2019 IEEE Intelligent Transportation Systems Conference (ITSC)
Auckland, NZ, October 27-30, 2019

# Multi-task Network for Panoptic Segmentation in Automated Driving

Andra Petrovai and Sergiu Nedevschi

*Abstract*— **In this paper, we tackle the newly introduced panoptic segmentation task. Panoptic segmentation unifies semantic and instance segmentation and leverages the capabilities of these complementary tasks by providing pixel and instance level classification. Current state-of-the-art approaches employ either separate networks for each task or a single network for both task and post processing heuristics fuse the outputs into the final panoptic segmentation. Instead, our approach solves all three tasks including panoptic segmentation with an end-to-end learnable fully convolutional neural network. We build upon the Mask R-CNN framework with a shared backbone and individual network heads for each task. Our semantic segmentation head uses multi-scale information from the Feature Pyramid Network, while the panoptic head learns to fuse the semantic segmentation logits with variable number of instance segmentation logits. Moreover, the panoptic head refines the outputs of the network, improving the semantic segmentation results. Experimental results on the challenging Cityscapes dataset demonstrate that the proposed solution achieves significant improvements for both panoptic segmentation and semantic segmentation.**

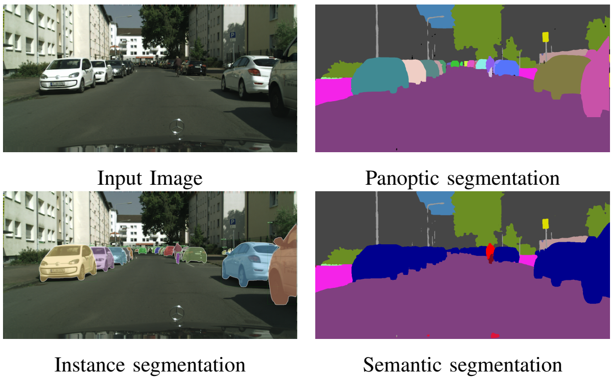
Input Image
Panoptic segmentation
Instance segmentation
Semantic segmentation

**Fig. 1:** We propose a unified network architecture for instance segmentation, semantic segmentation and panoptic segmentation. Instance and semantic segmentation are fused into the panoptic output providing pixel-level class information and instance IDs for objects.

## I. INTRODUCTION

Standing at the intersection between semantic and instance segmentation, panoptic segmentation enables a complete scene understanding at pixel level and at instance level for dynamic elements of the scene. Applications such as automated driving could benefit from the rich information provided by panoptic segmentation, which can enhance a sensor fusion based environment perception. Lately, the research community has given attention to both semantic and instance segmentation tasks and proposed solutions using deep convolutional neural networks (CNN). Each task has its own architecture particularities. In the case of semantic segmentation, Fully Convolutional Neural Networks (FCN) [28] [5] [43] [39] extract features using dilated residual blocks in order to preserve a higher output resolution. On the other hand, instance segmentation state-of-the-art results have been achieved by the Mask R-CNN framework [12] where a Feature Pyramid Network [25] provides a multi-scale feature representation for object detection and instance segmentation.

Semantic segmentation partitions an image into meaningful segments, which share a common representation. Dense pixel prediction classifies each pixel into one of a few classes. Classes can be categorized as *stuff* or background, representing uncountable elements in the scene that usually have repetitive textures, but not a specific size or shape such as sky, vegetation, buildings, road. Besides *stuff*, there are the *thing* classes or the foreground. *Things* are countable objects that have a well-defined shape. One of the advantage of semantic segmentation is that it performs well in *stuff* classification, since these classes are more structured. Moreover, it provides good delimitation between *stuff* and *thing* pixels such as road from car or sidewalk from pedestrian. On the other hand, semantic segmentation cannot distinguish between objects of the same class and since classification is achieved at pixel level, sometimes fails in classifying object classes belonging to the same category.

Instance segmentation as seen in Figure 1 predicts a semantic class and an instance label for each *thing* pixel in the image. It provides a mask for each object, masks which could overlap since classification is performed at instance level. Objects are detected and classified as a whole, therefore the semantic and instance class are propagated to each pixel in the instance mask.

Kirillov et al. [17] introduces the panoptic segmentation task as a unified semantic and instance representation. Panoptic segmentation is challenging since each task has led to different architectural design choices tailored to achieve state-of-the-art results on tasks specific benchmarks. Moreover, using separate networks brings high computational costs and high memory footprint. And finally, fusing the outputs of semantic and instance segmentation is not trivial, due to overlaps between instance masks or between instance masks and background.

Considering that these tasks are complementary, we want to leverage the capabilities of each task and propose unifying semantic and instance segmentation under one architecture.

