Real-Time Panoptic Segmentation from Dense Detections

Rui Hou*,1,2 Jie Li*,1 Arjun Bhargava1 Allan Raventos1 Vitor Guizilini1 Chao Fang1
Jerome Lynch2 Adrien Gaidon1
1Toyota Research Institute 2University of Michigan, Ann Arbor
1{firstname.lastname}@tri.global 2{rayhou, jerlynch}@umich.edu

Abstract

Panoptic segmentation is a complex full scene parsing task requiring simultaneous instance and semantic segmentation at high resolution. Current state-of-the-art approaches cannot run in real-time, and simplifying these architectures to improve efficiency severely degrades their accuracy. In this paper, we propose a new single-shot panoptic segmentation network that leverages dense detections and a global self-attention mechanism to operate in real-time with performance approaching the state of the art. We introduce a novel parameter-free mask construction method that substantially reduces computational complexity by efficiently reusing information from the object detection and semantic segmentation sub-tasks. The resulting network has a simple data flow that requires no feature map re-sampling, enabling significant hardware acceleration. Our experiments on the Cityscapes and COCO benchmarks show that our network works at 30 FPS on 1024 × 2048 resolution, trading a 3% relative performance degradation from the current state of the art for up to 440% faster inference.

1. Introduction

Scene understanding is the basis of many real-life applications, including autonomous driving, robotics, and image editing. Panoptic segmentation, proposed by Kirillov et al. [14], aims to provide a complete 2D description of a scene. This task requires each pixel in an input image to be assigned to a semantic class (as in semantic segmentation) and each object instance to be identified and segmented (as in instance segmentation). Facilitated by the availability of several open-source datasets (e.g. Cityscapes [5], COCO [20], Mapillary Vistas [23]), this topic has drawn a lot of attention since it was first introduced [15, 13, 36, 28].

In panoptic segmentation, pixels are categorized in two high level classes: stuff representing amorphous and uncountable regions (such as sky and road), and things covering countable objects (such as persons and cars). These two categories naturally split the panoptic segmentation task into two sub-tasks, namely semantic segmentation and instance segmentation. Most recent approaches use a single backbone for feature extraction and add various branches on top of the shared representations to perform each downstream task separately, generating the final panoptic prediction with fusion heuristics [13, 36, 28].

To date, most studies on panoptic segmentation focus on improving model accuracy, either by integrating more advanced semantic and instance segmentation methods [28, 16] or by introducing novel information flow and loss functions [15, 30]. None of these methods are suitable for real-time applications due to prohibitively slow inference speeds. A small subset of recent works is making progress towards faster panoptic segmentation algorithms [7, 38] but at a significant cost in terms of accuracy.

To achieve high quality panoptic segmentation under
real-time constraints, we identify two key opportunities for streamlining existing frameworks. Our first observation is that most accurate instance segmentation methods follow a “detect-then-segment” philosophy, but a significant amount of information is discarded during the “detect” phase. Specifically, “dense” object detection algorithms such as YOLO [29], RetinaNet [19] and FCOS [31] first generate a super-set of bounding box proposals (at least one per location), wherein multiple proposals may correspond to a single target object. Then, Non-Maximum Suppression (NMS) or an equivalent filtering process picks out predictions with the highest confidence and ignores the rest. This selection strategy discards lower ranking proposals generated by the network, even though they might have significant overlap with the ground truth. We instead propose to reuse dense bounding box proposals discarded by NMS to recover instance masks directly, i.e. without re-sampling features [36] or clustering post-processing [38].

Second, we observe that semantic segmentation captures much of the same information as detection, especially in existing panoptic segmentation frameworks. For example, in [36][13][15][28], class predictions for object detections are a subset of those for semantic segmentation, and are produced from identical representations. Hence, sharing computations across semantic segmentation and detection streams can significantly reduce the overall complexity.

Given these insights, we explore how to maximally reuse information in a single-shot, fully-convolutional panoptic segmentation framework that achieves real-time inference speeds while obtaining performance comparable with the state of the art. Our main contributions are threefold: (i) we introduce a novel panoptic segmentation method extending dense object detection and semantic segmentation by reusing discarded object detection outputs via parameter-free global self-attention; (ii) we propose a single-shot framework for real-time panoptic segmentation that achieves comparable performance with the current state of the art as depicted in Figure[1] but with up to 4x faster inference; (iii) we provide a natural extension to our proposed method that works in a weakly supervised scenario.

