

IMP: Instance Mask Projection for High Accuracy Semantic Segmentation of Things

Cheng-Yang Fu Tamara L. Berg Alexander C. Berg
Facebook AI

Abstract

In this work, we present a new operator, called Instance Mask Projection (IMP), which projects a predicted Instance Segmentation as a new feature for semantic segmentation. It also supports back propagation so is trainable end-to-end. Our experiments show the effectiveness of IMP on both Clothing Parsing (with complex layering, large deformations, and non-convex objects), and on Street Scene Segmentation (with many overlapping instances and small objects). On the Varied Clothing Parsing dataset (VCP), we show instance mask projection can improve 3 points on mIOU from a state-of-the-art Panoptic FPN segmentation approach. On the ModaNet clothing parsing dataset, we show a dramatic improvement of 20.4% absolutely compared to existing baseline semantic segmentation results. In addition, the instance mask projection operator works well on other (non-clothing) datasets, providing an improvement of 3 points in mIOU on Thing classes of Cityscapes, a self-driving dataset, on top of a state-of-the-art approach.

1. Introduction

This paper addresses producing pixel-accurate semantic segmentations. This is relevant for a wide range of applications, from self-driving, where predicting accurate localizations of objects, buildings, people, etc, (as illustrated in the Cityscapes dataset [7]), will be necessary for producing safe autonomous vehicles, to commerce, where accurate segmentations of the clothing items someone is wearing [43] will form a foundational building block for applications like visual search. Many other potential applications can be envisioned, especially in real-world scenarios where intelligent agents are using vision to perceive their surrounding environments, but for this paper we focus on two areas, street scenes and fashion outfits, as two widely differing settings to demonstrate the generality of our method.

We propose combining information from detection results, bounding box and instance mask prediction, as in Mask R-CNN [16]. The core of our approach is a new operator, Instance Mask Projection (IMP), that projects the

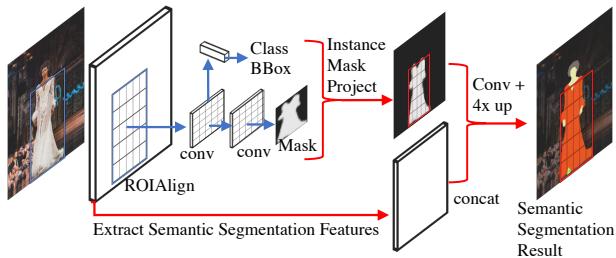


Figure 1: Example of Instance Mask Projection: An Instance Mask Projection operator takes the instance mask as the input (Class, Score, BBox, Mask) and project the results as the feature map for semantic segmentation prediction. In this example, the “Dress” is detected in the Instance Detection pipeline, then is transformed as the feature layer.

predicted masks (with uncertainty) from Mask R-CNN for each detection into a feature map to use as an auxiliary input for semantic segmentation, significantly increasing accuracy. Furthermore, in our implementations the semantic segmentation pipeline shares a trunk with the detector, as in Panoptic FPN [19], resulting in a fast solution.

This approach is most helpful for improving semantic segmentation of objects for which detection works well, movable foreground objects (things) as opposed to regions like grass (stuff). Using the instance mask output from a detector allows the approach to make decisions about the presence/absence/category of an object as a unit, and to explicitly estimate and use the scale of a detected object for aggregating features (e.g. in roi-pooling). In contrast, semantic segmentation must make the decision about object type over and over again at each location using a fixed scale for spatial context. The semantic segmentation prediction deals better with concave shapes than the instance mask prediction, in addition to offering high-resolution output.

As part of validating the effectiveness of this approach we demonstrate several new results:

- The object masks predicted by Mask R-CNN [16] are sometimes more accurate than semantic segmentation for some objects. See Sec. 4.1 and 4.2.

- Following this insight we design the Instance Mask Projection (IMP) operator to project these masks as a feature for semantic segmentation, see Sec. 3.1.
- Segmentation results with IMP significantly improve on the state of the art for semantic segmentation on clothing segmentation. Showing the best results on ModaNet [43], improving mean IOU from 51% for DeepLabV3+ to 71.4%. See sec. 4.2.
- Across three datasets, using features from IMP improves significantly over a Panoptic segmentation baseline (the same system without IMP) and produces state of the art results. See Sec. 4.3.

2. Related Work

Our work builds on current state-of-the-art object detection and semantic segmentation models which have benefited greatly from recent advances in convolution neural network architectures. In this section, we first review recent progress on object localization and semantic segmentation. Then, we describe how our proposed model fits in with other works which integrate both object detection and semantic segmentation.

2.1. Localizing Things

Initially, methods to localize objects in images mainly focused on predicting a tight bounding box around each object of interest. As the accuracy matured, research in object localization has expanded to not only produce a rectangular bounding box but also an instance segmentation, identifying which pixels corresponding to each object.

Object Detection: R-CNN [14] has been one of the most foundational lines of research driving recent developments in detection, initiating work on using the feature representations learned in CNNs for localization. Many related works continued this progress in two-stage detection approaches, including SPP Net [18], Fast R-CNN, [13] and Faster R-CNN [32]. In addition, single-shot detectors YOLO [31], SSD [26] have been proposed to achieve real-time speed. Many other recent methods have been proposed to improve accuracy. R-FCN [9] pools position-sensitive class maps to make predictions more robust. FPN [22] and DSSD [12] add top-down connections to bring semantic information from deep layers to shallow layers. FocalLoss [23] reduces the extreme class imbalance by decreasing influence from well-predicted examples.