*Andra Petrovai and Sergiu Nedevschi are with the Department of Computer Science, Technical University of Cluj-Napoca, Cluj-Napoca, Romania , andra.petrovai@cs.utcluj.ro, sergiu.nedevschi@cs.utcluj.ro

978-1-5386-7024-8/19/$31.00 ©2019 IEEE

We solve the output fusion by designing an end-to-end trainable network that can learn object occlusions and scene depth ordering. We employ a ResNet [13] architecture for feature extraction and a Feature Pyramid Network (FPN) for multi-scale feature representation. On top of the shared backbone we introduce individual network heads for the 4 tasks: Faster R-CNN object detection and classification [10], Mask R-CNN instance segmentation, [12], semantic segmentation and panoptic segmentation. The semantic segmentation head learns multi-resolution image features at 4 scales, following the Feature Pyramid Network (FPN). We design a lightweight decoder-style semantic segmentation head which encodes context information using Pyramid Pooling Module (PSP) [43] and recovers object boundaries and details using low-level features [5] and upsampling operations. The panoptic segmentation head performs semantic and instance level recognition by pixel-level classification. The semantic segmentation representations belonging to *thing* classes are enhanced by introducing corresponding features from the instance segmentation head guided by the 2D bounding box detection. We consider *stuff* classes as well as the instance labels as the panoptic segmentation classes, which could vary with each image. We perform experiments on the Cityscapes [7] dataset, in which our model achieves top performing results, with 75.4% mIoU and 57.3% PQ.

## II. RELATED WORK

**Semantic segmentation:** Fully Convolutional Neural Networks (FCN) [28] achieve state-of-the-art results for semantic segmentation and dominate the benchmarks. Coarse grained representations provide better localization and stronger context information, while high resolution features provide details of finer scales such as shape and boundaries. As both aspects are equally important for semantic segmentation, several mechanism have been proposed to achieve both good localization and pixel level classification. Dilated convolutions [4][5][6] with various dilation rates in the last residual blocks enlarge the field of view of filters by capturing long-range information without decimating the resolution. An alternative to dilated convolutions represent deformable convolutions [9], where the network learns filters with adaptive receptive fields. Important advances in semantic segmentation have been achieved with Pyramid Networks. Spatial Pyramid Networks (PSPNet) [43] employ pooling pyramid after the last dilated convolutional layer and exploit global context information by fusing pooled features at four scales. Atrous Spatial Pyamid Pooling (ASPP) [4], [5] capture multi-scale representations with parallel filters at various sampling rates. Dilated convolutional neural networks have a high memory footprint since dilated filters generate high resolution features. Much attention has been directed to another architectural design, the encoder-decoder [31][36][1][37][24][39][19], where the encoder, a usually deeper and narrower network learns contextual information and the decoder, a lightweight network recovers the resolution loss and shape of the segments. In [15][42][8] the authors extend the Mask R-CNN framework and add a segmentation head on top of the Feature Pyramid Network to achieve comparable results with single-task networks for dense pixel prediction. Our network follows the encoder-decoder architecture for semantic segmentation, in which the shared backbone encodes feature representations at different scales and the Feature Pyramid Network and the segmentation head recovers shapes and spatial information.

**Instance segmentation:** Instance segmentation approaches usually follow two directions and are either region-based or semantic segmentation based. Region-based instance segmentation generate candidate instance regions using state-of-the-art detectors such as Faster R-CNN [10]. Other methods directly propose candidate masks [3][33][34]. Mask R-CNN [12] has demonstrated outstanding performance on benchmarks [7][26] and the COCO detection challenge [17]. Based on a shared backbone, the network can be jointly trained for 2D bounding box detection and classification and also instance segmentation. Variants of Mask R-CNN include Cascade R-CNN [2], Non-local networks [40]. Other types of instance segmentation approaches start from the semantic segmentation and perform clustering of pixels to obtain instances [18] [27][29] [14]. Other works such as PersonLab [30], CornerNet [20] introduce keypoint guided instance segmentation, while Box2Pix [38] predicts pixel-wise offset vectors from the object centers as well as semantic segmentation and bounding boxes. Region-based methods achieve top-performing results on detection benchmarks, therefore we choose as baseline the Mask R-CNN model for object detection and instance segmentation and propose an improved segmentation and panoptic head.

**Panoptic segmentation:** Panoptic segmentation unifies semantic and instance segmentation into one output in which each pixel receives a semantic class and each *thing* pixel receives an instance label. Kirillov *et al.* [17] introduce a simple and robust baseline for both tasks by extending the Mask R-CNN with a lightweight dense prediction branch and a simple post-processing step solves instance overlaps to obtain the final panoptic format. In [8] the authors design an improved semantic segmentation head based on Atrous Spatial Pyramids on top of Mask R-CNN and introduce a novel output fusion of semantic segmentation and instance segmentation outputs based on instance label propagation following the semantic path at category level. UPSNet [42] provides a unified framework on top of Mask R-CNN and and proposes a parameter-free panoptic segmentation head that leverages logits from the segmentation and instance head. A weakly supervised model was developed by Li *et al.* [22], where *thing* classes are supervised with bounding boxes and *stuff* classes with image tags. Li *et al.* [23] introduce attention modules at proposal and mask level to enhance background features in the background branch. A Non Maxima Suppresion (NMS)-like procedure solves overlaps betwen *thing* and *stuff* and generates the panoptic segmentation map. TASCNet [21] ensures *stuff* and *thing* mask alignment using a cross-task consistency loss, facilitating semantic and instance fusion. Our main contribution in this area represents the design of a new semantic segmentation head on top of the

