This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Seamless Scene Segmentation

Lorenzo Porzi, Samuel Rota Bulò, Aleksander Colovic, Peter Kontschieder
Mapillary Research
research@mapillary.com

Abstract

In this work we introduce a novel, CNN-based architecture that can be trained end-to-end to deliver seamless scene segmentation results. Our goal is to predict consistent semantic segmentation and detection results by means of a panoptic output format, going beyond the simple combination of independently trained segmentation and detection models. The proposed architecture takes advantage of a novel segmentation head that seamlessly integrates multi-scale features generated by a Feature Pyramid Network with contextual information conveyed by a light-weight DeepLab-like module. As additional contribution we review the panoptic metric and propose an alternative that overcomes its limitations when evaluating non-instance categories. Our proposed network architecture yields state-of-the-art results on three challenging street-level datasets, i.e. *Cityscapes, Indian Driving Dataset and Mapillary Vistas.*

1. Introduction

Scene understanding is one of the grand goals for automated perception that requires advanced visual comprehension of tasks like semantic segmentation (*Which semantic category does a pixel belong to?*) and detection or instance-specific semantic segmentation (*Which individual object segmentation mask does a pixel belong to?*). Solving these tasks has large impact on a number of applications, including autonomous driving or augmented reality. Interestingly, and despite sharing some obvious commonalities, both these segmentation tasks have been predominantly handled in a disjoint way ever since the rise of deep learning, while earlier works [49, 50, 56] already approached them in a joint manner. Instead, independent trainings of models, with separate evaluations using corresponding performance metrics, and final fusion in a post-processing step based on task-specific heuristics have seen a revival.

The work in [24] introduces a so-called *panoptic* evaluation metric for joint assessment of semantic segmentation of *stuff* and instance-specific *thing* object categories, to encourage further research on this topic. Stuff is defined as non-countable, amorphous regions of similar texture or material while things are enumerable, and have a defined shape. Few works have started adopting the panoptic metric in their methodology yet, but reported results remain significantly below the ones obtained from fused, individual models. All winning entries on designated panoptic Segmentation challenges like *e.g.* the Joint COCO and Mapillary Recognition Workshop 2018[1], were based on combinations of individual (pre-trained) segmentation and instance segmentation models, rather than introducing streamlined integrations that can be successfully trained from scratch.

The use of separate models for semantic segmentation and detection obviously comes with the disadvantage of significant computational overhead. Furthermore, and due to a lack of cross-pollination of models, there is no way of enforcing labeling consistency between individual models. Moreover, we argue that individual models supposedly spend significant amounts of their capacity on modeling redundant information, whereas sensible architectural choices in a joint setting are leading to favorable or on par results, but at much reduced computational costs.

In this work we introduce a novel, deep convolutional neural network based architecture for *seamless scene segmentation*. Our proposed network design aims at jointly addressing the tasks of semantic segmentation and instance segmentation. We present ideas for interleaving information from segmentation and instance-segmentation modules and discuss model modifications over vanilla combinations of standard segmentation and detection building blocks. With our findings, we are able to train high-quality, seamless scene segmentation models without the need of pre-trained recognition models. As result, we obtain a state-of-the-art, single model that jointly produces semantic segmentation and instance segmentation results, at a fraction of the computational cost required when combining independently trained recognition models.

We provide the following contributions in our work:

- Streamlined architecture based on a single network backbone to generate complete semantic scene segmentation for stuff and thing classes
- A novel segmentation head integrating multi-scale features from Feature Pyramid Network, with contextual information provided by a light-weight, DeepLab-inspired module

[1] http://cocodataset.org/workshop/coco-mapillary-eccv-2018.html

- Re-evaluation of the panoptic segmentation metric and refinement for more adequate handling of stuff classes
- Comparisons of the proposed architecture against individually trained and fused segmentation models, including analyses of model parameters and computational requirements
- Experimental results on challenging driving scene datasets like Cityscapes [10], Indian Driving Dataset [51], and Mapillary Vistas [39], demonstrating state-of-the-art performance.

2. Related Works

Semantic segmentation is a long-standing problem in computer vision research [3, 26, 27, 48] that has significantly improved over the past five years, thanks in great part to advances in deep learning. The works in [2, 38] have introduced encoder/decoder CNN architectures for providing dense, pixel-wise predictions by taking *e.g*. a fully convolutional approach. The more recent DeepLab [5] exploits multi-scale features via parallel filters from convolutions with different dilation factors, together with globally pooled features. Another recent Deeplab extension [9] integrates a decoder module for refining object boundary segmentation results. In [8], a meta-learning technique for dense prediction tasks is introduced, that learns how to design a decoder for semantic segmentation. The pyramid scene parsing network [59] employs i) a pyramidal pooling module to capture sub-region representations at different scales, followed by upsampling and stacking with respective input features and ii) an auxiliary loss applied after the *conv4* block of a ResNet-101 backbone. The works in [57, 58] propose aggregation of multi-scale contextual information using dilated convolutions, which have proven to be particularly effective for dense prediction tasks, and are a generalization of the conventional convolution operator to expand its receptive field. RefineNet [32] proposes a multi-path refinement network to exploit multiple abstraction levels of features for enhancing the segmentation quality of high-resolution images. Other works like [45, 53] are addressing the problem of class sample imbalance by introducing loss-guided, pixel-wise gradient reweighting schemes.