2. Related Work

2.1. Instance Segmentation

Instance segmentation requires distinct object instances in images to be localized and segmented. Recent works can be categorized into two types: two-stage and single-stage methods. Represented by Mask R-CNN [10] and its variations [22][4][12], two-stage algorithms currently claim the state of the art in accuracy. The first stage proposes a set of regions of interest (RoIs) and the second predicts instance masks from features extracted using RoIAlign [10]. This feature re-pooling and re-sampling operation results in large computational costs that significantly decrease efficiency, rendering two-stage models challenging to deploy in real-time systems. Single-stage methods, on the other hand, predict instance location and shape simultaneously. Some single-stage methods follow the detect-then-segment approach, with additional convolutional heads attached to single-stage object detectors to predict mask shapes [37][32][3][5]. Others learn representations for each foreground pixel and perform pixel clustering to assemble instance masks during post-processing [24][8][6][25][17]. The final representation can be either explicit instance-aware features [24][17], implicitly learned embeddings [6][25], or affinity maps with surrounding locations at each pixel [8].

Following the detect-then-segment philosophy, our work tackles instance segmentation solely based on object detection predictions. In this sense, it is similar to works which densely predict location-related information to separate different instances [24][8][6]. However, even though these methods claim real-time performance, their location proposal and pixel clustering processes are formulated in an iterative pixel-based manner, which makes inference time dramatically increase with the number of instances. Our framework regresses bounding boxes in such a way that proposals can be selected directly through NMS.

Our framework is also similar to the Proposal-Free Network (PFN) [17], as we use instance bounding box coordinates as features to group same-instance pixels. However, PFN relies on predicting the number of instances and on a clustering algorithm which is sensitive to parameter tuning. We use bounding boxes as both proposals and embeddings, and apply a straightforward correlation matrix-based approach for mask recovery. Casting instance segmentation as an object detection problem dramatically decreases the number of model parameters and hyper-parameters involved in training.

2.2. Panoptic Segmentation

Originally proposed in [14], panoptic segmentation has been widely accepted by the computer vision community as a major task for dense visual scene understanding. Each pixel in an image needs to be assigned a semantic label as well as a unique identifier (ID) if it is an object instance. Pixels with the same label belong to the same category, and pixels with the same ID (if present) furthermore belong to the same object instance.

Current state-of-the-art panoptic segmentation work uses a multi-decoder network to perform prediction with redundant information, as well as more sophisticated instance or semantic segmentation heads for better performance [36][13][15][21][28][16].

While obtaining state-of-the-art accuracy on panoptic segmentation, these methods are far from real-time applications mainly due to two reasons: 1) the instance segmenta-
Panoptic segmentation from dense detections. We obtain a panoptic segmentation $P$ from a semantic segmentation $S$ and dense bounding box predictions $B$. We first select the highest-confidence “query” bounding boxes through NMS; then we estimate a location based mask probability $\hat{P}_{loc}$ through self-attention of the query boxes and the dense box predictions. Finally, we refine instance masks with the semantic probability map $\hat{P}_{sem}$ and produce our panoptic output.

3. Panoptic from Dense Detections

3.1. Problem Formulation

In this section, we describe our approach to address the panoptic segmentation task ($P$) by solving a semantic segmentation task ($S$) and a dense bounding box detection task ($B$) as depicted in Figure 2. The objective of panoptic segmentation is to predict semantic and instance IDs for each pixel $(x, y)$ in the input image:

$$P(x, y) = (c, k), \quad c \in \{1, ..., N\}, k \in \mathbb{N},$$

where $c$ is the semantic class ID, $k$ is the instance ID (with 0 for all stuff classes) and $N$ is the total number of classes, including stuff ($N_{\text{stuff}}$) and things ($N_{\text{things}}$) classes. In the semantic segmentation sub-task, we predict a distribution over semantic classes for each pixel $(x, y)$, $S(x, y)$. We denote $P_{sem}(x, y, c)$ as the predicted probability at pixel $(x, y)$ for semantic class $c$, given by:

$$\hat{P}_{sem}(x, y, c) = S(x, y) \cdot c, \quad S(x, y) \in \mathbb{R}^N$$

In the dense bounding box detection sub-task, we predict at least one bounding box at each image pixel:

$$B(x, y) = B, \quad B = (b, c),$$

where $(x_1, y_1)$ and $(x_2, y_2)$ are the coordinates of the top-left and bottom-right corners of bounding box $B$ that pixel $(x, y)$ belongs to; $c$ is the predicted class ID for the corresponding bounding box. We note that because $S$ and $B$ are of fixed dimensions, they can be directly learned and predicted by fully convolutional networks.
3.2. Parameter-Free Mask Construction