Instance Segmentation: Compared to early instance segmentation works [8, 21], Mask R-CNN [16] identifies the core issue for mask prediction as ROI-pooling box misalignment and proposes a new solution, ROI-Alignment using bilinear interpolation to fix quantization error. Path Aggregation Network [25] pools results on multiple layers rather than one to further improve results.

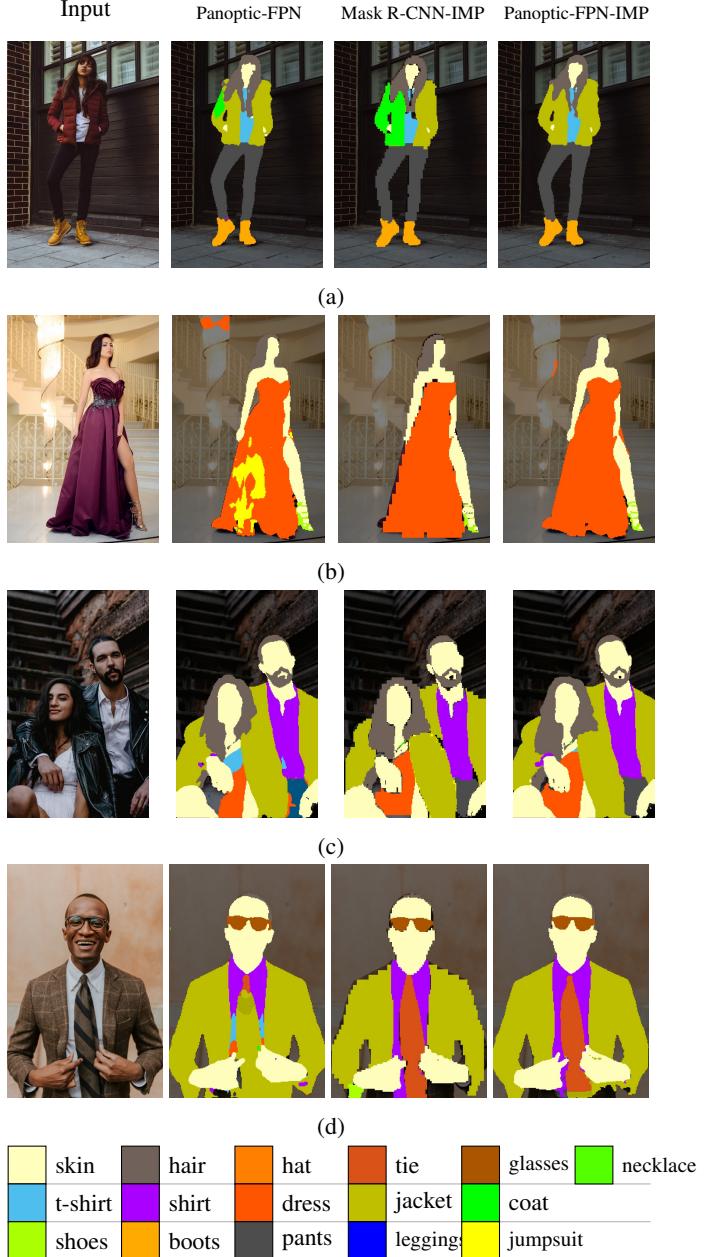


Figure 2: From left to right, images, results of Panoptic-FPN, results of Mask R-CNN-IMP, results of our final model, Panoptic-FPN-IMP. Figure 2b, Figure 2c and 2d show Mask R-CNN-IMP generates cleaner results than Panoptic-FPN. Figure 2a shows combining semantic segmentation features and IMP can fix problems happened in both. Figure 2b shows Mask R-CNN-IMP causes less false positives. The visualization images are not from either Varied Clothing Dataset nor ModaNet [43] to avoid potential copyright questions. All images shown are licensed. See more examples in Figure 6.

2.2. Semantic Segmentation

Fully Convolutional Networks (FCN) [35] has been the foundation for many recent semantic segmentation models. FCN uses convolution layers to output semantic segmentation results directly. Most current semantic segmentation approaches can be roughly categorized into two types, dilated convolution, or encoder-decoder based methods. We describe each, and graphical model enhancements below.

Dilated Convolution: Dilated convolution [39, 4] increases the dilated kernels to learn larger receptive fields with fewer convolutions, producing large benefits in semantic segmentation tasks where long range context is useful. Thus, many recent approaches [6, 41, 40, 34] have incorporated dilated convolution. Deformable Convolution Network [10] takes this idea one step further, learning to predict the sampling area to improve the convolution performance instead of using a fixed geometric structure.

Encoder-Decoder Architecture: SegNet [36] and UNET [33] proposed adding a decoder stage, to upsample the feature resolution and produce higher resolution semantic segmentations. Encoder-decoder frameworks have also been widely adopted in other localization related areas of computer vision, such as Facial Landmark Prediction [17], Human Key Point Detection [28], Instance Segmentation [30], and Object Detection [22, 12].

Graphical Models: Although deep learning approaches have improved semantic segmentation results dramatically, the output result is often still not sharp enough. One common approach to alleviate these issues is to apply a CRF-based approach to make the output more aligned with the color differences. Fully connected CRF [6, 5], and Domain Transform [3] are two such approaches that can be trained with neural networks in an end-to-end manner. Soft Segmentation [1] fuses high-level semantic information with low-level texture and color features to carefully construct a graph structure, whose corresponding Laplacian matrix and its eigenvectors reveal the semantic objects and the soft transitions between them. Soft segments can then be generated via eigen decomposition. Although using graphical models can make the prediction boundary align with the color differences, it can also cause small objects to disappear due to excessive smoothing. Additionally, these methods all rely on good semantic segmentation results.