Instance-specific semantic segmentation has recently gained large attention in the field, with early, random field-based works in [21, 49]. In [17] a simultaneous detection and segmentation algorithm is developed that classifies and refines CNN features obtained from regions under R-CNN [16] bounding box proposals. The work in [18] emphasizes on refining object boundaries for binary segmentation masks initially generated from bounding box proposals. In [12] a multi-task network cascade is introduced that, beyond sharing features from the encoder in all following tasks, subsequently adds blocks for i) bounding box generation, ii) instance mask generation and iii) mask categorization. Another approach [11] introduces instance fully convolutional networks that assemble segmentations from position-sensitive score maps, generated by classifying pixels based on their relative positions. The follow-up work in [31] builds upon Faster R-CNN [43] for proposal generation and additionally includes position-sensitive outside score maps. InstanceCUT [25] obtains instance segmentations by solving a Multi-Cut problem, taking instance-agnostic semantic segmentation masks and instance-aware, probabilistic boundary masks as inputs, provided by a CNN. The work in [1] also introduces an approach where an instance CRF provides individual instance masks based on exploiting box, global and shape cues as unary potentials, together with instance-agnostic semantic information. In [36], sequential grouping networks are presented that run a sequence of simple networks for solving increasingly complex grouping problems, eventually yielding instance segmentation masks. DeepMask [40] first produces an instance-agnostic segmentation mask for an input patch, which is then assigned to a score corresponding to how likely this patch it to contain an object. At inference, their approach generates a set of ranked segmentation proposals. The follow-up work SharpMask [41] augments the networks with a top-down refinement approach. Mask R-CNN [19] forms the basis of current state-of-the-art instance segmentation approaches. It is a conceptually simple extension of Faster R-CNN, adding a dedicated branch for object mask segmentation in parallel to the existing ones for bounding box regression and classification. Due to its importance in our work, we provide a more thorough review in the next section. The work in [37] proposes to improve localization quality of objects in Mask R-CNN via integration of multi-scale information as bottom-up path augmentation.

Joint segmentation and instance-segmentation approaches date back to [50], introducing a Bayesian approach for scene representation by establishing a scene parsing graph to explain both, segmentation of stuff and things. Other works before the era of deep learning often built upon CRFs where [49] alternatingly refined pixel labelings and object instance predictions, and [56] framed holistic scene understanding as a structure prediction problem in a graphical model, defined over hierarchies of regions, scene types, *etc*. The recently proposed work in [22] addresses automated loss balancing in a multi-task learning problem based on analysing the homoscedastic uncertainty of each task. Even though their work addresses three tasks at the same time (semantic segmentation, instance segmentation and depth estimation), it fails to demonstrate consistent improvements over semantic segmentation and instance segmentation alone and lacks of comparisons to comparable baselines. The supervised variant in [30] generates panoptic segmentation results but *i)* requires separate (external)

input for bounding box proposals and *ii)* exploits a conditional random field during inference, increasing the complexity of the model. The work in [14] attempts to introduce a unified architecture related to our ideas, however, the reported results remain significantly below those of reported state-of-the-art methods. Independently and simultaneously to our paper, a number of works [23, 29, 35, 54, 55] have proposed panoptic segmentation provided by a single deep network, confirming the importance of this task to the field. While comparable in complexity and architecture, we obtain improved performance on challenging street-level image datasets like Cityscapes and Mapillary Vistas.

3. Proposed Architecture

The proposed architecture consists of a backbone working as feature extractor and two task-specific branches addressing semantic segmentation and instance segmentation, respectively. Hereafter, we provide details about each component and refer to Fig. 1 for an overview.

3.1. Shared Backbone

The backbone that we use throughout this paper is a slightly modified ResNet-50 [20] with a Feature Pyramid Network (FPN) [33] on top. The FPN network is linked to the output of the modules conv2, conv3, conv4 and conv5 of ResNet-50, which yield different downsampling factor, namely $\times 4$, $\times 8$, $\times 16$ and $\times 32$, respectively. Akin to the original FPN architecture, we have a variable number of additional lower resolution scales covering downsampling factors of $\times 64$ and $\times 128$, depending on the dataset. The main modification in ResNet-50 is the replacement of all Batch Normalization (BN) + ReLU layers with synchronized Inplace Activated Batch Normalization (iABN$^{\text{sync}}$) proposed in [46], which uses LeakyReLU with slope 0.01 as activation function due to the need of invertible activation functions. This modification gives two important advantages: we gain up to 50% additional GPU memory since the layer performs in-place operations, and the synchronization across GPUs ensures a better estimate of the gradients in multi-GPU trainings with positive effects on convergence.

3.2. Instance Segmentation Branch

The instance segmentation branch follows the state-of-the-art Mask R-CNN [19] architecture, but with some modifications described next. This branch is structured into a region proposal head and a region segmentation head.

