for handling the intra-class conflicts. In detail, the conflicted area will be assigned to the box with smallest area or highest confidence for "Smallest Area" and "Highest Confidence", respectively. Though "Smallest Area" and "Highest Confidence" show minor PQ differences with "Closest Border-ness" [23], they may produce straight contours with rigid corners originating from the bounding boxes. To alleviate these bad visualizations, "Closest Border-ness" allows sophisticated assignment of each pixel in the conflicted area, hence enables smooth contours in visualizations as shown in Figure 6.

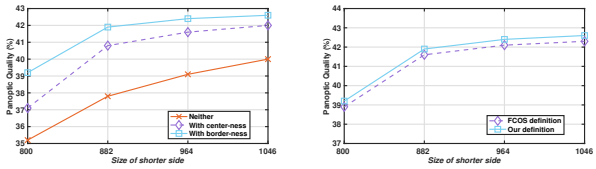
**Mask head matters** From Table 1, 2 and 3, we observe an interesting phenomenon that the box mAP of our method is usually higher than that of Panoptic FPN and UPSNet, while the mask mAP is relatively lower. Given that our method excels in semantic segmentation, we conjecture that the mask head significantly boosts the mask mAP for those Mask RCNN based method. To verify our conjecture, we replace the semantic segmentation with groundtruth labels, which will eliminate the contribution of mask head to the final panoptic segmentation output.

The experimental results are shown in Table 6. It can be seen that the groundtruth semantic segmentation can significantly boost the performance of UPSNet and our method, which is consistent with the observations in [34]. Also, we note that UPSNet achieves higher PQ than our method without groundtruth semantic segmentation, while ours surpasses it if groundtruth semantic segmentation is utilized, due to our higher detection mAP. This demonstrates that the mask head plays a key role for achieving promising PQ. The ablations of using ground-truth bounding boxes are also presented, the gains of our method is less than that of UPSNet, which is as expected.

**Why increasing image size helps?** In light of the good performances achieved by increasing train/test image sizes, a natural question to ask is: what is the rationale behind increasing image size. As presented in Table 2 of [1], large input size is usually beneficial for semantic segmentation. Since our LPSNet generates the instance mask based on semantic segmentation and bounding box prediction, the improved segmentation accuracy can lead to better mask mAP. Some researchers even compromise model size to increase image resolution [21].

Table 6. **With/without groundtruth semantic segmentation (SS) and bound boxes.** The experiments are conducted on COCO val set with ResNet-50 as backbone.

| | box mAP | segmentation IoU | mask mAP | PQ |
|---|---|---|---|---|
| UPSNet | 37.8 | 54.3 | 34.3 | 42.5 |
| UPSNet + SS | 37.8 | - | 46.1 | 72.0 |
| UPSNet + Box | - | 54.5 | 51.0 | 60.8 |
| Ours | 39.2 | 54.5 | 26.8 | 39.1 |
| Ours + SS | 39.2 | - | 47.0 | 74.1 |
| Ours + Box | - | 54.5 | 51.8 | 53.0 |




(a) Effect of border-ness. (b) Different positive sample def.

Figure 7. The experiments are conducted on COCO datset with ResNet-50 backbone. (a). Our border-ness demonstrates advantages to center-ness on various train/test image size. (b). Our definition of positive samples leads to improved performances than that of FCOS [31].

**How much does border-ness and mask-based positive sample contribute?** We conduct controlled experiments to verify the effectiveness of our proposed border-ness (Figure 4) and mask-based positive sample (Figure 3). As shown in Figure 7, the border-ness and mask-based positive sample can boost the PQ of the proposed methods. In particular, the gain of border-ness can be around 2 PQ with image shorter size as 800.

## 5. Conclusion

To efficiently and effectively tackle panoptic segmentation, we have proposed one-stage, anchor-free and proposal-free network called LPSNet. By introducing a non-parametric panoptic head, we decompose panoptic segmentation into two sub-tasks object detection and semantic segmentation, both of which can be solved in fully convolutional per-pixel prediction style. As shown in experiments, LPSNet compares favourably against the popular two-stage approaches like UPSNet and Panoptic FPN in terms of both accuracy and efficiency. Given its effectiveness and efficiency, we hope that LPSNet can serve as a strong and simple alternative of current mainstream anchor-based approaches. In the future, we plan to further improve LPSNet by borrowing merits from the mask head, as well as accelerating it towards real-time inference.

## References


[1] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *CVPR*, 2018.

[2] 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.

[3] Daan de Geus, Panagiotis Meletis, and Gijs Dubbelman. Panoptic segmentation with a joint semantic and instance segmentation network. *arXiv preprint*, 2018.

[4] Daan de Geus, Panagiotis Meletis, and Gijs Dubbelman. Fast panoptic segmentation network. *arXiv preprint arXiv:1910.03892*, 2019.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[6] 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.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

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

[9] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *CVPR*, 2018.

[10] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollar. Panoptic feature pyramid networks. In *CVPR*, 2019.

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

[12] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, Lei Li, and Jianbo Shi. Foveabox: Beyound anchor-based object detection. *IEEE Transactions on Image Processing*, 2020.

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

[14] Jie Li, Allan Raventos, Arjun Bhargava, Takaaki Tagawa, and Adrien Gaidon. Learning to fuse things and stuff. *arXiv preprint arXiv:1812.01192*, 2018.

[15] Qizhu Li, Anurag Arnab, and Philip H.S. Torr. Weakly- and semi-supervised panoptic segmentation. In *ECCV*, 2018.

[16] 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.

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

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[19] 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.

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

[21] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. `http://presentations.cocodataset.org/COCO17-Detect-UCenter.pptx`. No. 1 of COCO 2017 Instance Segmentation Track.

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[23] Bastian Leibe Mark Weber, Jonathon Luiten. Single-shot panoptic segmentation. *arXiv preprint arXiv:1911.00764*, 2019.

[24] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017.

[25] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017.

[26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS Workshop*, 2017.

[27] Lorenzo Porzi, Samuel Rota Bulo, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In *CVPR*, 2019.

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

[29] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. Adaptis: Adaptive instance selection network. In *ICCV*, 2019.

[30] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[31] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019.

[32] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In *ECCV*, 2020.

[33] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018.

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

[35] 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 preprint arXiv:1902.05093*, 2019.

[36] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.

[37] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.