Given $S$ and $B$, we introduce a parameter-free mask reconstruction algorithm to produce instance masks based on a global self-attention mechanism. Note that the following operations are embarrassingly parallel. We first obtain a reduced set of bounding box proposals from $B$ through NMS:

$$E_{\text{query}} = \{B_j\}, \quad B_j = (b_j, c_j). \quad (4)$$

We denote the proposal set as $E_{\text{query}}$ because we will use them to “search” for instance masks. For each query box $B_j$, we construct a global mask probability map given by:

$$M(x, y, j) = \hat{P}_{\text{loc}}(x, y, j) \cdot \hat{P}_{\text{sem}}(x, y, c_j), \quad (5)$$

where $\hat{P}_{\text{loc}}(x, y, j)$ is an estimated probability that pixel $(x, y)$ is inside object $j$’s bounding box. We estimate this probability by self-attention between the global set of boxes $B$ and the query box $B_j$ with Intersection over Union (IoU):

$$\hat{P}_{\text{loc}}(x, y, j) = \text{IoU}(B(x, y), B_j), \quad (6)$$

where IoU($B_i, B_j$) = intersection($b_i, b_j$)/union($b_i, b_j$), and $\hat{P}_{\text{sem}}(x, y, c_j)$ is the predicted probability that pixel $(x, y)$ shares the same semantic class $c_j$ given by Eq. (5).

To construct the final instance masks $\{M_j\}$, we apply a simple threshold $\sigma$ to the global mask probability map:

$$M_j(x, y) = M(x, y, j) > \sigma. \quad (7)$$

To produce a final panoptic segmentation, we follow the conventional fusion strategy in [14]. A graphical illustration of the complete method is shown in Figure 2. We demonstrate the efficacy of our proposed method in a novel real-time single-stage architecture in the following section. We note, however, that this method can be generalized to any architecture that provides predictions of $B$ and $S$.

4. Real-Time Panoptic Segmentation Network

We propose a single-stage panoptic segmentation network capable of real-time inference, as shown in Figure 3. As in other recent works on panoptic segmentation [25][13], our architecture is built on a ResNet-style [11] encoder with a Feature Pyramid Network (FPN) [18]. Our FPN module consists of 5 levels corresponding to strides of 128, 64, 32, 16 and 8 with respect to the input image.

Our panoptic segmentation architecture is inspired by FCOS [31], a fast and accurate fully convolutional, anchor-less object detector. However, we use a finer-grained target assignment scheme and additional single-convolution layers for semantic segmentation and a novel “levelness” prediction, which will be described later in Section 4.1. Our framework also leverages a fully tensorizable mask construction algorithm described in Section 4 and we propose an explicit instance mask loss which further improves the quality of the final panoptic segmentation.

4.1. Target Assignment

We formulate object detection as a per-pixel prediction problem. Specifically, let $F_i \in \mathbb{R}^{h_i \times w_i \times C}$ be the feature map at layer $i$ of the FPN, with stride $z$. We assign a target to each location $(x, y)$ on $F_i$ for all $i$. Since each $(x, y)$ is the center of a receptive field, we can recover the pixel location in the original image using the relation: $(x_o, y_o) = ([\frac{x}{z}] + x, [\frac{y}{z}] + y)$. If a pixel location $(x_o, y_o)$ falls within one of the ground truth instance masks $M^i_j$ : $\{M_j^i\}$, then we consider it a foreground sample. Since instance masks are non-overlapping, location $(x_o, y_o)$ is associated with only one mask $M_j^i$ and its corresponding bounding box $B_j^i$. This avoids the “ambiguous pixel” issue described in [31]. If location $(x_o, y_o)$ is associated with ground truth $B_j^i$, then we assign the following regression offsets $t_{xy}^i = (l, r, t, b)$ to it:

$$l = x_o - x - 1, \quad t = y_o - y - 1, \quad r = x_2 - x, \quad b = y_2 - y, \quad (8)$$

where $b_j^i = (x_1, y_1, x_2, y_2)$ are the bounding box coordinates as defined in Eq. 3. These 4-directional offsets are also showcased in Figure 2. We define $T_i$ to be the set of regression targets $\{t_{xy}^i\}$ assigned to feature map $F_i$.

Since it is possible that locations on multiple FPN levels $F_i$ resolve to the same $(x_o, y_o)$, we disambiguate them by removing offset $t_{xy}^i$ from level $i$ if it does not satisfy the following policy:

$$t_{xy}^i \in T_i, \text{iff} \quad m_i - 1 < \max(l, t, r, b) \leq m_i, \quad (9)$$

where $m_i$ is the heuristic value of maximum object size for target assignment in $F_i$. At each location, we also predict centerness, $o_{xy}^i$ [31]:

$$o_{xy}^i = \sqrt{\frac{\min(l, r)}{\max(l, r)}} \times \frac{\min(t, b)}{\max(t, b)}. \quad (10)$$

During NMS, we multiply the bounding box confidence with predicted centerness to down-weight bounding boxes predicted near object boundaries. For each location, we also predict the object class, $c_{xy}^i$, of the assigned bounding box $B_j^i$. Thus, we have a 6-dimensional label, $(t_{xy}^i, c_{xy}^i, o_{xy}^i)$, at each foreground location $(x_o, y_o)$ on each $F_i$.