Region Proposal Head (RPH). The RPH introduces the notion of an *anchor*. An anchor is a reference bounding box (*a.k.a.* region), centered on one of the available spatial locations of the RPH's input and having pre-defined dimensions. The set of pre-defined dimensions is chosen in advance, depending on the dataset and the scale of the FPN output (see details in Sec. 5). We denote by $\mathcal{A}$ all anchors that can be constructed by combining a position on an available, spatial location and a dimension from the pre-defined set, and which are entirely contained in the image. Given an anchor a we denote its position (in the image coordinate system) by (u_a, v_a) and its dimensions by (w_a, h_a). The role of RPH is to apply a transformation to each anchor in order to obtain a new bounding box proposal together with an objectness score, that assesses the validity of the region. To this end, RPH applies a 3×3 convolution with 256 output channels and stride 2 to the outputs of the backbone, followed by iABN$^{\text{sync}}$, and a 1×1 convolution with $5N_{\text{anchors}}$ channels, which provide a bounding box proposal with an objectness score for each anchor in $\mathcal{A}$. In more details, for each anchor $a \in \mathcal{A}$ the transformed bounding box has center $(\hat{u}, \hat{v}) = (u_a + o_u w_a, v_a + o_v h_a)$, dimensions $(\hat{w}, \hat{h}) = (w_a\, e^{o_w}, h_a\, e^{o_h})$ and objectness score $\hat{s} = \sigma(o_s)$, where $(o_u, o_v, o_w, o_h, o_s)$ represents the output from the 1×1 convolution for anchor a, and $\sigma(\cdot)$ is the sigmoid function. The resulting set of bounding boxes are then fed to the region segmentation head, with distinct filtering steps for training and test time.

Region Segmentation Head (RSH). Each region proposal $\hat{r} = (\hat{u}, \hat{v}, \hat{w}, \hat{h})$ obtained from RPH is fed to RSH, which applies ROIAlign [19], pooling features directly from the kth output of the backbone within region $\hat{r}$ with a 14×14 spatial resolution, where k is selected based on the scale of $\hat{r}$ according to the formula $k = \max(1, \min(4, \lfloor 3 + \log_2(\sqrt{\hat{w}\hat{h}}/224) \rfloor))$ [19]. The result is forwarded to two parallel sub-branches: one devoted to predicting a class label (or void) for the region proposal together with class-specific corrections of the proposal's bounding box, and the other devoted to providing class-specific mask segmentations. The first sub-branch of RSH is composed of two fully-connected layers with 1024 channels, each followed by Group Normalization (GN) [52] and LeakyReLU with slope 0.01, and a final fully-connected layer with $5N_{\text{classes}} + 1$ output units. The output units encode, for each possible class c, class-specific correction factors $(o_u^c, o_v^c, o_w^c, o_h^c)$ that are used to compute a new bounding box centered in $(\hat{u}^c, \hat{v}^c) = (\hat{u}_o + o_u^c \hat{w}, \hat{v}_r + o_v^c \hat{h})$ with dimensions $(\hat{w}^c, \hat{h}^c) = (\hat{w}\, e^{o_w^c}, \hat{h}\, e^{o_h^c})$. This operation generates from $\hat{r}$ and for each class c a new class-specific region proposals given by $\hat{r}^c = (\hat{u}^c, \hat{v}^c, \hat{w}^c, \hat{h}^c)$. In addition, we have $N_{\text{classes}} + 1$ units providing logits for a softmax layer that gives a probability distribution over classes and void, the latter label assessing the invalidity of the proposal. The probability associated to class c is used as score function $\hat{s}^c$ for the class-specific region proposal $\hat{r}^c$. The second sub-branch applies four 3×3 convolution layers each with 256 output channels. As for the first sub-branch each convolution is followed by GN and LeakyReLU. This is followed

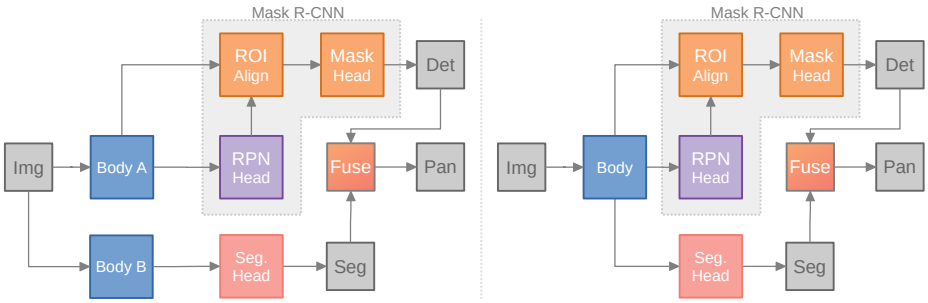

Figure 1: Comparison of two architectures for panoptic segmentation. Left: Separate models (including bodies) for detection and segmentation. Both predictions are fused to obtain the final panoptic prediction. Right: Shared body between the heads.

by a 2×2 deconvolution layer with output stride 2 and 256 output channels, GN, LeakyReLU, and a final 1×1 convolution with N_{classes} output channels. This yields, for each class, 28×28 logits that provide class-specific mask foreground probabilities for the given region proposal via a sigmoid. The resulting mask prediction is combined with the output of the segmentation branch described below.

3.3. Semantic Segmentation Branch

The semantic segmentation branch takes in input the outputs of the backbone corresponding to the first four scales of FPN. We apply independently to each input (not sharing parameters) a variant of the DeepLabV3 head [6] that we call *Mini-DeepLab* (MiniDL, see Fig. 2) followed by an upsampling operation that yields an output downsampling factor of $\times 4$ and 128 output channels. All the resulting streams are concatenated and the result is fed to a final 1×1 convolution layer with N_{classes} output channels. The output is bilinearly upsampled to the size of the input image. This provides the logits for a final softmax layer that provides class probabilities for each pixel of the input image. Note that each convolution in the semantic segmentation branch, including MiniDL, is followed by iABN$^{\text{sync}}$ akin to the backbone.