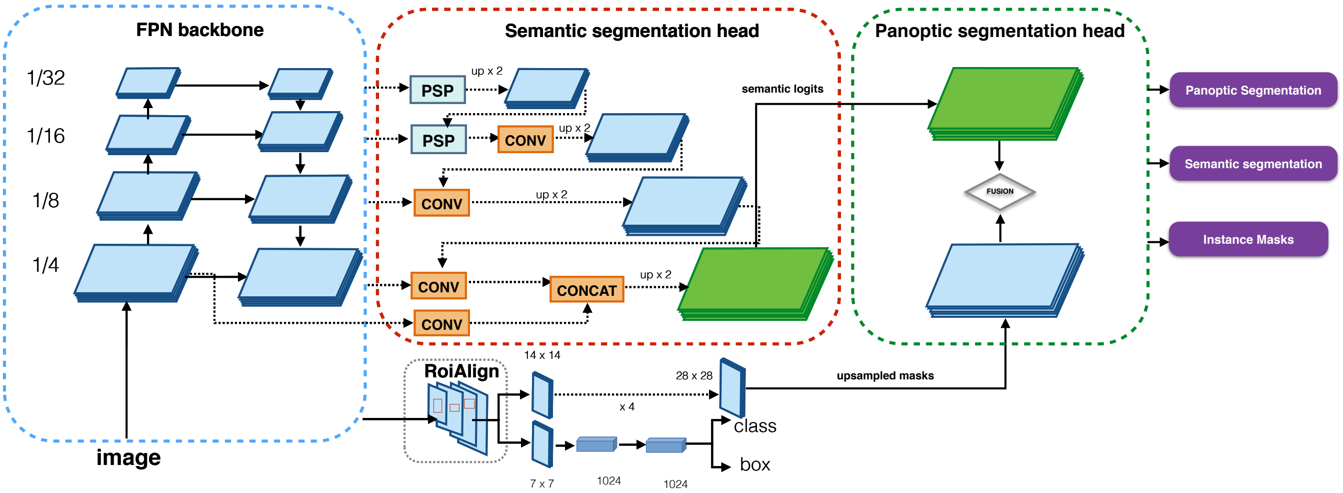

**Fig. 2:** A shared ResNet-FPN network performs 4 tasks: object detection and classification, instance segmentation, semantic segmentation and panoptic segmentation. PSP denotes the Pyramid Pooling Module. The semantic segmentation head is described in detail in Section III-A. The panoptic head fuses the semantic segmentation logits and upsampled masks.

Feature Pyramid Network and the design of an end-to-end trainable unified network that outputs panoptic segmentation and avoids hand crafted post-processing steps.

## III. Panoptic Segmentation Network

We propose a unified network for instance, semantic and panoptic segmentation. We build upon the strong baseline of Mask R-CNN and design a decoder-style semantic segmentation head which classifies each pixel in the image of both *thing* and *stuff* categories. The panoptic segmentation head connects the semantic and instance branch and enables supervised guided fusion of the two outputs within the network. In the following section, we provide details about the network architecture and implementation details. The network architecture overview can be seen in Figure 2.

### A. Model Architecture

**Backbone:** The Mask R-CNN framework employs a shared convolutional network backbone based on ResNet [13] and Feature Pyramid Network (FPN) [25] for feature extraction. FPN encodes multi-scale representations from $1/32$ to $1/4$ and is built in a top-down manner by upsampling low resolution features and merging them with higher level features via lateral connections.

**Instance Segmentation Head:** We adopt the detection and instance segmentation head from Mask R-CNN. In the first stage, the detector proposes object candidates, while in the second stage, candidate bounding boxes are regressed and classified and a binary mask is predicted for each *thing* object. The mask logits are further processed by our panoptic head for feature enhancement at object level.