4.2. Unified Panoptic Head

We design a unified panoptic head predicting both semantic segmentation and dense bounding boxes from the multi-scale features maps.

**Per Level Predictions** At each location $(x, y)$ of FPN level $F_i$, we predict dense bounding boxes using two feature towers, one for localization and the other for semantics, as shown in Figure 3. Each tower contains 4 sequential convolutional blocks (Conv + GroupNorm + ReLU). The towers are shared across different FPN levels. We directly predict bounding box offsets $t_{xy}$, centerness $o_{xy}$, and bounding
Our model uses ResNet-50-FPN as backbone. The multi-scale feature maps are fed into a unified panoptic head. We predict dense bounding boxes at each FPN Level. We also upsample and concatenate intermediate feature maps across all levels to predict global levelness and semantic logits.

Box class probabilities $\hat{c}_{xy}$. The bounding box offset is predicted from the localization tower and the box class probability distribution is predicted from the semantics tower. We adopt an IoU Loss (11) on bounding box regression:

$$L_{box,reg} = \frac{1}{N_{fg}} \sum_{xy} L_{IoU}(\hat{b}_{xy}, b_{t,xy}) 1_{fg}(x,y), \quad (11)$$

where $N_{fg}$ is the number of foreground (i.e. things) pixels according to the ground truth mask, $\hat{b}_{xy}$ and $b_{t,xy}$ are the absolute bounding box coordinates computed from Eq. 8 using $\hat{t}_{xy}$ and $t_{t,xy}$, and $1_{fg}(x,y)$ is an indicator function yielding 1 when $(x,y)$ corresponds to a foreground.

We compute a loss on predicted centerness at the same locations $(x,y)$ of each FPN level $F_i$ using a Binary Cross Entropy (BCE):

$$L_{center} = \frac{1}{N_{fg}} \sum_{xy} L_{BCE}(\hat{c}_{xy}, c_{xy}) 1_{fg}(x,y). \quad (12)$$

Finally, we predict a probability distribution over object classes $\hat{c}_{xy} \in \mathbb{R}^{N_{thng}}$ for all feature locations $(x,y)$ including background pixels. For our box classification loss $L_{box,cls}$, we use a sigmoid focal loss as in [31], averaged over the total number of locations across all FPN levels.

Global Predictions. In addition to the per level predictions, we leverage the intermediate features from the two towers ($F_{loc}^i$ and $F_{sem}^i$) to globally predict:

1. Levelness $I$: the FPN level that the bounding box at each location $(x,y)$ belongs to ($N_l$= the number of FPN levels +1 logits for each location with 0 reserved for background pixels).
2. Semantic segmentation $S$: the semantic class probability distribution over $N$ classes.

As depicted in Figure 3, we upsample each $F_{loc}^i$ and $F_{sem}^i$ to an intermediate size of $(H/4, W/4)$ and concatenate them into a global $F_{loc}$ and $F_{sem}$. The levelness is predicted from $F_{loc}$ through a single convolutional layer and is supervised by the FPN level assignment policy defined in (9). The levelness is trained using a multi-class cross-entropy loss:

$$L_{levelness} = L_{CE}(I, I). \quad (13)$$

At inference time, for every $(x,y)$ we have one bounding box prediction $b_{xy}^{(i)}$ coming from each FPN level $F_i$. Levelness tells us which $b_{xy}^{(i)}$ to include in our global set of dense bounding box predictions $B$:

$$B(x,y) = b_{xy}^{(argmax I(x,y))}. \quad (14)$$

Instead of using a separate branch for semantic segmentation [15, 36, 13], we reuse the same features as bounding box classification. Doing so dramatically reduces the number of parameters and inference time of the network. We predict the full class semantic logits from $F_{sem}$, which we supervise using a cross-entropy loss:

$$L_{semantics} = L_{CE}(S, S'), \quad (15)$$

where $S'$ denotes semantic labels. We bootstrap this loss to only penalize the worst 30% of predictions as in [34, 27].

4.3. Explicit Mask Loss

As discussed in [31], the quality of bounding box prediction tends to drop with distance from the object center. This hurts the performance of our mask construction near boundaries. In order to refine instance masks, we introduce a loss that aims to reduce False Positive (FP) and False Negative (FN) pixel counts in predicted masks:
\[ L_{\text{mask}} = \frac{1}{|B_{\text{query}}|} \sum_{j} \beta_j \left( E_{\text{FP}_j} + E_{\text{FN}_j} \right), \]  

where \( \beta_j \) is the IoU between proposal \( b_j \) and its associated target box \( b'_j \), \( N_j \) is the count of foreground pixels in the ground truth mask for \( b'_j \), and \( E_{\text{FP}_j} \) and \( E_{\text{FN}_j} \) are proxy measures for the counts of FP and FN pixels in our predicted mask for box \( b_j \):  

\[ E_{\text{FP}_j} = \sum_{xy} \text{IoU}(B(x,y), b_j) \mathbb{I}_{(x,y) \notin M'_j}, \]  

\[ E_{\text{FN}_j} = \sum_{xy} (1 - \text{IoU}(B(x,y), b_j)) \mathbb{I}_{(x,y) \in M'_j}. \]

For Cityscapes experiments, we set \( \lambda \) to 1 for simplicity. For COCO experiments, we drop \( \lambda \) to 0.4 to account for the increased magnitude of the cross-entropy loss that results from the larger ontology.