2.3. Combined Detection & Semantic Segmentation

In part due to newly released datasets, such as COCO-Stuff [2], research efforts toward integrating object detection/instance segmentation and semantic segmentation in a single network have increased. Panoptic Segmentation [20] proposed a single evaluation metric to integrate instance segmentation and semantic segmentation. Following these efforts, Panoptic FPN [19] showed that the FPN architecture can easily integrate both tasks in one network trained

end-to-end. Earlier work, Blitznet [11], also demonstrated that both tasks can be improved in multitask training. One related improvement on Panoptic FPN is UPSNet [38]. This uses a projection like our instance mask projection for a different purpose. UPSNet [38] uses projected instance masks stacked with semantic segmentation outputs to make decision about which type of prediction (an instance mask or semantic segmentation) to use at each location. This decision is made using softmax (without learning). Instead our approach uses the projected instance masks as features to improve semantic segmentation, as orthogonal improvement.

Although we use Mask R-CNN [16] / Panoptic FPN [19] architectures for producing instance segmentation and semantic segmentation predictions, our mask project operator is general and could alternatively make use of other instance and semantic segmentation methods as baseline models. Our method can easily take advantage of future development on both tasks to provide better combined results.

3. Model

Our goal is to develop a joint instance/semantic segmentation framework that can directly integrate predictions from instance segmentation to produce a more accurate semantic segmentation labeling. Our model is able to take advantage of recent advances in instance segmentation algorithms like Mask R-CNN [16] as well as advancements in semantic segmentation models [19]. In this section, we first explain the proposed Instance Mask Projection (IMP) operator (Sec 3.1). Next we describe how this is used to augment and improve various base models (Sec 3.2).

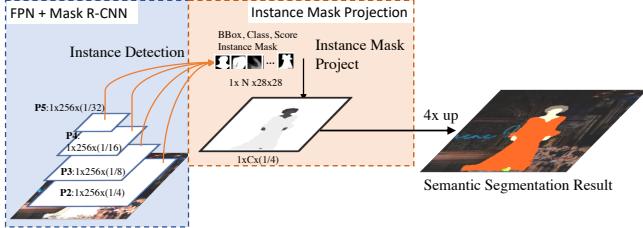
3.1. IMP: Instance Mask Projection

The Instance Mask Projection operator projects the segmentation masks from an instance mask prediction, defined on a detection bounding box, onto a canvas defined over the whole image. This canvas is then used as an input feature layer for semantic segmentation¹.

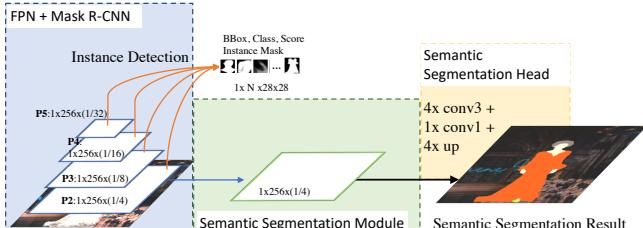
Each predicted instance mask has a Class, Score, BBox location, and $h \times w$ Mask². First the score for each pixel in the Mask is scaled by the object Score for the Class. Then locations in the canvas layer for the Class are sampled from the scaled mask. Note that the canvas is updated only if the scaled mask value is larger than the current canvas value. This is illustrated in Figure 1 where a “dress” is detected by Mask R-CNN and then projected onto the canvas in its detected BBox location. The projected layer shows the low resolution Instance mask which predicts outline of the dress, while the next step of semantic segmentation uses some of the FPN feature layers as well as the canvas as features and will produce a more accurate parse.

¹The resolution of the canvas can be chosen according to which feature layer is attached.

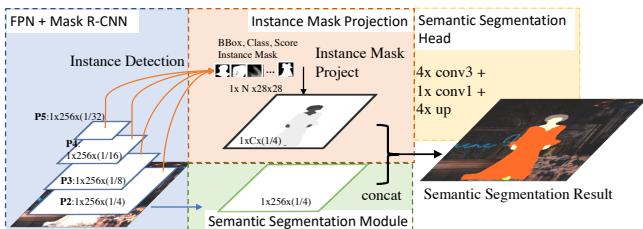
²The resolution of Mask is 28×28 in Mask R-CNN



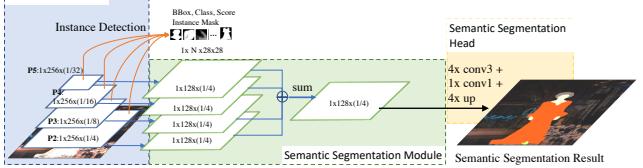
(a) Mask R-CNN-IMP



(b) Panoptic-P2



(c) Panoptic-P2-IMP



(d) Panoptic-FPN

Figure 3: Variants of models we used in the experiments. (a) Mask R-CNN-IMP Uses the IMP to generate the semantic segmentation prediction directly without any learning parameters. (b) Panoptic-P2 uses the P2 layer in FPN to generate semantic segmentation, which is the minimal way to add semantic segmentation in FPN architecture. (c) Panoptic-P2-IMP demonstrates how to apply IMP on Panoptic-P2. (d) Panoptic-FPN combines the features layers {P2, P3, P4, P5} for semantic segmentation. See Figure 4 for Panoptic-FPN-IMP.

This operation can be formulated as follows:

$$\text{canvas}(c, p_{xy}) = \max(\text{canvas}(c, p_{xy}), S_i M_i(\text{pre}_i(p_{xy}))),$$

where there is a canvas layer for each class c , p_{xy} is a location in the canvas, pre_i maps a point in the canvas to a location in the instance mask M_i for bounding box i , and S_i is the detection score for box i . Note this is only computed for p_{xy} where $\text{pre}_i(p_{xy})$ is in the box.