**Semantic segmentation Head:** The semantic segmentation head predicts per pixel classification for both *thing* and *stuff* segments. We start with the baseline design of the FPN with a 4-scale pyramid at $1/32$, $1/16$, $1/8$ and $1/4$ from the original scale. We follow the original implementation with 256 output feature maps at each FPN level. Each level of the pyramid is augmented with a set of operations specific to its scale: lower levels capture context information, while higher levels highlight detailed features. We employ the Pyramid Pooling Module (PSP) [43] for capturing long-range dependencies at scale $1/32$ and $1/16$ as seen in 3. The PSP module applies average pooling operations with different pooling rates. The output of the pooling operations have sizes $[1 \times 1]$, $[2 \times 2]$, $[3 \times 3]$ and $[6 \times 6]$. Next, the output is scaled to the size of the input $1/32$ or $1/16$ respectively. The PSP output from scale $1/32$ is upsampled two times and concatenated with the PSP output from scale $1/16$. Next, we apply feature dimension reduction with a $1 \times 1$ convolution to 128 channels and another [ $3 \times 3$, 128] convolution, Batch Normalization, ReLU follows. For upsampling, we perform bilinear interpolation followed by a [$3 \times 3$, 128] convolution. This upsampling stage is repeated at scale $1/8$. The resulted feature maps which are $1/4$ smaller than the original image scale are concatenated with low-level features from corresponding feature representations in the ResNet backbone. Chen *et al.* [6] demonstrates that using low-level features in the decoder leads to better perfomance. We follow this design choice and reduce the channels of the low-level features to 32 using a [$1 \times 1$] convolution. The concatenated features are passed through 2 convolution operations with [$1 \times 1$, 256] filters and [$3 \times 3$, 256] filters. Finally, a $1 \times 1$ convolution generates the final class predictions. Different from the baseline [15], where the semantic head predicts an *other* class for all objects, pixels are classified into all *stuff* and *thing* classes. To obtain the final predictions, we associate per-pixel softmax and the semantic segmentation head is trained to minimize the bootstrapped cross-entropy loss [35].

**Panoptic Segmentation Head:** The panoptic segmentation head predicts per-pixel classification for *stuff* classes and instance-level classification for *thing* classes. The panoptic outputs are generated by merging the semantic segmentation

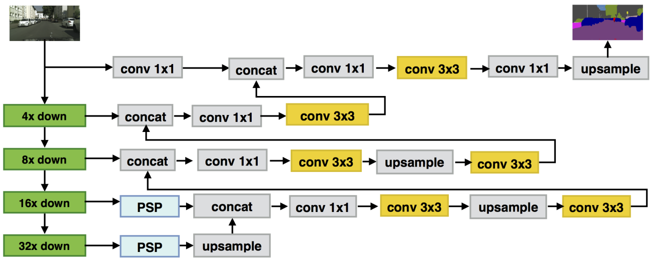

**Fig. 3:** Semantic segmentation architecture. The PSP Module captures context information at the lower scales of the FPN pyramid. Multiple upsampling stages interleaved with convolutions follow at each level. The semantic segmentation output is $1/2$ from the original image resolution.

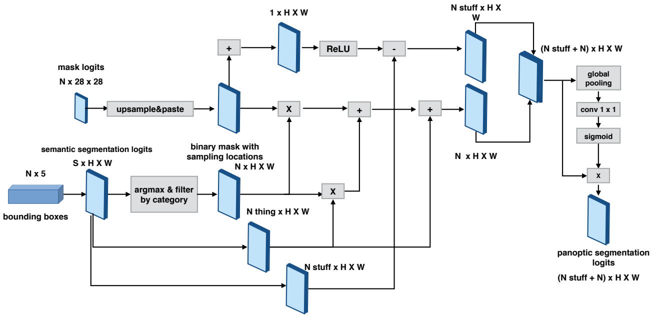

**Fig. 4:** Panoptic head. Panoptic masks logits are constructed by adding the instance masks logits with the semantic logits. Sampling locations in the object bounding box are determined by prominent features from the semantic logits at category level.

logits and the instance segmentation logits predicted by the two corresponding heads. Figure 4 illustrates the fusion process.

Let us denote $S$ the semantic segmentation logits with $S \in \mathbb{R}^{C_{seg} \times H \times W}$. $C_{seg}$ represent the number of the semantic segmentation classes, while $H$ and $W$ are the size of the semantic segmentation logits, in our case $\frac{1}{2}$ of the original image scale. The segmentation classes $C_{seg}$ contain both *thing* and *stuff* classes along a *void* class for the ignored pixels in training: $C_{seg} = C_{stuff} + C_{thing} + 1$.

The instance segmentation head produces a logit tensor $T$ with $T \in \mathbb{R}^{C_{inst} \times C_{thing} \times H' \times W'}$. In our case, we use a Mask R-CNN type of instance segmentation head with $H' = W' = 28$ and $C_{inst}$ are the number of instances in the image.

The panoptic segmentation head predicts the semantic class for *stuff* classes and the instance IDs for *thing* classes, following the design choice in [42]. The panoptic logits $P$ have the form $P \in \mathbb{R}^{C_{stuff} + C_{inst} \times H \times W}$ where $C_{stuff}$ is the number of semantic classes for the *stuff* pixels and $C_{inst}$ represents the number of instances in the image and has a different value for each image.