MiniDL. The MiniDL module consists of 3 parallel sub-branches. The first two apply a 3×3 convolution with 128 output channels with dilations 1 and 6, respectively. The third one applies a 64×64 average pooling operation with stride 1 followed by a padding with boundary replication to recover the spatial resolution of the input and a 1×1 convolution with 128 output channels. The outputs of the 3 sub-branches are concatenated and fed into a 3×3 convolution layer with 128 output channels, which delivers the final output of the MiniDL module.

As opposed to DeepLabV3, we do not perform the global pooling operation in our MiniDL module for two reasons: i) it breaks translation equivariance if we change the input resolution at test time, which is typically the case and ii) since we work with large input resolutions, it is preferable to limit the extent of contextual information. Instead, we replaced the global pooling operation with average pooling in the 3rd sub-branch with a fixed large kernel size and stride 1, but without padding. The lack of padding yields an output resolution which is smaller than the input resolution and we re-establish the input resolution by replicating the boundary of the resulting tensor, *i.e.* we employ a padding layer with boundary replication. By doing so, we generalize the solution originally implemented in DeepLabV3, for we obtain the same output at training time if we keep the kernel size equal to the *training* input resolution, but we preserve translation equivariance at test time, and can reduce the extent of contextual information by properly fixing the kernel size.

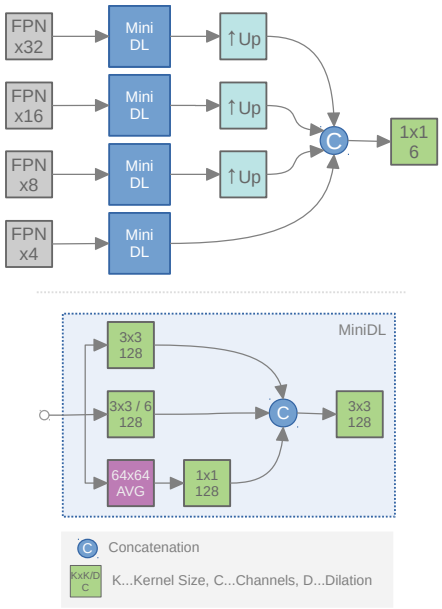

Figure 2: Segmentation Head (top) and the architecture of the Mini Deeplab (MiniDL) module (bottom), which is used in the head.

3.4. Training losses

The two branches of the architecture are supported with distinct losses, which are detailed below. We denote by

$\mathcal{Y} = \{1, \ldots, N_{\text{classes}}\}$ the set of class labels, and assume for simplicity input images with fixed resolution $H \times W$.

Semantic segmentation branch. Let $Y_{ij} \in \mathcal{Y}$ be the semantic segmentation ground truth for a given image and pixel position (i, j) and let $P_{ij}(c)$ denote the predicted probability for the same pixel to be assigned class $c \in \mathcal{Y}$. The per-image segmentation loss that we employ is a weighted per-pixel log-loss that is given by

$$L_{ss}(P, Y) = -\sum_{ij} \omega_{ij} \log P_{ij}(Y_{ij}) \,.$$

The weights are computed following the simplest version of [45] with $p = 1$ and $\tau = \frac{4}{WH}$. This corresponds to having a pixel-wise hard negative mining, which selects the 25% worst predictions, *i.e.* $\omega_{ij} = \tau$ for all (i, j) within the 25% pixels yielding the lowest probability $P_{ij}(Y_{ij})$, and $\omega_{ij} = 0$ otherwise.

Instance segmentation branch. The losses for the instance segmentation branch and the training procedure are derived from the ones proposed in Mask R-CNN [19]. We refer to [42] for additional details due to the lack of space.

3.5. Testing and Panoptic Fusion

At test time, given an input image I we extract features F with the backbone and generate region proposals with corresponding objectness scores by applying RPH. We filter the resulting set of bounding boxes with Non-Maxima Suppression (NMS) guided by the objectness scores. The surviving proposals are fed to the RSH (first sub-branch) together with F in order to generate class-specific region proposals with corresponding class probabilities. A second NMS pass is applied on the resulting set of bounding boxes, this time independently per class guided by the class probabilities. The resulting class-specific bounding boxes are fed again to RSH together with F, but this time through the second sub-branch which provides the corresponding mask predictions. The extracted features F are fed in parallel to the segmentation branch, which provides class probabilities for each pixel. The output of RSH and the segmentation branch are finally fused using the strategy given below, in order to deliver the final panoptic segmentation.

Fusion. The fusion operation is inspired by the one proposed in [24]. We start iterating over predicted instances in reverse classification score order. For each instance we mark the pixels in the final output that belong to it and are still unassigned, provided that the latter number of pixels covers at least 50% of the instance. Otherwise we discard the instance thus resembling a NMS procedure. Remaining unassigned pixels take the most likely class according to the segmentation head prediction, if it belongs to stuff, or void if it belongs to thing. Finally, if the total amount of pixels of any stuff class is smaller than a given threshold (4096 in our case) we mark all those pixels to void.

4. Revisiting Panoptic Segmentation

In this section we review the panoptic segmentation metric [24] (*a.k.a.* PQ metric), which evaluates the performance of a so-called panoptic segmentation, and discuss a limitation of this metric when it comes to stuff classes.