This operator is applied over all detection boxes for each class independently to obtain the canvas($C \times H/scale \times W/scale$). In the experiments the scale is 4, but this can be adjusted according to the attached feature layer.

We concatenate the IMP canvas with the feature layer(s) (either P2 or P2-5) to let the network use this as a strong prior for object location, allowing the semantic segmentation part of the model to focus on making improvements to the instance predictions during learning.

3.2. Adding IMP to Base Models

Mask R-CNN-IMP

Figure 3a illustrates **Mask R-CNN-IMP** which uses Mask R-CNN as a base model and adds IMP to project the instance masks to a canvas used as an approximate semantic segmentation. This does not involve any learning or additional processing for semantic segmentation after projection and already performs well for some objects.

Panoptic-P2, Panoptic-P2-IMP, Semantic-P2

Next we consider lightweight versions of Panoptic FPN [19] as the base model. Panoptic FPN extends the Mask R-CNN network architecture to predict both instance segmentation and semantic segmentation. The added semantic segmentation head takes input from multiple layers of the Feature Pyramid Network (FPN) [22] used in Mask R-CNN. We perform some experiments with a lightweight version we call **Panoptic-P2** that only takes features from the P2 layer of the FPN for use by the semantic prediction head (and does not use group norm) shown in Figure 3b. When we also remove the RPN and bounding box prediction heads from **Panoptic-P2**, leaving just the semantic head attached to P2 we call the network **Semantic-P2**. We experiment with adding instance mask projection to **Panoptic-P2**, and call this **Panoptic-P2-IMP** (illustrated in Figure 3c).

Panoptic-FPN, Panoptic-FPN-IMP, Semantic-FPN

Next, we experiment with adding IMP to the full Panoptic FPN [19] calling this **Panoptic-FPN-IMP**. We also experiment with two ablated versions, **Panoptic-FPN** alone (see Figure 3d) and **Semantic-FPN** which drops the RPN and bounding box heads from Panoptic-FPN.

Figure 4 illustrates Panoptic-FPN-IMP which uses the $\text{conv3x3}(128) + \text{GroupNorm} [37] + \text{ReLU} + \text{Bilinear up-sampling}(2x)$ as the upsampling stage. For P3(scale/8), P4(scale/16), P5(scale/32) layers, we first upsample each to (1/4) scale. For the P2 layer, we apply conv3x3 to reduce the dimension from 256 to 128. Then, we sum these 4 layers to $(128 \times H/4 \times W/4)$ and concatenate with the Instance Mask projected layer to form the feature layer($(128 + C) \times H/4 \times W/4$). Finally, we apply 4 conv3x3 and 1 conv1x1 layers to generate semantic segmentation predictions. In contrast to FPN-P2, all conv3x3 use Group Norm.

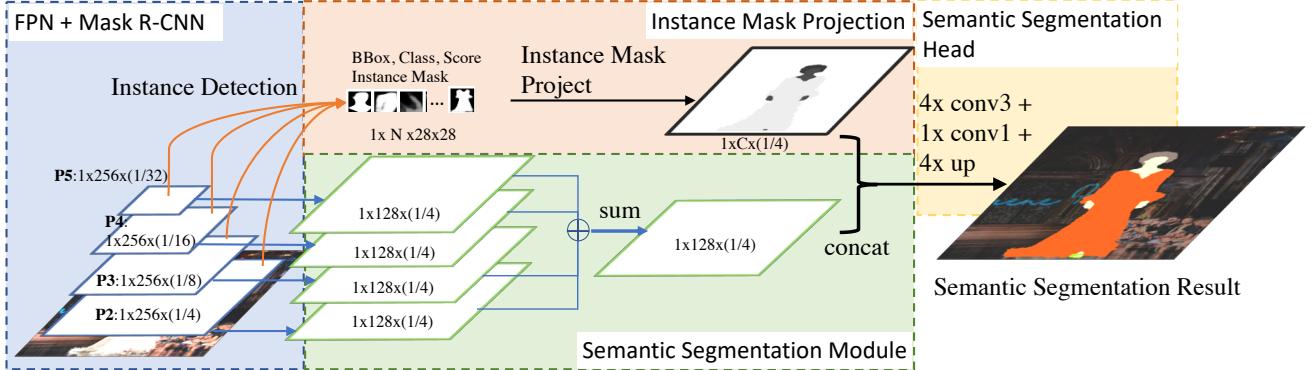


Figure 4: Architecture: **Panoptic-FPN-IMP**: Our full model contains four parts. The first part is FPN + Mask R-CNN which is used for Object/Instance Detection. The Instance Mask Projection Module takes the output of instance detection to generate the feature layer($1 \times C_x 1/4$). For the Semantic Segmentation Moduel, we adopts the Panoptic FPN [19] which upsamples and transforms {P2, P3, P4, P5} to $1 \times 128 \times 1/4$ and sums them. Then we concatenate the results of instance mask projection and semantic segmentation module and forward to the semantic segmentation prediction head. See Figure 3 for other models.

3.3. Training

We adopt a two stage training solution, first training a Mask R-CNN detection/instance segmentation model then using this as an initial prediction for training our full model. Pre-training is incorporated for practical reasons to reduce training time (without pre-training the IMP will vary significantly over training iterations, making convergence slow). In the first stage, we follow the Mask R-CNN training settings but adjust the parameters for 4 GPU machines (Nvidia 1080 Ti) by following the Linear Scaling Rule [15]. For implementation we use PyTorch v1.0.0 [29] and base our code on the maskrcnn-benchmark repository [27].