5. Experiments

In this section, we evaluate our method on standard panoptic segmentation benchmarks. We compare our performance to the state of the art in both accuracy and efficiency. We also provide an extensive ablative analysis.

5.1. Datasets

The Cityscapes panoptic segmentation benchmark [5] consists of urban driving scenes with 19 classes, 8 thing classes, containing instance level labels, and 11 stuff classes. In our experiments, we only use the images annotated with fine-grained labels: 2975 for training, 500 for validation. All the images have a resolution of 1024 × 2048.

COCO [20] is a large-scale object detection and segmentation dataset. Following the standard protocol, we use the 2017 edition with 118k training and 5k validation images. The labels consist of 133 classes with 80 thing classes containing instance level annotations.

5.2. Metrics

We use the Panoptic Quality (PQ) metric proposed by [14] as summary metric:

\[ PQ = \frac{\sum_{(p,g) \in TP} \text{IoU}(p,g) \mathbb{I}_{\text{IoU}(p,g) > 0.5}}{|TP| + \frac{1}{2} |FP| + \frac{1}{2} |FN|}, \]  

where \( p \) and \( g \) are matched predicted and ground-truth segments, and TP, FP, FN denote true positives, false positives, and false negatives, respectively. A positive detection is defined by \( \text{IoU}(p,g) > 0.5 \). We also provide standard metrics on sub-tasks, including Mean IoU for semantic segmentation and average over AP\textsuperscript{r} for instance segmentation.

5.3. Implementation details

All the models in our experiments are implemented in PyTorch and trained using 8 Tesla V100 GPUs. Inference timing is done on Tesla V100 GPU with batch size 1.

For Cityscapes experiments, our models are trained using a batch size of 1 per GPU, weight decay of \( 1e^{-4} \), learning rate of 0.013 for 48k total steps, decreasing by a factor of 0.1 at step 30k and 44k. We apply a random crop of 1800 × 900 and re-scale the crops randomly between (0.7, 1.3). For COCO experiments, our models are trained using a batch size of 2 per GPU, weight decay of \( 1e^{-4} \), learning rate of 0.01 for 180k steps with learning rate with steps at 120k and 160k. For data augmentation, we randomly resize the input image to a shortest side length in (640, 720, 800). No cropping is applied.

All our models use a ResNet-50 Backbone with ImageNet pretrained weights provided by Pytorch [26]. We freeze BatchNorm layers in the backbone for simplicity, similar to [36][15].

5.4. Comparison to State of the Art

We compare the accuracy and efficiency of our proposed model to the state-of-the-art two-stage and single-stage panoptic segmentation algorithms that use a ResNet-50 or lighter backbone. We only report and compare to single-model prediction results to avoid ambiguity of the inference time during test-time augmentation. For inference time, we report the average inference time plus NMS processing time over the whole validation set. For Cityscapes, our model takes the full resolution as input size. For COCO, we resize all images to a longer side of 1333px as input. Our quantitative results are reported in Table 1 and Table 2.

Our model outperforms all the single-stage methods by a significant margin in both accuracy and inference speed. We are also closing the gap between the single-stage and slow but state-of-the-art two-stage methods, even outperforming some of them. To better highlight the potential of our method towards deployment in real-world systems and its parallelization benefits, we conduct a simple optimization by compiling the convolutional layers in our model using TensorRT [2]. It enables our model to operate in real-time (30.3 FPS on a V100 GPU) at full resolution on Cityscapes videos vs. 10.1 FPS without optimization (cf. Figure 1).

We also provide some qualitative examples comparing to one of the best two-stage methods, UPSNet [36], in Figure 4. Our method presents little degradation compared to UPSNet. In fact, our models provide better mask estimation on rare-shape instances (cf. the car with a roof-box,
We provide a simple yet promising extension of our model to the weakly supervised setting in which we have ground truth bounding boxes and semantic segmentation but no instance masks. A few straightforward changes are required: (1) regarding bounding box target assignment described in section 4.1, we now consider a pixel foreground or the small child) thanks to the self-attention in mask association. We observe particularly good results on unusual shapes, because unlike Mask-RCNN style methods[10], our mask construction process can associate pixels with a detection even if they fall outside its bounding box.