PQ metric. A panoptic segmentation assigns each pixel a stuff class label or an instance ID. Instance IDs are further given a thing class label (*e.g.* pedestrian, car, *etc.*). As opposed to AP metrics used in detection, instances are not overlapping. The PQ metric is computed for each class independently and averaged over classes (void class excluded). This makes the metric insensitive to imbalanced class distributions. Given a set of ground truth segments $\mathcal{S}_c$ and predicted segments $\hat{\mathcal{S}}_c$ for a given class c, the metric collects a set of True Positive matches as $\text{TP}_c = \{(s, \hat{s}) \in \mathcal{S}_c \times \hat{\mathcal{S}}_c \,:\, \text{IoU}(s, \hat{s}) > 0.5\}$. This set contains all pairs of ground truth and predicted segments that overlap in terms of IoU more than 0.5. By construction, every ground truth segment can be assigned at most one predicted segment and vice versa. The PQ metric for class c is given by

$$\text{PQ}_c = \frac{\sum_{(s,\hat{s}) \in \text{TP}_c} \text{IoU}(s, \hat{s})}{|\text{TP}_c| + \frac{1}{2}|\text{FP}_c| + \frac{1}{2}|\text{FN}_c|} \,,$$

where FP_c is the set False Positives, *i.e.* unmatched predicted segments for class c, and FN_c is the set False Negatives, *i.e.* unmatched segments from ground truth for class c. The metric allows also specification of void classes, both in ground truth and actual predictions. Pixels labeled as void in the ground truth are not counted in IoU computations and predicted segments of any class c that overlap with void more than 50% are not counted in FP_c. Also, ground truth segments for class c that overlap with predicted void pixels more than 50% are not counted in FN_c. The final PQ metric is obtained by averaging the class-specific PQ scores:

$$\text{PQ} = \frac{1}{N_{\text{classes}}} \sum_{c \in \mathcal{Y}} \text{PQ}_c \,.$$

We further denote by PQ_{Th} and PQ_{St} the average of thing-specific and stuff-specific PQ scores, respectively.

The issue with stuff classes. One limitation of the PQ metric is that it over-penalizes errors related to stuff classes, which are by definition not organized into instances. This derives from the fact that the metric does not distinguish stuff and thing classes and applies indiscriminately the rule that we have a true positive if the ground truth and the predicted segment have IoU greater than 0.5. De facto it regards all pixels in an image belonging to a stuff class as a single big instance. To give an example of why we think this is sub-optimal, consider a street scene with two sidewalks and assume that the algorithm confuses one of the two with road (say the largest) then the segmentation quality

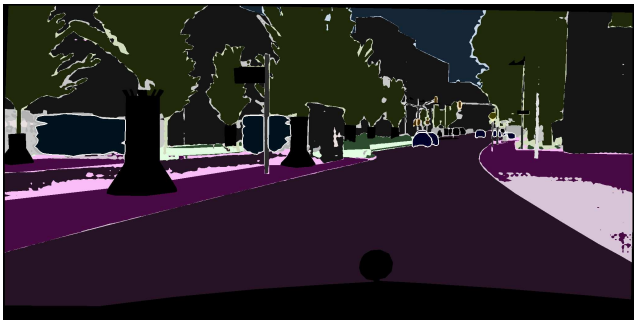

Figure 3: Prediction on a Cityscapes validation set image, where light colored areas highlight conducted errors. Several classes, *e.g*. pole (IoU 0.49) and traffic light (IoU 0.46), are just below the PQ acceptance threshold, while the sidewalk class (IoU 0.62) is just above it. Thus, the former will be overly penalized (PQ $\rightarrow$ 0), while the latter will contribute positively (PQ $\rightarrow$ 0.62), even if they look qualitatively similar. Best viewed in color and with digital zoom.

on sidewalk for that image becomes 0. A real-world example is provided in Fig. 3, where several stuff segments are severely penalized by the PQ metric, not reflecting the real quality of the segmentation. The >0.5-IoU rule for thing classes is convenient because it renders the matching between predicted and ground truth instances easy, but this is a problem to be solved only for thing classes. Indeed, predicted and ground truth segments belonging to stuff classes can be directly matched independently from their IoU because each image has at most one instance of them.

Suggested alternative. We propose to maintain the PQ metric only for thing classes, but change the metric for stuff classes. Specifically, let S_c be the set of ground truth segments of a given class c and let $\hat{S}_c$ be the set of predicted segments for class c. Note that each image can have at most 1 ground truth segment and at most 1 predicted segment of the given stuff class. Let $\mathcal{M}_c = \{(s, \hat{s}) \in S_c \times \hat{S}_c : \mathrm{IoU}(s, \hat{s}) > 0\}$ be the set of matching segments, then the updated metric for class c becomes:

$$\mathrm{PQ}^{\dagger}_c = \begin{cases} \frac{1}{|S_c|} \sum_{(s,\hat{s}) \in \mathcal{M}_c} \mathrm{IoU}(s, \hat{s}) \,, & \text{if } c \text{ is stuff class} \\ \mathrm{PQ}_c \,, & \text{otherwise.} \end{cases}$$

Furthermore, we denote by $\mathrm{PQ}^{\dagger}$ the final version of the proposed panoptic metric, which averages $\mathrm{PQ}^{\dagger}_c$ over all classes, *i.e*.

$$\mathrm{PQ}^{\dagger} = \frac{1}{N_{\text{classes}}} \sum_{c \in \mathcal{Y}} \mathrm{PQ}^{\dagger}_c \,.$$

Similarly to PQ, the proposed metric is bounded in $[0, 1]$ and implicitly regards a stuff segment of an image as a single instance. However, we do not require the prediction of stuff classes to have IoU>0.5 with the ground truth.