PANOPTIC STUFF: In order to build the panoptic segmentation logits, we employ the output of the semantic segmentation head and the instance segmentation head. The first $C_{stuff}$ channels in the panoptic output represent background features. To refine the semantic segmentation channels corresponding to *stuff* classes and decrease the prominence of *thing* classes we introduce contextual cues guided by the instance masks. We construct a mask tensor with all the instance masks from the image by taking the output of the instance segmentation head, which is $C_{inst}$ masks of size $28 \times 28$ and upsampling the masks to their scale in the original image and paste them to their corresponding location in the $H \times W$ map. After that, we apply a ReLU activation to enhance instance masks features. Finally, from each *stuff* channel of the semantic segmentation logits we subtract the activated upsampled instance map.

PANOPTIC INSTANCES: The panoptic instance logits have the same size as the number of instances in the image $C_{inst}$. On each channel we have the mask of an instance obtained from the combined output logits of the semantic and instance segmentation head. The instance segmentation masks are sampled over a low-resolution $28 \times 28$ grid and after upsampling, the masks display coarser boundaries, especially for very large objects. On the other hand, the semantic segmentation masks are more precise, therefore we create the panoptic instance outputs from the instance masks sampled by the semantic segmentation logits. We construct the panoptic instance logits in the following way: for each instance $i$ detected by the instance segmentation head, we determine its semantic $classID$ and its semantic category $cat$ (vehicle, person, two-wheeled). Next, we sample the most prominent features from the semantic segmentation logits corresponding to its category in the object bounding box to obtain $B_i$, *i.e.*, we take *max* along channel dimension from $S$ where *argmax* $\in C_{cat}$ and $C_{cat}$ represent the classes belonging to a category. Using the same sample locations, we take the corresponding mask logits $U_i$ from the upsampled mask at channel class ID $T_{classID}$, where $U_i \in \mathbb{R}^{1 \times H \times W}$. The final panoptic output for instance $i$ with $i \in C_{inst}$ is $P_i = B_i + U_i$. We choose to take the activated pixels from the semantic segmentation logits at category level and not at class level, since semantic segmentation sometimes fails in classifying pixels of different *thing* classes (e.g. bus, truck) but belonging to the same category (e.g. vehicle).

Panoptic logits belonging to both *stuff* and *thing* classes are then passed through an attention module. We want to model the dependencies between instance masks and reweight background channels in order to enhance relevant classes and reduce the weight of the others. First operation in the attention module is the global average pooling which captures global context for each channel. Then, a $[1 \times 1]$ convolution models the dependencies between channels. Finally, a sigmoid activation is applied and the panoptic logits are scaled by the attention vector.

## B. Implementation Details

**Training setup:** We implement our model in PyTorch [32] and train it on a system with 4 GPUs. We train the model end-to-end with a single optimization step. We train and test our implmentation on the Cityscapes dataset. Each mini-batch has 1 image/GPU. The images are augmented by random horizontal flipping and scaling with the shorter edge randomly sampled from $[800, 1024]$. We choose stochastic gradient descent (SGD) as optimizer with momentum 0.9, weight decay $1e-4$ and a "poly" learning rate policy starting from a base learning rate of $1e-2$. Our model converges in 12k iterations. Layer normalization is done with Group Normalization [41] layers, which are invariant to batch size. In all our experiments, we initialize the network from available pretrained weights for bounding box detection and classification and instance mask detection [11] on the Microsoft COCO dataset [26]. We note that the semantic segmentation head and the panoptic segmentation head are not initialized, therefore are trained from scratch. Our final loss is computed as follows: $L = L_{cls} + L_{box} + L_{mask} + L_{seg} + L_{pan}$.

**Panoptic Segmentation Training:** The panoptic segmentation logits during training will be $P \in \mathbb{R}^{C_{stuff} + C_{inst} \times H \times W}$ , where $C_{inst}$ are the number of ground truth masks, $C_{stuff}$ is fixed during both training and inference and $H$ and $W$ are half of the original image scale. The panoptic logits of $thing$ objects are built as follows: each ground truth mask is associated to a predicted mask with the highest bounding box intersection over union. The class of the predicted mask is considered the class of the ground truth. The order of the panoptic logits for instance channels is the same as the order used to construct the panoptic segmentation ground truth. The obtain the final panoptic predictions we apply softmax over the panoptic logits and minimize the bootstrapped cross entropy loss during training.

**Panoptic Segmentation Inference:** During inference, $C_{inst}$ will be the number of instances in the image after a few filtering operations on the bounding box and masks predictions from the bound box detection and instance mask heads. First, class-agnostic non-maxima supression with a IoU of 0.5 is applied on the predicted bounding boxes to resolve overlaps. Next, we sort the remaining boxes and filter out the ones whose confidence score is lower than 0.6. Then masks with a large class-wise overlap are removed. If the non-overlapping number of pixels over the total number of pixels is lower than 0.5, the mask is discarded. Finally, small *stuff* areas in the image are that have an area lower than a threshold is set to void. The threshold used in our experiments is 100. Softmax over the panoptic logits is applied for obtaining per-pixel confidence scores. If the maximum belongs to the one of the $C^{i}_{stuff}$ channels then the pixels receives the semantic class $C^{i}_{stuff}$, while if the maximum belongs to one of the $C_{inst}$ classes, then we have an instance mask with instance ID equal to $C_{inst}$ and the semantic class equal to the class of the instance mask used to construct that panoptic output channel.