4. Experiments

We evaluate our proposed model on two different tasks: clothing parsing and street scene segmentation.

4.1. Varied Clothing Dataset

The Varied Clothing Dataset is for clothing parsing – where the goal is to assign an apparel category label (e.g. shirt, skirt, sweater, coat, etc) to each pixel in a picture containing clothing. This is an extremely challenging segmentation problem due to clothing deformations and occlusions due to layering. The dataset depicts 25 clothing categories, plus skin, hair, and background labels, with pixel-accurate polygon segmentations, hand labeled on 6k images. The dataset covers a wide range of depictions, including: real-world pictures of people, layflat images (clothing items arranged on a flat surface), fashion-runway photos, and movie stills. Special care is taken to sample clothing photos from around the world, across varied body shapes, in widely varied poses, and with full or partial-bodies visible.

Since this dataset was initially collected for clothing parsing, a single garment may be split into multiple seg-

ments (e.g. a shirt worn under a buttoned blazer may appear as a segment at the neck, plus 2 shirt cuff segments at each wrist). To convert the semantic segmentations into instance annotations, each segment (connected component) is treated as an instance with corresponding bounding box. This definition is slightly different than COCO [24] or Cityscapes [7] and produces more small instances. However, we experimentally observe benefits to this approach over combining all segments from a garment into a single instance/BBox because it doesn't require the model to make long range predictions across large occlusions.

In our experiments, the train and validation sets contain 5493 and 500 images respectively and all images are 1280×720 pixels or higher. For training the first stage, we use an ImageNet Classification pre-trained model, with prediction layer weights initialized according to a normal distribution(mean=0, standard derivation=0.01). We set batch size to 8, learning rate to 0.01, and train for 70,000 iterations, dropping the learning rate by 0.1 at 40,000 and 60,000 iterations. We also use this setting for training the second stage (including the semantic segmentation branch). For the input image, we resize the short side to 800 pixels and limit the long side to 1333.

Table 1 shows the performance of our models under different settings with ResNet-50 as the backbone network. First, we report the performance of baseline instance (row 1) and semantic segmentation models (rows 3-4). Next, we show results on Panoptic models that integrate instance and semantic segmentation (Panoptic-P2 and Panoptic-FPN, rows 5 and 7). Adding our proposed IMP operator significantly increases semantic segmentation performance when incorporated into each of these base models (rows 6 and 8), improving absolute performance of

	Model	BBox	Mask	Semantic mIOU	mAcc
1	Mask R-CNN	29.9	26.7	NA	NA
2	Mask R-CNN-IMP			43.91	56.93
3	Semantic-P2	NA	NA	37.00	48.57
4	Semantic-FPN	NA	NA	42.66	55.19
5	Panoptic-P2	29.8	26.4	37.14	48.82
6	Panoptic-P2-IMP	30.6	26.8	46.59	59.24
7	Panoptic-FPN	29.6	26.7	45.01	57.08
8	Panoptic-FPN-IMP	30.4	26.8	47.03	61.52

Table 1: Ablation Study on Varied Clothing Dataset. The backbone network is ResNet-50. We train the model with different settings, Panoptic-P2 v.s. Panoptic-FPN, w/wo Instance Mask Projection(IMP), w/wo BBox/Mask prediction head. For the BBox, and Mask, we use the COCO evaluation metric. For the semantic segmentation metric, we use meanIOU and mean Accuracy.

Panoptic-P2 by 9.45 mIOU and 1.42 in mAcc, and improving Panoptic-FPN by 2.02 mIOU and 4.44 in mAcc. For reference, we also experiment with adding IMP to the base Mask R-CNN model (row 2), and achieve semantic segmentation performance better than Semantic-FPN and Panoptic-P2, and comparable to Panoptic-FPN without requiring any dedicated semantic segmentation branch.

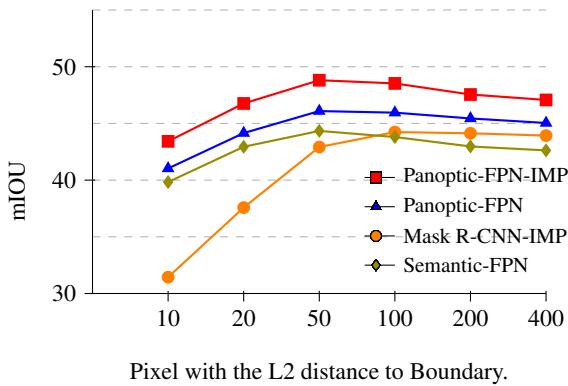


Figure 5: Analysis of performance of pixel within the distance to the boundary. In this Figure, we adopt the Panoptic FPN as the backbone network and shows 4 models, Semantic-FPN, Mask R-CNN-IMP, Panoptic-FPN, and Panoptic-FPN-IMP.

Another question we consider is how much this method helps refining object boundaries, since producing an accurate object contour may be necessary for applications like visual search or virtual clothing try-on. In Figure 5, we analyze the mIOU of pixels within 10-400 L2 distance from the boundary. Generally, we observe that for pixels close to the boundary, semantic and instance/semantic methods all perform much better than Mask R-CNN-IMP and this gap decreases for larger distances. This is because Mask R-

CNN generates 28×28 instance masks. Therefore, once we project the instance segmentation results on the canvas, the boundary will not be sharp, but pixels near the center of the object will be labeled correctly. We also generally observe larger improvements of the IMP operator on pixels near the boundary, with benefits dropping off for central pixels.