### 5.5. Ablative Analysis

We provide an ablative analysis on the key modules of our method in Table3:

| Method          | Backbone         | PQ  | PQ\textsuperscript{th} | PQ\textsuperscript{st} | mIoU | AP  | GPU  | Inference Time |
|-----------------|------------------|-----|-------------------------|-------------------------|------|-----|------|----------------|
| **Two-Stage**   |                  |     |                         |                         |      |     |      |                |
| TASCNet [15]    | ResNet-50-FPN    | 55.9| 50.5                    | 59.8                    | -    | -   | V100 | 160ms          |
| AUNet [16]      | ResNet-50-FPN    | 56.4| 52.7                    | 59.0                    | 73.6 | 33.6| -    | -              |
| Panoptic-FPN    | ResNet-50-FPN    | 57.7| 51.6                    | 62.2                    | 75.0 | 32.0| -    | -              |
| AdapTS [30]     | ResNet-50        | 59.0| 55.8                    | 61.3                    | 75.3 | 32.3| -    | -              |
| UPSNet [36]     | ResNet-50-FPN    | 59.3| 54.6                    | 62.7                    | 75.2 | 33.3| V100 | 140ms\*        |
| Seamless Panoptic [28] | ResNet-50-FPN | 60.2| 55.6                    | 63.6                    | 74.9 | 33.3| V100 | 150ms\*        |

| **Single-Stage**|                  |     |                         |                         |      |     |      |                |
| DeeperLab [38]  | Wider MNV2       | 52.3| -                       | -                       | -    | -   | V100 | 251ms          |
| FPSNet [37]     | ResNet-50-FPN   | 55.1| 48.3                    | 60.1                    | -    | -   | TITAN RTX | 114ms          |
| SSAP [38]       | ResNet-50       | 56.6| 49.2                    | -                       | -    | 31.5| 1080Ti | >260ms         |
| DeeperLab [38]  | Xception-71     | 56.5| -                       | -                       | -    | -   | V100  | 312ms          |
| Ours            | ResNet-50-FPN   | 58.8| 52.1                    | 63.7                    | 77.0 | 29.8| V100  | 99ms \*        |

Table 2: **Performance on COCO-validation.** We bold the best single-stage methods and underline the best number across the two categories. †: method includes multiple-forward passes. ∗: Our replicated result from official sources using the same evaluation environment as our model.

Table 3: **Ablative analysis.** We compare the impact of different key modules/designs in our proposed network. We also present a weakly supervised model trained without using instance masks.

| Method               | Backbone         | PQ  | PQ\textsuperscript{th} | PQ\textsuperscript{st} | mIoU | AP  | GPU  | Inference Time |
|----------------------|------------------|-----|-------------------------|-------------------------|------|-----|------|----------------|
| **Two towers**       |                  |     |                         |                         |      |     |      |                |
| Panoptic-FPN [13]    | ResNet-50-FPN    | 33.3| 45.9                    | 28.7                    | -    | -   | -    | 94ms           |
| AdapTS [30]          | ResNet-50        | 35.9| 40.3                    | 29.3                    | -    | -   | -    | 94ms           |
| AUNet [16]           | ResNet-50-FPN    | 39.6| 49.1                    | 25.2                    | -    | -   | -    | 94ms           |
| UPSNet [36]          | ResNet-50-FPN    | 42.5| 48.5                    | 33.4                    | 110ms\* | -   | -    | 94ms           |
| **Single-Stage**     |                  |     |                         |                         |      |     |      |                |
| DeeperLab [38]       | Xception-71      | 33.8| -                       | -                       | -    | -   | -    | 94ms           |
| SSAP [38]            | ResNet-50        | 36.5| -                       | -                       | -    | -   | -    | 94ms           |
| Ours                 | ResNet-50-FPN    | 37.1| 41.0                    | 31.3                    | 63ms | -   | -    | 94ms           |

Then, we introduce levelness, which leads to almost a 1 point boost in PQ. Without levelness, it is still possible to compute \(P_{IoU}\) as in Eq. 6. We can compute the \(IoU\) between each query box and the predicted bounding boxes from every FPN level, and then take a max along the FPN levels. However, this operation suffers from ambiguity between object boundaries and background.

Finally, in the fourth row, we introduce our explicit mask loss from Section 4.3 which further refines mask association and construction for thing classes resulting in a higher PQ\textsuperscript{th} and corresponding bump in PQ.

### 5.6. Weakly supervised extension

We provide a simple yet promising extension of our model to the weakly supervised setting in which we have ground truth bounding boxes and semantic segmentation but no instance masks. A few straightforward changes are required: (1) regarding bounding box target assignment described in section 4.1, we now consider a pixel foreground as long as it is inside a ground truth bounding box, (2)
we update the foreground indicator function \( \mathbb{1}_{fg}(x, y) \) in Eq. 11 and Eq. 12 from ‘in-mask’ to ‘in-box’, and (3) we train without using the Mask Loss. Our weakly supervised Cityscapes model achieves promising accuracy and even outperforms some fully supervised methods, as shown in the last row of Table 3.