5. Experimental Results

We assess the benefits of our proposed network architecture on multiple street-level image datasets, namely Cityscapes [10], Mapillary Vistas [39] and the Indian Driving Dataset (IDD) [51]. All experiments were designed to provide a fair comparison between baseline reference models and our proposed architecture design choices. To increase transparency of our proposed design contributions, we deliberately leave out model extensions like path aggregation network extensions [7, 37], deformable convolutions [13] or Cascade R-CNN [4]. We do not apply test time data augmentation (multi-scale testing or horizontal flipping) or explicit use of model ensembles, *etc*., as we assume that such bells and whistles approximately equally increase recognition performances for all methods. All models were only pre-trained on ImageNet [47]. We use the following terminology in the remainder of this section: *Ours Independent* refers to fused, but individually trained models (Fig. 1 left) each following the proposed design, and *Ours Combined* refers to the unified architecture in Fig. 1 (right).

Model and Training Hyperparameters. Unless otherwise noted, we take all the hyperparameters of the instance segmentation branch from [19]. These hyperparameters are shared by all the models we evaluate in our experiments, and are exhaustively listed in [42]. We initialize our backbone model with weights extracted from PyTorch's ImageNet-pretrained ResNet-50 despite using a different activation function, motivated by findings in our prior work [46]. We train all our networks with SGD, using a fixed schedule of 48k iterations and learning rate 10^{-2}, decreasing the learning rate by a factor 10 after 36k and 44k iterations. At the beginning of training we perform a warm-up phase where the learning rate is linearly increased from $\frac{1}{3} \cdot 10^{-2}$ to 10^{-2} in 200 iterations.[2] During training the networks receive full images as input, randomly flipped in the horizontal direction, and scaled such that their shortest side measures $\lfloor 1024 \cdot t \rfloor$ pixels, where t is randomly sampled from $[0.5, 2.0]$. Training is performed on batches of 8 images using a computing node equipped with 8 Nvidia V100 GPUs. At test time, images are scaled such that their shortest size measures 1024 pixels (preserving aspect ratio).

5.1. Cityscapes

Cityscapes [10] is a street-level driving dataset with images from 50 central-European cities. All images were recorded with a single camera type, image resolution of 1024×2048, and during comparable weather and lighting conditions. It has a total of 5,000 pixel-specifically annotated images (2,975/500/1,525 for training, validation and test, respectively), and additionally provides 19,998 images

[2] Note that the warm-up phase is not strictly needed for convergence. Instead, we adopt it for compatibility with [19].

Figure 4: Qualitative results obtained by our proposed combined architectures. Top row: Cityscapes. Middle row: IDD. Bottom row: Vistas. Best viewed in color and with digital zoom.

forming the *coarse extra* set, where only coarse annotations per image are available (which we have not used in our experiments). Images are annotated into 19 object classes (11 stuff and 8 instance-specific).

For *Ours Independent*, we trained each recognition model independently, using the hyperparameter settings described above (again, each with a ResNet-50+FPN backbone). For the semantic segmentation model, we obtain a baseline segmentation result of 73.8%(mean Intersection-over-Union [15]), which is comparable to 75.2% reported in [28] (using a DenseNet-169 backbone), 73.6% using DeepLab2 in combination with a ResNet-101 backbone as reported in [45], or 74.6% with a ResNet-152 in [53]. The instance-segmentation AP_M (mean average precision on masks) results of our single model baseline are 31.9%, which is slightly above the reported baseline score in Mask R-CNN [19] (31.5% w/o COCO [34] pre-training).

Fusing the results of our individually trained models (*Ours Independent*) delivers PQ $= 59.8\%$, $PQ_{St} = 64.5\%$, $PQ_{Th} = 64.5\%$ and $PQ^{\dagger} = 59.0\%$. We furthermore provide results of *Ours Combined* in Tab. 1, performing equally well on PQ and $PQ^{\dagger}$. This is remarkable, given the significantly reduced number of model parameters (see discussion in Section 5.4) and when assuming that the fusion of individually trained models could lead to an ensemble effect (often deliberately used to improve test results, at the cost of increased computational complexity).

In addition, we show results of jointly trained networks from independent, concurrently appearing works [14, 23, 29, 54, 55], with focus on comparability of network architectures and data used for pre-training. In Tab. 1 we abbreviate the network backbones as R50, R101 or X71 for ResNet50, ResNet101 or Xception Net71, respectively, and provide datasets used for pre-training (**I** = ImageNet and **C** = COCO). All our proposed variants outperform the direct competitors by a considerable margin, *i.e.*, our baseline models as well as jointly trained architectures are better.

The last entry in Tab. 1 shows results for another variant of our network where we deactivated freezing of all parameters and dropped weight decay on the batch normalization parameters (keeping the rest as described above). We can see that this gives another boost in terms of PQ. Finally, the top row in Fig. 4 shows some qualitative seamless segmentation results obtained with our architecture.

5.2. Indian Driving Dataset (IDD)

IDD [51] was introduced for testing perception algorithm performance in India. It comprises 10,003 images from 182 driving sequences, divided in 6,993/981/2,029 images for training, validation and test, respectively. Images are either of 720p or 1080p resolution and were obtained from a front-facing camera mounted on a car roof. The dataset is annotated into 26 classes (17 stuff and 9 instance-specific), and we report results for *level 3* labels.