## IV. EXPERIMENTS

In this section we provide experimental results on the Cityscapes dataset and we discuss design choices and their influence on performance.

| Method | AP | mIoU | PQ |
|---|---|---|---|
| Mask-RCNN [12] | 36.4 | - | - |
| PSPNet [43] | - | 71.7 | - |
| Unified baseline [16] | 37.0 | 71.6 | - |
| + PSP + improved segmentation head | 37.8 | 73.3 | - |
| + baseline panoptic head | 38.0 | 75.1 | 56.7 |
| + panoptic attention | 38.1 | 75.3 | 56.9 |
| + panoptic background refinement | 38.3 | 75.4 | 57.3 |

**TABLE I:** Ablation study. Instance, semantic segmentation and panoptic segmentation results on the Cityscapes *validation* set. All methods use a ResNet50 backbone. The mIoU is computed from the panoptic head if applicable.

### A. Experimental Setup

**Cityscapes** Cityscapes [7] is a dataset of 5000 urban scene images with pixel-level and instance-level annotations. The dataset is split into 2975 train, 500 val and 1525 test. Annotations are provided for 19 classes, from which 11 *stuff* and 8 *thing* classes.

**Evaluation metrics** We evaluate semantic segmentation using standard mIoU (mean Intersection over Union) metric. For instance segmentation the AP@[.5:.05:.95] (Average Precision over classes and 10 IoU levels from 0.5 to 0.95 with a step size of 0.05 ) is used. For panoptic segmentation we adopt the following metrics: PQ (Panoptic Quality), RQ (Recognition Quality) and SQ (Semantic Quality).

### B. Performance on the Cityscapes dataset

We evaluate our network with a ResNet-FPN backbone. The backbone, the mask head and box head are pretrained on the MS COCO [26] dataset. Our model is trained on the fine train image set only and multi-scale testing is not used in evaluation.

In Table I we present ablation study for the ResNet50-FPN based network. Compared to Mask R-CNN for instance segmentation and PSPNet for semantic segmentation we observe an improvement for both tasks due to multi-task learning. We compare our model to the baseline model based on Mask R-CNN with a semantic segmentation head with two $[3 \times 3]$ convolutions at each FPN level. Compared to the baseline, introducing the PSP at lower FPN levels, employing a different upsampling strategy and low-level features brings a $1.7\%$ improvement for mIoU. By introducing the panoptic head, we can infer semantic segmentation logits which provide a $1.8\%$ increase. The baseline panoptic head constructs the panoptic logits by adding the semantic segmentation logits and the instance segmentation logic [42]. The panoptic attention module, background refinement and category level logits selection improve the baseline with $0.6\%$ in PQ.

In Table II, we analyze the influence of the panoptic head in a multi-task setting. We observe that a lower quality semantic segmentation can be improved at a low computational cost with $3\%$ to $5\%$ by having a good classification and mask detection from the instance segmentation branch.

In Figure 5, we present visual results for instance, semantic and panoptic segmentation. We can observe that the instance masks from the instance segmentation head have