In Figure 2, we show some qualitative examples. In some cases, 2b, 2d, Mask R-CNN-IMP already produces a better semantic segmentation than the Panoptic-FPN architecture. We also observe that often, when an object is small (tie, watch), or plain and covering a large area, IMP enhanced methods generally perform better. In Figure 2a, by combining the semantic segmentation features and IMP, our model fixes category confusions occurring on different regions of an object. Although most training images in the Varied Clothing Dataset only contain one person per image, we see that our model generalizes well to complicated examples containing multiple people (Figure 2c).

4.2. ModaNet

ModaNet [43] is a large clothing parsing dataset, containing annotations for BBox, instance-level masks, and semantic segmentations. It contains 55k images (52,377 images in training and 2,799 images in validation), sampled on an existing fashion focused dataset of images from the Chictopia website. The ModaNet data is relatively low resolution (640x480 or smaller) compared to the Varied Clothing Dataset data, sampled to generally contain a single full-body depiction of a standing person, centrally located in the image. 13 clothing categories are labeled (without skin, hair, or background) at relatively high fidelity (but less pixel-accuracy than the Varied Clothing Dataset).

We use a similar two-stage ImageNet classification pre-training method as for the Varied Clothing Dataset, training for 90k iterations, dropping the learning rate at 60k and 80k iterations. Here, we resize the input image to limit its short side to 600 and long side to 1000. During training, we use multi-scale training by randomly changing the short side to {400, 500, 600, 700, 800}.

Table 3 shows experimental results demonstrating the addition of the IMP operator. We evaluate baseline models, Semantic-P2 and Panoptic-P2, 64.60% and 65.93% mIOU, respectively. Compared to these models, we see that Mask R-CNN-IMP can generate better results on semantic segmentation without a dedicated semantic segmentation head. This also matches our previous experiments on the Varied Clothing Dataset. Adding IMP to Panoptic-P2, Panoptic-P2-IMP achieves a semantic performance of 69.65%, outperforming Panoptic-P2 by 3.72% mIOU and Panoptic-FPN-IMP even further improves mIOU to 71.41%.

In Table 2, we also train our final model, Panoptic-FPN-IMP with ResNet-101 and compare to the baseline results provided by ModaNet [43]. First, our model achieves 20.4% absolutely mIOU improvement compared

Model	mean	bag	belt	boots	foot-wear	outer	dress	sunglasses	pants	top	shorts	skirts	head wear	scarf& tie
FCN-32 [35]	35	27	12	32	33	36	28	25	51	38	40	28	33	17
FCN-16 [35]	37	26	19	32	38	35	25	37	51	38	40	23	41	16
FCN-8 [35]	38	24	21	32	40	35	28	41	51	38	40	24	44	18
FCN-8satonce [35]	38	26	20	31	40	35	29	36	50	39	38	26	44	16
CRFasRNN [42]	41	30	18	41	39	43	32	36	56	40	44	26	45	22
DeepLabV3+ [6]	51	42	28	40	51	56	52	46	68	55	53	41	55	31
Ours:														
R50 Panoptic-P2-IMP	69.7	74.8	57.4	59.7	59.4	69.2	64.2	68.5	77.2	67.7	71.9	62.7	75.3	97.5
R50 Panoptic-FPN-IMP	71.1	77.1	58.1	57.9	59.1	72.2	68.2	68.4	80.4	68.7	72.5	67.9	76.2	97.9
R101 Panoptic-FPN-IMP	71.4	77.9	59.0	58.8	59.4	72.0	68.3	68.6	79.3	69.1	74.1	67.8	76.4	97.9

Table 2: Comparison to the baseline models provided by ModaNet on IOU metric. Our model shows 20.4% absolute improvement for mean IOU. For certain categories, especially those whose size is quite small such as belt, sunglasses, headwear and scarf & tie, our models show dramatic improvement. For simplicity, we use R50 and R101 to represent ResNet-50 and ResNet-101.

Model	BBox	Mask	Semantic (mIOU)
Semantic-P2	NA	NA	64.60
Panoptic-P2	57.2	55.5	65.93
Mask R-CNN-IMP	57.2	55.5	66.23
Panoptic-P2-IMP	58.0	55.9	69.65
Panoptic-FPN-IMP	57.8	55.6	71.41

Table 3: Results on ModaNet with ResNet-50 as the backbone model. Panoptic-P2-IMP and Mask R-CNN-IMP both provide improvements on semantic segmentation compared to Semantic-P2 and Panoptic-P2.

to the best performing semantic segmentation algorithm, DeepLabV3+, provided by ModaNet. Plus, we achieve more consistent results, scoring over 50% IOU for each class. Compared to the baseline results, our model does extremely well on small objects, e.g. belt, sunglasses, headwear, scarf&tie (on scarf&tie we achieve 97.9% mIOU). We have some speculations about these improvements. Compared to semantic segmentation methods which tend to base their predictions on fixed scale local regions, object detection takes context from the dynamically chosen region around the object, providing an advantage for segmentation. We also observe improvements on confusing classes, e.g. the bottom part of a dress is visually similar to a skirt. Purely semantic segmentation methods may not be able to differentiate ambiguous cases as well as methods that exploit context determined by object detection.

4.3. Cityscapes

We also experiment on Cityscapes [7], an ego-centric self-driving car dataset. All images are high-resolution (1024×2048) with 19 semantic segmentation classes, and instance-level masks for 8 thing-type categories. The collection contains two sets, fine-annotation and coarse-annotation sets. We focus our experiments on fine-annotation, containing 2975/500/1525 train/val/test images.