Method	Body	Data	Cityscapes PQ	PQ$_{St}$	PQ$_{Th}$	PQ†	AP$_M$	IoU	Vistas PQ	PQ$_{St}$	PQ$_{Th}$	PQ†	AP$_M$	IoU
de Geus *et al*. [14]	R50	**I**	-	-	-	-	-	-	17.6	27.5	10.0	-	-	34.7
Supervised in [30]	R101	**I**	47.3	52.9	39.6	-	24.3	71.6	-	-	-	-	-	-
FPN-Panoptic [23]	R50	**I**	57.7	62.2	51.6	-	32.0	75.0	-	-	-	-	-	-
TASCNet [29]	R50	**I+C**	59.2	61.5	56.0	-	37.6	77.8	32.6	34.4	31.1	-	18.5	-
UPSNet [54]	R50	**I**	59.3	62.7	54.6	-	33.3	75.2	-	-	-	-	-	-
DeeperLab [55]	X71	**I**	56.3	-	-	-	-	-	32.0	-	-	-	-	55.3
Ours Combined	R50	**I**	59.8	63.6	54.6	59.0	33.0	76.2	36.2	40.0	33.6	37.5	16.5	45.8
Ours Combined no freeze|decay BN	R50	**I**	60.2	63.6	55.6	59.6	33.3	74.9	35.8	39.8	33.0	37.2	16.2	45.6

Table 1: Comparison of validation set results on Cityscapes and Vistas with related works. Used network bodies include R101, R50 and X71 for ResNet-101, ResNet-50 and Xception-71, respectively. *Data* indicates datasets used for pre-training where **I** = ImageNet and **C** = COCO. All results in [%].

The recognition models for *Ours Independent* obtained segmentation and instance segmentation results of IoU $=$ 67.2% and AP$_M$ $=$ 29.8%, respectively. The numbers reported as baselines in [51] for semantic segmentation are 55.4% using ERFNet [44] and 66.6% for dilated residual nets [58] and again Mask R-CNN for instance-specific segmentation on a ResNet-101 body yielding AP$_M$ $=$ 26.8%. Those numbers supposedly belong to the test set, while no numbers are reported for validation. Moreover, *Ours Independent* yields PQ $=$ 47.2%, PQ$_{St}$ $=$ 46.6%, PQ$_{Th}$ $=$ 48.3% and PQ† $=$ 48.8%. For *Ours Combined* we obtain PQ $=$ 46.9%, PQ$_{St}$ $=$ 45.9%, PQ$_{Th}$ $=$ 48.7%, PQ† $=$ 48.5%, AP$_M$ $=$ 29.8% and IOU $=$ 67.5%. In the key metrics PQ and PQ† the results differ by ≤ 0.3 points, and we again stress that the numbers for *Ours Combined* are provided from network architectures with significantly less parameters.

The middle row in Fig. 4 shows seamless segmentation results obtained by our combined architecture.

5.3. Mapillary Vistas

Mapillary Vistas [39] is one of the richest, publicly available street-level image datasets today. It comprises 25k high-resolution (on average 8.6 MPixels) images, split into sets of 18k/2k/5k images for training, validation and test, respectively. We only used the training set during model training while evaluating on the validation set. Vistas shows street-level images from all over the world, with images captured from driving cars as well as pedestrians taken them on a sidewalk. It also has large variability in terms of weather, lighting, capture time during day and season, sensor type, *etc*., making it a very challenging road scene segmentation benchmark. Accounting for this, we modify some of our model's hyper-parameters and training schedule as follows: we use anchors with aspect ratios in $\{0.2, 0.5, 1, 2, 5\}$ and area $(2 \times D)^2$, where D is the FPN level's downsampling factor; we train on images with shortest side scaled to $\lfloor 1920 \cdot t \rfloor$, where t is randomly sampled from $[0.8, 1.25]$; we train for a total of 192k iterations, decreasing the learning rate after 144k and 176k iterations.

Scores for both, *Ours Combined* and its slightly modified variant discussed at the end of Section 5.1 are given in Tab. 1. We obtain $+4.2\%$ and $+3.6\%$ PQ score over DeeperLab [55] and TASCNet [29], respectively. More details are given in [42]. We also show seamless scene segmentation results in the bottom row of Fig. 4.

5.4. Computational Aspects

Here, we discuss computational aspects when comparing two individually trained recognition models against our combined model architecture. When fused, the two task-specific models have $\approx 78.06M$ parameters, which are $\approx 51.8\%$ more than our combined architecture ($\approx 51.43M$). The majority of saved parameters belong to the backbone. The amount of computation is similarly reduced, *i.e*. the combined, independently trained models require $\approx 50.4\%$ more FLOPs due to two inference steps per test image. In absolute terms, the individual models require ≈ 0.864 TFLOP while our combined architectures requires ≈ 0.514 TFLOP on 1024×2048 image resolution, respectively.

6. Conclusions

In this work we have introduced a novel CNN architecture for producing *seamless scene segmentation results*, *i.e*. jointly acting semantic segmentation and instance segmentation modules operating on top of a single network backbone. We depart from the prevailing approach of training individual recognition models, and instead introduce a multi-task architecture that benefits from interleaving network components as well as a novel segmentation module. Moreover, we revisit the panoptic metric used to assess combined segmentation and detection results, and propose a relaxed alternative for handling stuff segments. Our findings include that we can generate state-of-the-art recognition results that are significantly more efficient in terms of computational effort and model sizes, when compared to combined, individual models.