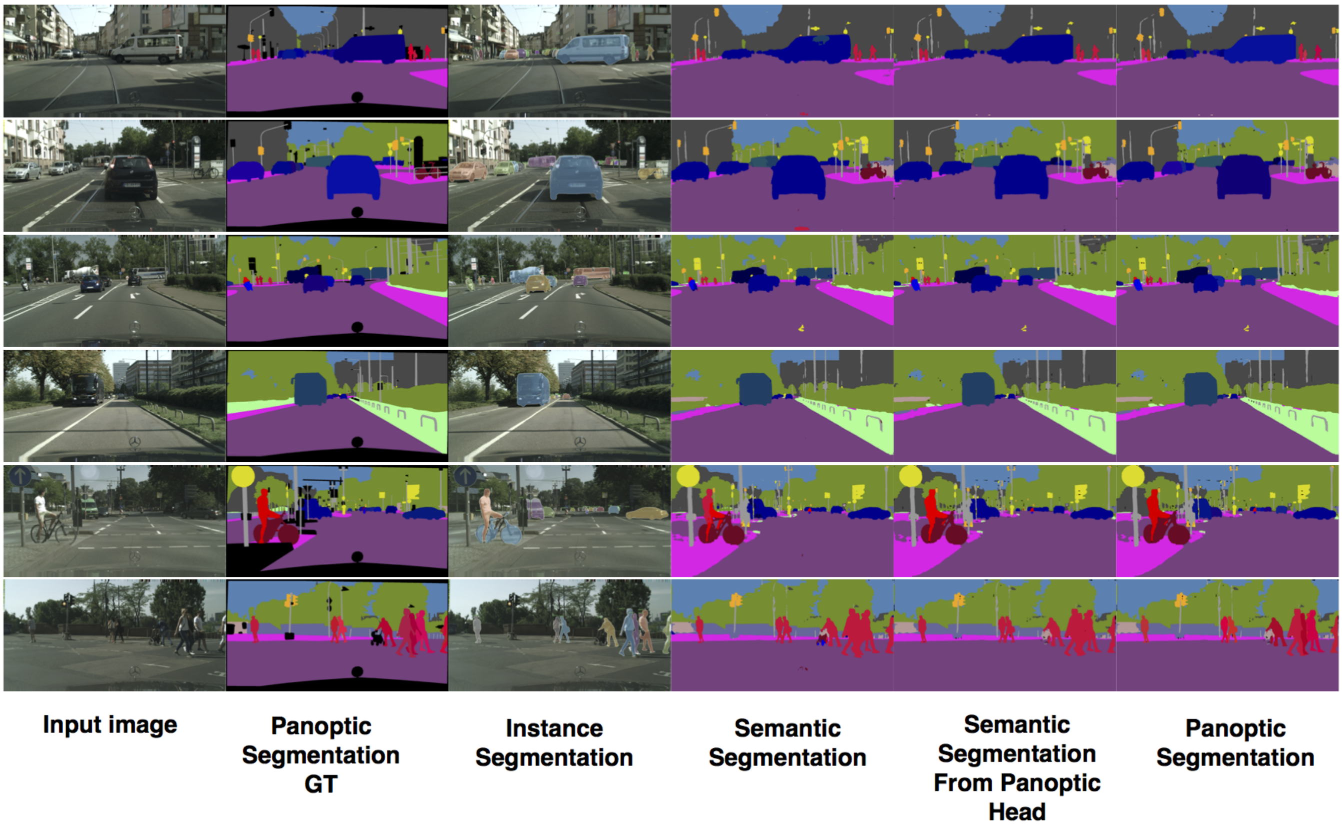

**Fig. 5:** Visual examples of panoptic segmentation. Each instance is colored with a different shade of the same color.

| PQ | SQ | RQ | AP | mIoU seg | mIoU pan |
|---|---|---|---|---|---|
| 44.9 | 75.3 | 56.9 | 35.5 | 60.1 | 64 |
| 48 | 76.7 | 60.4 | 37.2 | 63.6 | 67 |
| 49 | 76.8 | 61.5 | 38.4 | 64.7 | 69.6 |
| 50.8 | 77.6 | 63.6 | 36.6 | 68 | 71.3 |

**TABLE II:** The influence of the panoptic head on performance. The quality of the semantic segmentation increases in some cases with up to 5%.

a coarse boundary especially for large objects, nonetheless, the detection and classification performance is good. On the other hand, semantic segments boundaries are more precise, but classification errors on *thing* classes are determined by similar features for different classes belonging to the same category. By extracting semantic segmentation from the panoptic head, we can see an improvement both for background and foreground classes. Classification at instance level provided by the panoptic segmentation head propagates the semantic information to each pixel in the mask, therefore classification errors at pixel-level are corrected. In the panoptic segmentation image, the instance masks are better aligned to objects and more precise compared the instance masks from the instance segmentation head, due to adding more context and background information. Moreover, background classification is also improved due to the complementary mask attention information. The visual results are validated by the evaluation on the Cityscapes dataset from Table III.

In Table IV we compare our solution with state-of-the-art models. Since panoptic segmentation is a relatively new task there are only a few models evaluated on the Cityscapes dataset. We note that Cityscapes does not have an evaluation server for panoptic segmentation, therefore all our experiments are done on the validation set. Compared to [22], we achieve higher PQ, especially $\text{PQ}^{Th}$, suggesting that their instance segmentation network performs worse than ours. TASCNet and UPSNet employ a ResNet50-FPN backbone for the network pretrained on MS COCO. Mask AP is improved by 1% at the cost of reduced mIoU and PQ. One reason for the lower mIoU and PQ score would be that our semantic segmentation and panoptic segmentation branches are trained from scratch, while theirs is pre-trained on COCO. We achieve comparable results in terms of mIoU and larger AP than PanopticFPN [15], which has a ResNet101 backbone pretrained on ImageNet.

Finally, as far as runtime is concerned, the network runs in 252 ms on a $1024 \times 2048$ Cityscape image on an NVidia GTX 1080Ti GPU.

| Method | road | sidewalk | building | wall | fence | pole | traffic light | traffic sign | vegetation | terrain | sky | person | rider | car | truck | bus | train | motorbike | bike | mIoU |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Semantic Head | 96.0 | 83.5 | 91.3 | 47.0 | 54.4 | 60.5 | 67.4 | 75.6 | 91.8 | 60.3 | 93.9 | 80.8 | 56.3 | 93.9 | 67.9 | 76.8 | 65.2 | 53.8 | 74.4 | 73.3 |
| Panoptic Head | 97.8 | 83.3 | 91.2 | 47.6 | 53.4 | 59.3 | 67.7 | 75.9 | 91.8 | 60.5 | 93.9 | 83.0 | 66.4 | 94.1 | 74.2 | 85.9 | 76.3 | 66.1 | 64.4 | 75.4 |