6. Conclusion

In this work, we propose a single-stage panoptic segmentation framework that achieves real-time inference with a performance competitive with the current state of the art. We first introduce a novel parameter-free mask construction operation that reuses predictions from dense object detection via a global self-attention mechanism. Our architecture dramatically decreases computational complexity associated with instance segmentation in conventional panoptic segmentation algorithms. Additionally, we develop an explicit mask loss which improves panoptic segmentation quality. Finally, we evaluate the potential of our method in weakly supervised settings, showing that it can outperform some recent fully supervised methods.

Acknowledgments

We would like to thank the TRI ML team, especially Dennis Park and Wolfram Burgard for their support and insightful comments during the development of this work.

Figure 4: Panoptic segmentation results on CityScapes and COCO comparing our predictions and UPSNet [36]. We leave it to the reader to guess which results are ours (the answer is in [1]).
Appendix

We provide more detailed descriptions, metrics, and visualizations for our proposed approach that were not included in the main text due to space limitation:

• per-class metrics of our model on Cityscapes and COCO (Tables 1 and 2 in the main text);
• implementation details for the comparison without levelness presented in the ablative analysis (Table 3 in the main text);
• detailed description and discussion of the proposed weakly supervised application of our method (Section 5.6 Table 3 in the main text);
• discussion on the degradation curve of our method with lighter backbones.

A. Per-class Performance

We provide per-class PQ metrics of our models on Cityscapes and COCO dataset in Table A4[13] and Table A6[14] which corresponds respectively to the entries in Table 1 and Table 2 in the main text. For both datasets, \( \sigma = 0.3 \) is used as the foreground mask acceptance probability in Eq. 8, which is determined through hyper-parameter grid search.

B. Contribution of Levelness Map

As we have shown in the paper (also depicted in Figure 3), our proposed model predicts a global levelness map using feature maps produced by the localization tower at all scale levels. This levelness map indicates which scale level does the bounding box at each location \((x, y)\) belongs to (with \(0\) reserved for background). At inference time, the levelness map is used to provide indexes while we assemble the global dense bounding box prediction \(B(x, y)\) from each FPN level \((B_{xy}^{i})\), according to Eq. 15.

However, the levelness is not a necessity in our proposed model, as the assembling process can be done without it. In our ablative analysis in Section 5.5, we compared our method to a simple alternative assembling approach without levelness to justify the value of such design.

This alternative solution, instead of assembling a unique bounding box for each feature location \((x, y)\), carries all the bounding box predictions from different FPN levels \(i\) to the foreground mask probability estimation:

\[
B(x, y; i) = b_{xy}^{i} \tag{A21}
\]

Then the location based foreground probability becomes:

\[
\hat{P}_{\text{loc}}(x, y; i; j) = \text{IoU}(B(x, y; i), B_{j}) \tag{A22}
\]

We can still obtain a single probability for each location by taking the maximum along the level dimension \(i\):

\[
\hat{P}_{\text{loc}}(x, y, j) = \arg\max_{i}(\hat{P}_{\text{loc}}(x, y; j)) \tag{A23}
\]

We report the resulting model performance of this alternative solution in the second entry of the ablative analysis table (Table 3 in the main text). For simplicity, this comparison on levelness is done without the mask loss. All other configurations are the same as the default model. This comparison indicates that the levelness map, with only 1 additional convolutional layer, is able to provide a better cross-level indication resulting in an increase in performance (+1\% PQ).

| Class     | PQ  | SQ  | RQ  |
|-----------|-----|-----|-----|
| road      | 97.58 | 97.88 | 99.69 |
| building  | 87.67 | 89.49 | 97.96 |
| fence     | 34.89 | 73.27 | 47.62 |
| traffic light | 45.98 | 70.97 | 64.79 |
| vegetation | 88.78 | 90.27 | 98.35 |
| sky       | 86.73 | 92.16 | 94.10 |
| rider     | 49.61 | 70.76 | 70.11 |
| truck     | 47.16 | 83.12 | 56.74 |
| bus       | 60.55 | 83.30 | 72.69 |
| train     | 55.94 | 83.91 | 66.67 |
| motorcycle| 44.54 | 72.68 | 61.28 |

Table A4: Per-class Performance on Cityscapes

C. Weakly Supervised Application

In the weakly supervised scenario discussed in Section 5.6 of the main text, we consider relying only on semantic and bounding box labels for the panoptic segmentation task.

We report the resulting model performance of this alternative solution in the second entry of the ablative analysis table (Table 3 in the main text). For simplicity, this comparison on levelness is done without the mask loss. All other configurations are the same as the default model. This comparison indicates that the levelness map, with only 1 additional convolutional layer, is able to provide a better cross-level indication resulting in an increase in performance (+1\% PQ).