For Cityscapes, we use the COCO model as the pre-

trained model, reusing the weights in the prediction layer for all classes except “Rider” which does not exist in COCO (weights are randomly initialized). Then, the input is resized to 1024×2048 , or 800×1600 randomly. We follow Panoptic FPN [19] to add three data augmentations: *multi-scaling*, *color distortion*, and *hard bootstrapping*. For multi-scaling, the short side of the input image is resized to $\{512, 724, 1024, 1448, 2048\}$ randomly and cropped to 512×1024 . The color distortion randomly increases/decreases brightness, contrast, and saturation 40%, and shifts the Hue $\{-0.4, 0.4\}$. Hard bootstrapping selects the top 10, 25, 50 percent of pixels for the loss function. In contrast to Varied Clothing Dataset and ModaNet, we skip the first-stage training, since the pretrained model from COCO already provides strong enough performance. We set batch size to 16, learning rate to 0.005, and train for 130,000 iterations, dropping the learning rate by 0.1 at 80,000 and 110,000 iterations.

For Cityscapes, we focus evaluations on the FPN-Panoptic network (ablation study in Table 5). Model(a) is the Mask R-CNN model. Model(b) is the Panoptic-FPN model without data augmentation. For ColorJitter, model(b) and (d) are the comparison set (improvement from ColorJitter is not clear). In model(d) to model(h), Multi-scale training definitely helps a lot and also reduces overfitting on BBox/Mask prediction. For hard bootstrapping, we see consistent improvements when setting the lower ratio from Model(e), Model(f), to Model(g). Instance Mask Projection provides around 1.35/1.5 improvement in Model(b) to Model(c) without any data augmentation and Model(i) to Model(j) with all data augmentations.

Compared to the Varied Clothing Dataset and ModaNet, we observe less dramatic overall improvement from IMP. However, one reason is that only 8 of 19 classes are “thing” like categories where we expect our method to be most helpful. In Table 4, we show two comparison sets (with and without data augmentation) for each Cityscapes class. For the Stuff classes, the difference are minor, except ‘Wall’

Type	Stuff class										Things class								
Model	road	side-walk	building	wall	fence	pole	traffic light	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	
<i>Without all the Data Augmentation</i>																			
IMP	97.7	81.7	91.2	41.2	51.7	58.8	67.3	74.6	91.6	59.3	93.8	81.2	60.3	93.6	61.4	80.4	63.2	57.0	76.1
IMP	97.6	81.5	91.2	39.6	52.0	59.2	66.6	74.9	91.5	59.7	93.8	81.9	64.7	93.8	63.9	81.6	74.0	63.5	76.7
<i>With all the Data Augmentation</i>																			
IMP	97.7	82.5	91.7	45.0	56.4	61.4	69.6	77.1	91.7	60.1	94.3	82.4	64.0	94.7	74.5	84.5	77.6	62.9	77.9
IMP	97.9	83.6	91.4	38.3	55.9	62.0	69.9	77.5	91.9	59.8	94.5	83.5	69.1	95.1	83.9	91.4	83.1	67.2	78.7

Table 4: Comparisons of per Class IOU with and without IMP on Cityscapes. We show two scenarios without (top) and with (bottom) data augmentation. We see Instance Mask Projection(IMP) improves both scenarios. For Thing classes, we see 4.2/3.2 mIOU improvement with/without all data augmentation.

(-1.6/-6.7). For the Thing classes, certain classes are improved dramatically, especially those that have fewer training instances or that are smaller, i.e. Rider, Truck, Bus, Train, Motorcycle. In fact, over all Thing classes we observe a mIOU increase of 4.2/3.2, with and without data augmentation respectively.

Besides ResNet-50, we also train our final model, Panoptic-FPN-IMP with ResNet-101 and ResNeXt-101-FPN to compare with state-of-the-art methods on Cityscapes val set (Table 6). Our method is still better than Panoptic FPN [19], though the improvements are reduced when using more complex models. We still see our simple model can achieve similar performance to those models using heavily engineering methods.

	Model	Color	MS	BS	IMP	Box	Mask	mIOU
a	Mask R-CNN					40.9	35.5	NA
b	Panoptic-FPN				✓	36.9	32.7	72.74
c						36.9	32.5	74.09
d		✓				36.8	32.8	73.12
e		✓		0.50	✓	37.8	34.0	73.81
f		✓		0.25		38.4	34.1	73.93
g		✓		0.10		38.7	34.7	74.94
h		✓			✓	39.9	35.9	75.99
i		✓		0.10		40.7	36.5	76.11
j		✓		0.10	✓	39.8	35.8	77.49

Table 5: Performance Analysis of each module used on Cityscapes val set. For simplicity, we use the following abbreviation: **MS**:multi-scale training, **Color**:Color Jitter, **BS**:Hard Bootstrapping, **IMP**:Instance Mask Projection,

4.4. Inference Speed Analysis

Table 7, shows some speed performance analysis for each dataset. Due to the different number of instance classes and input resolutions, the speed performance of models can vary. In experiments, we find the results are quite consistent and very efficient, adding IMP only costs 1~2 ms in inference on top of each baseline model. In all experiments, the result is from a single output without any bells and whistle.

Method	Backbone	mIOU
PSANet101 [41]	ResNet-101-D8	77.9
Mapillary [34]	WideResNet-38-D8	79.4
DeeplabV3+ [6]	X-71-D16	79.6
Panoptic FPN [19]	ResNet-101-FPN	77.7
	ResNeXt-101-FPN	79.1
Ours:Panoptic-FPN-IMP	ResNet-50-FPN	77.5
	ResNet-101-FPN	78.3
	ResNeXt-101-FPN	79.4

Table 6: Comparisons on Cityscapes val set. Our models obtain 0.6 and 0.3 mIOU improvement over Panoptic-FPN [19] on the same backbone architectures.