References

[1] Anurag Arnab and Philip H.S. Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *(CVPR)*, 2017. 2

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015. 2

[3] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *(ECCV)*, pages 44–57. 2008. 2

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *(CVPR)*, June 2018. 6

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *CoRR*, abs/1606.00915, 2016. 2

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. 4

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018. 6

[8] Liang-Chieh Chen, Maxwell D. Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jonathon Shlens. Searching for efficient multi-scale architectures for dense image prediction. *(NIPS)*, 2018. 2

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *(ECCV)*, September 2018. 2

[10] 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. 2, 6

[11] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun Sun. Instance-sensitive fully convolutional networks. In *(ECCV)*, 2016. 2

[12] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 2

[13] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *(ICCV)*, Oct 2017. 6

[14] Daan de Geus, Panagiotis Meletis, and Gijs Dubbelman. Panoptic segmentation with a joint semantic and instance segmentation network. *CoRR*, abs/1809.02110, 2018. 3, 7, 8

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *(IJCV)*, 88(2):303–338, 2010. 7

[16] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *(CVPR)*, 2014. 2

[17] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *(ECCV)*, pages 297–312, 2014. 2

[18] Zeeshan Hayder, Xuming He, and Mathieu Salzmann. Boundary-aware instance segmentation. In *(CVPR)*, 2017. 2

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *(ICCV)*, 2017. 2, 3, 5, 6, 7

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *CoRR*, abs/1603.05027, 2016. 3

[21] Xuming He and Stephen Gould. An exemplar-based crf for multi-instance object segmentation. In *(CVPR)*, 2014. 2

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

[23] Alexander Kirillov, Ross B. Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. *CoRR*, abs/1901.02446, 2018. 3, 7, 8

[24] Alexander Kirillov, Kaiming He, Ross B. Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. *CoRR*, abs/1801.00868, 2018. 1, 5

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

[26] Peter Kontschieder, Samuel Rota Bulò, Marcello Pelillo, and Horst Bischof. Structured labels in random forests for semantic labelling and object detection. *(PAMI)*, 36, 2014. 2

[27] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *(NIPS)*, 2011. 2

[28] Ivan Kreso, Sinisa Segvic, and Josip Krapac. Ladder-style densenets for semantic segmentation of large natural images. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017. 7

[29] Jie Li, Allan Raventos, Arjun Bhargava, Takaaki Tagawa, and Adrien Gaidon. Learning to fuse things and stuff. *CoRR*, abs/1812.01192, 2018. 3, 7, 8

[30] Qizhu Li, Anurag Arnab, and Philip H.S. Torr. Weakly- and semi-supervised panoptic segmentation. In *(ECCV)*, 2018. 2, 8

[31] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. *CoRR*, abs/1611.07709, 2016. 2

[32] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *(CVPR)*, 2017. 2

[33] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016. 3

[34] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. *CoRR*, abs/1405.0312, 2014. 7

[35] Huanyu Liu, Chao Peng, Changqian Yu, Jingbo Wang, Xu Liu, Gang Yu, and Wei Jiang. An end-to-end network for panoptic segmentation. *CoRR*, abs/1903.05027, 2019. 3

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

[37] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia.

Path aggregation network for instance segmentation. In *(CVPR)*, June 2018. 2, 6

[38] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *(CVPR)*, pages 3431–3440, 2015. 2

[39] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *(ICCV)*, October 2017. 2, 6, 8

[40] Pedro H. O. Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. 2015. 2

[41] Pedro H. O. Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *(ECCV)*, 2016. 2

[42] Lorenzo Porzi, Samuel Rota Bulò, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. *CoRR*, 2019. 5, 6, 8

[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *(NIPS)*, 2015. 2

[44] E. Romera, J. M. lvarez, L. M. Bergasa, and R. Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):263–272, 2018. 8

[45] S. Rota Bulò, G. Neuhold, and P. Kontschieder. Loss max-pooling for semantic image segmentation. In *(CVPR)*, July 2017. 2, 5, 7

[46] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of DNNs. In *(CVPR)*, 2018. 3, 6

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karphathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *(IJCV)*, 2015. 6

[48] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *(IJCV)*, 81(1):2–23, 2007. 2

[49] J. Tighe, M. Niethammer, and S. Lazebnik. Scene parsing with object instances and occlusion ordering. In *(CVPR)*, 2014. 1, 2

[50] Z. Tu, X. Chen, A.L. Yuille, and S.-C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *(IJCV)*, 2005. 1, 2

[51] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and C V Jawahar. Indian driving dataset (IDD): A dataset for exploring problems of autonomous navigation in unconstrained environments. In *(WACV)*, 2019. 2, 6, 7, 8

[52] Yuxin Wu and Kaiming He. Group normalization. In *(ECCV)*, September 2018. 3

[53] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. High-performance semantic segmentation using very deep fully convolutional networks. *CoRR*, abs/1604.04339, 2016. 2, 7

[54] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. *CoRR*, abs/1901.03784, 2019. 3, 7, 8

[55] 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. 3, 7, 8

[56] Jian Yao, Sanja Fidler, and Raquel Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *(CVPR)*, 2012. 1, 2

[57] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *Int. Conf. on Learning Representations (ICLR)*, 2016. 2

[58] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *(CVPR)*, 2017. 2, 8

[59] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *CoRR*, abs/1612.01105, 2016. 2