These labels present a weaker supervision for the panoptic segmentation task, as there are ambiguous pixels between overlapping bounding boxes of the same category.

In our proposed algorithm, we directly regress the bounding box and semantic classification. The construction of instance masks relies on the accuracy of the bounding box predictions instead of the explicit modeling of instance shapes. Thus, our method is robust to the absence of foreground mask information. In our implementation, we relax the pixel assignment during bounding box prediction as depicted in Figure A5 favoring the smallest bounding box in overlapping cases.

As shown in Table 3, our weakly supervised model achieved 55.7 PQ, i.e. 95\% of the performance of the fully supervised model. Some qualitative examples are also provided in Figure A6.
Figure A6: Weakly supervised panoptic segmentation
Our proposed algorithm obtain promising and practical prediction results, trained with only bounding box and semantic labels,

Table A5: Performance of Our Proposed Framework with Different Configuration on CityScapes

D. Degradation Curves

In the experiment section of the main text, we provide performance analysis with the best performing configuration of our models. However, in real-world systems, it is a common practice to apply light-weight backbones and low-resolution input images for better inference efficiency at the expense of accuracy. In this section, we provide a wider spectrum of the expected performance of our proposed method when working with downgraded configurations. We train our proposed framework with multiple ResNet-FPN backbones of different depths. More specifically, we replace our default ResNet-50-FPN backbone with ResNet-34-FPN and ResNet-18-FPN, and train these two models using the same set of hyper-parameters (e.g. learning rate) as the default one. The two shallower backbones are also pre-trained with the ImageNet dataset. We test the trained models on the CityScapes dataset and report the results in terms of accuracy and inference speed. The results are presented in Table A5. As expected, shallower backbones decrease model performance especially for instance segmentation, but further accelerates inference speed. We also train and evaluate our models on CityScapes with images of half of the original size, namely 512 × 1024 px. We report the performance in Table A5 as well.

Table A6: Per-class Performance on COCO
References

[1] Ours results are in the middle for row 2,5 and on the right for row 1,3,4.

[2] TensorRT python library. https://developer.nvidia.com/tensorrt

[3] Daniel Bolya, Chong Zhou, Fanxi Xiao, and Yong Jae Lee. YOLACT: real-time instance segmentation. CoRR, abs/1904.02689, 2019.

[4] Kai Chen, Jiannin Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Zhiwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4974–4983, 2019.

[5] 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 Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3213–3223, 2016.

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

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

[8] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaichi Huang. Ssap: Single-shot instance segmentation with affinity pyramid. arXiv preprint arXiv:1909.01616, 2019.

[9] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In European Conference on Computer Vision, pages 297–312. Springer, 2014.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[12] Zhaojin Huang, Lichao Huang, Yongchao Wang, Chang Huang, and Xingang Wang. Mask scoring r-cnn. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6409–6418, 2019.

[13] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6399–6408, 2019.

[14] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9404–9413, 2019.

[15] Jie Li, Allan Raventos, Arjun Bhargava, Takaaki Tagawa, and Adrien Gaidon. Learning to fuse things and stuff. arXiv preprint arXiv:1812.01192, 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 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7026–7035, 2019.

[17] Xiaodan Liang, Liang Lin, Yunchao Wei, Xiaohui Shen, Jianchao Yang, and Shuicheng Yan. Proposal-free network for instance-level object segmentation. IEEE transactions on pattern analysis and machine intelligence, 40(12):2978–2991, 2017.

[18] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2117–2125, 2017.

[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014.

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

[22] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8759–8768, 2018.

[23] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In Proceedings of the IEEE International Conference on Computer Vision, pages 4990–4999, 2017.

[24] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8837–8845, 2019.

[25] Alejandro Newell, Zhiqun Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In Advances in Neural Information Processing Systems, pages 2277–2287, 2017.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc., 2019.
[27] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4151–4160, 2017.

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

[29] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

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

[31] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. *arXiv preprint arXiv:1904.01355*, 2019.

[32] Jonas Uhrig, Eike Rehder, Björn Fröhlich, Uwe Franke, and Thomas Brox. Box2pix: Single-shot instance segmentation by assigning pixels to object boxes. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 292–299. IEEE, 2018.

[33] Mark Weber, Jonathon Luiten, and Bastian Leibe. Single-shot panoptic segmentation, 2019.

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

[35] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. *arXiv preprint arXiv:1909.13226*, 2019.

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

[37] Wenqiang Xu, Haiyang Wang, Fubo Qi, and Cewu Lu. Explicit shape encoding for real-time instance segmentation. *arXiv preprint arXiv:1908.04067*, 2019.

[38] 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. *CoRR*, abs/1902.05093, 2019.

[39] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 516–520. ACM, 2016.