**TABLE III:** Class mIoU on Cityscapes *validation* set from semantic segmentation head and from panoptic segmentation head.

| **Method** | **PQ** | **SQ** | **RQ** | **PQ$^{Th}$** | **PQ$^{St}$** | **mIoU** | **AP** |
|---|---|---|---|---|---|---|---|
| Li *et al.* [22] | 53.8 | - | - | 42.5 | 62.1 | 71.6 | 28.6 |
| PanopticFPN-ResNet101 [15] | 58.1 | - | - | 52 | 62.5 | 75.7 | 33.0 |
| TASCNet-COCO [21] | 59.2 | - | - | 56 | 61.5 | 77.8 | 37.6 |
| UPSNet - COCO [42] | 60.5 | 80.9 | 73.5 | 57.0 | 63.0 | 77.8 | 37.8 |
| Ours - COCO | 57.3 | 79.1 | 70.7 | 53.9 | 59.7 | 75.6 | 38.3 |

**TABLE IV:** Comparative study on the Cityscapes *validation* set. Unless specified the model is pretrained on ImageNet, otherwise on COCO. All models but PanopticFPN have a ResNet50-FPN backbone.

## V. CONCLUSION

In this work, we propose a unified framework for instance, semantic and panoptic segmentation. The network is end-to-end trainable, has a shared residual FPN backbone and multiple heads for each task. The first contribution of the paper is an improved semantic segmentation head on top of the Feature Pyramid Network with Pyramid Pooling Modules. The second contribution of our work is the design of an improved panoptic head by a mask attention module at background and instance level. The proposed network heads bring improvements compared to the baseline and the panoptic segmentation head refines the semantic output, showing consistent increase in performance.

## ACKNOWLEDGMENT

This work was supported by the EU H2020 UP-Drive project under grant nr. 688652, SEPCA project PN III PCCF, no. 9/2018 and MULTISPECT project PN III PCE, no. 60/2017.

## REFERENCES

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[2] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018.

[3] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1312–1328, 2012.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[5] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. In *arXiv preprint arXiv:1706.05587*, 2017.

[6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[8] A. D. Costea, A. Petrovai, and S. Nedevschi. Fusion scheme for semantic and instance-level segmentation. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3469–3475. IEEE, 2018.

[9] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[10] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[11] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. `https://github.com/facebookresearch/detectron`, 2018.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018.

[15] A. Kirillov, R. Girshick, K. He, and P. Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019.

[16] A. Kirillov, K. He, R. Girshick, and P. Dollár. A unified architecture for instance and semantic segmentation. `http://presentations.cocodataset.org/COCO17-Stuff-FAIR.pdf`, 2017.

[17] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019.

[18] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother. Instancecut: from edges to instances with multicut. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5008–5017, 2017.

[19] I. Kreso, J. Krapac, and S. Segvic. Ladder-style densenets for semantic segmentation of large natural images. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 238–245, Oct 2017.

[20] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018.

[21] J. Li, A. Raventos, A. Bhargava, T. Tagawa, and A. Gaidon. Learning to fuse things and stuff. *arXiv preprint arXiv:1812.01192*, 2018.

[22] Q. Li, A. Arnab, and P. H. Torr. Weakly-and semi-supervised panoptic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 102–118, 2018.

[23] Y. Li, X. Chen, Z. Zhu, L. Xie, G. Huang, D. Du, and X. Wang. Attention-guided unified network for panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7026–7035, 2019.

[24] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[27] S. Liu, J. Jia, S. Fidler, and R. Urtasun. Sgn: Sequential grouping networks for instance segmentation. In *CVPR*, pages 3496–3504, 2017.

[28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[29] A. Newell, Z. Huang, and J. Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems*, pages 2277–2287, 2017.

[30] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–286, 2018.

[31] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[32] A. Paszke, S. Gross, S. Chintala, and G. Chanan. Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. *PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration*, 6, 2017.

[33] P. O. Pinheiro, R. Collobert, and P. Dollár. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pages 1990–1998, 2015.

[34] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *European Conference on Computer Vision*, pages 75–91. Springer, 2016.

[35] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.

[36] E. Romera, J. M. Alvarez, 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.

[37] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[38] J. Uhrig, E. Rehder, B. Fröhlich, U. Franke, and T. Brox. Box2pix: Single-shot instance segmentation by assigning pixels to object boxes. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 292–299. IEEE, 2018.

[39] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. Understanding convolution for semantic segmentation. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 1451–1460. IEEE, 2018.

[40] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[41] Y. Wu and K. He. Group normalization. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[42] X. Yuwen, L. Renjie, Z. Hengshuang, H. Rui, B. Min, Y. Ersin, and U. Raquel. Upsnet: A unified panoptic segmentation network. In *CVPR*, 2019.

[43] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017.