Resolution	Backbone	Model	Speed(ms)
Varied Clothing Dataset			
800×1333	R50	Mask R-CNN	92
		Mask R-CNN-IMP	94
		Semantic-FPN	103
		Panoptic-FPN	110
		Panoptic-FPN-IMP	111
ModaNet			
600×1000	R50	Panoptic-FPN-IMP	72
	R101	Panoptic-FPN-IMP	87
Cityscapes			
1024×2048	R50	Mask R-CNN	151
		Panoptic-FPN	194
		Panoptic-FPN-IMP	195
	R101	Panoptic-FPN-IMP	243
	X101	Panoptic-FPN-IMP	401

Table 7: Speed performance analysis. In this table, we show the speed performance for each model. For simplicity, we use the following abbreviations:**R50**:ResNet-50. **R101**:ResNet-101. **X101**:ResNeXt-101

5. Conclusion

In this work, we propose a new operator, Instance Mask Projection, which projects the results of instance segmentation as a feature representation for semantic segmentation. This operator is simple but powerful. Experiments adding IMP to Panoptic-P2/Panotpic-FPN show consistent

improvements, with negligible increases in inference time. Although we only apply it on the Panoptic-P2/Panoptic-FPN, this operator can generally be applied to other architectures as well.

Appendix

Varied Clothing Dataset Classes

Class	Super Class	# Train	# Val	Area(x^2)
Hair	Body	7,260	635	192
Skin	Body	34,795	3,074	119
Top/T-shirt	G-Top	4,364	424	221
Sweater/Cardigan	G-Top	1,906	148	266
Jacket/Blazer	G-Top	2,360	183	261
Coat	G-Top	1,597	161	279
Shirt/Blouse	G-Top	2,650	244	229
Vest	G-Top	266	20	220
Pants/Jeans	G-Bottom	2,763	217	261
Tights/Leggings	G-Bottom	930	116	214
Shorts	G-Bottom	532	60	203
Socks	G-Bottom	803	80	174
Skirt	G-Bottom	1,281	114	262
Dress	G-Whole	2,728	241	340
Jumpsuit	G-Whole	273	31	370
Shoes	Footwear	6,619	591	118
Boots	Footwear	1,801	109	142
Hat/Headband	Accessories	983	111	192
Scarf/Tie	Accessories	909	88	274
Watch/Bracelet	Accessories	2,627	206	86
Bag	Accessories	3,284	263	186
Gloves	Accessories	431	41	210
Necklace	Accessories	1,711	134	131
Glasses	Accessories	1,329	129	89
Belt	Accessories	1,035	95	110

Table 8: Varied Clothing Dataset Class Definition and statistics.

Table 8 shows the class definition and statistics of the Varied Clothing Dataset. Because we convert each segment(connected component) of semantic segmentation into an instance annotation, the number of training instance is much more than usual. The details can be found in Sec. 4.1 in the main submission. Another is the diverse classes. In contrast to ModaNet [43], in Varied Clothing Dataset, the confusing classes are not grouped. For example, Jacket/Blazer to Coat. This makes it more challenging for semantic segmentation approaches to generate clean results.

In Figure 6, we show more qualitative examples besides Figure 2. We use ResNet-50-FPN as the backbone model and train the model on the Varied Clothing Dataset. Figure 6 contains more diverse photos, such as vintage photos, layflat photos and images with full or half-bodies visible. Although Mask R-CNN-IMP can generate cleaner results than Panoptic-FPN, Mask R-CNN-IMP also incurs

poor performance on boundaries of large objects which was caused by the low resolution output of Mask R-CNN³. Our final model Panoptic-FPN-IMP can generate sharp semantic segmentation results but also makes labeling of pixels from the same objects consistent.

class	Difference		#Instances	Total area
		DA		
Person	0.7	1.1	17,395	64,901,113
Rider	4.4	5.1	1,660	7,169,330
Car	0.2	0.4	26,180	380,112,819
Truck	2.5	9.4	466	14,657,648
Bus	1.2	6.9	350	12,684,337
Train	9.8	5.5	158	11,643,940
Motorcycle	6.5	4.3	705	5,037,718
Bicycle	0.6	0.8	3,433	14,646,908
Average	3.2	4.2		

Table 9: Analysis of Semantic Segmentation classes which are also Instance Segmentation. There is a correlation if the class has fewer instances and area, it gets more improvement from Instance Mask Projection. DA: with Data Augmentation.

More discussions on Cityscapes dataset.

Table 9 shows the mIOU difference of Thing classes of Cityscapes with and without the data augmentation. This Table is part of Table 4 but adds number of instances and area information. We found out the improvement is also similar to the clothing datasets. First, the classes with less examples are improved more. See Train(#158), Bus(#350), Truck(#466), and Motorcycle(#705). Another is the improvement among the confusing classes. Although Rider contains enough examples, its similarity to Person, makes its mIOU lower. Our model is useful to distinguish these cases and increases the mIOU of Rider significantly.

Figure 7 shows the visualization examples of results of our models. We found that the qualitative results are also similar to the clothing datasets. Our final model, Panoptic-FPN-IMP, provides leaner results. See the better results of segments of Bus and Truck in Figure 7a and 7b. Another interesting case is Rider which means the person on the motorcycle or bicycle. The top part of Rider of Panoptic-FPN in Figure 7c and 7d are misclassified as Person. But with Instance Mask Projection, our final model shows correct labeling of all pixels of Rider.

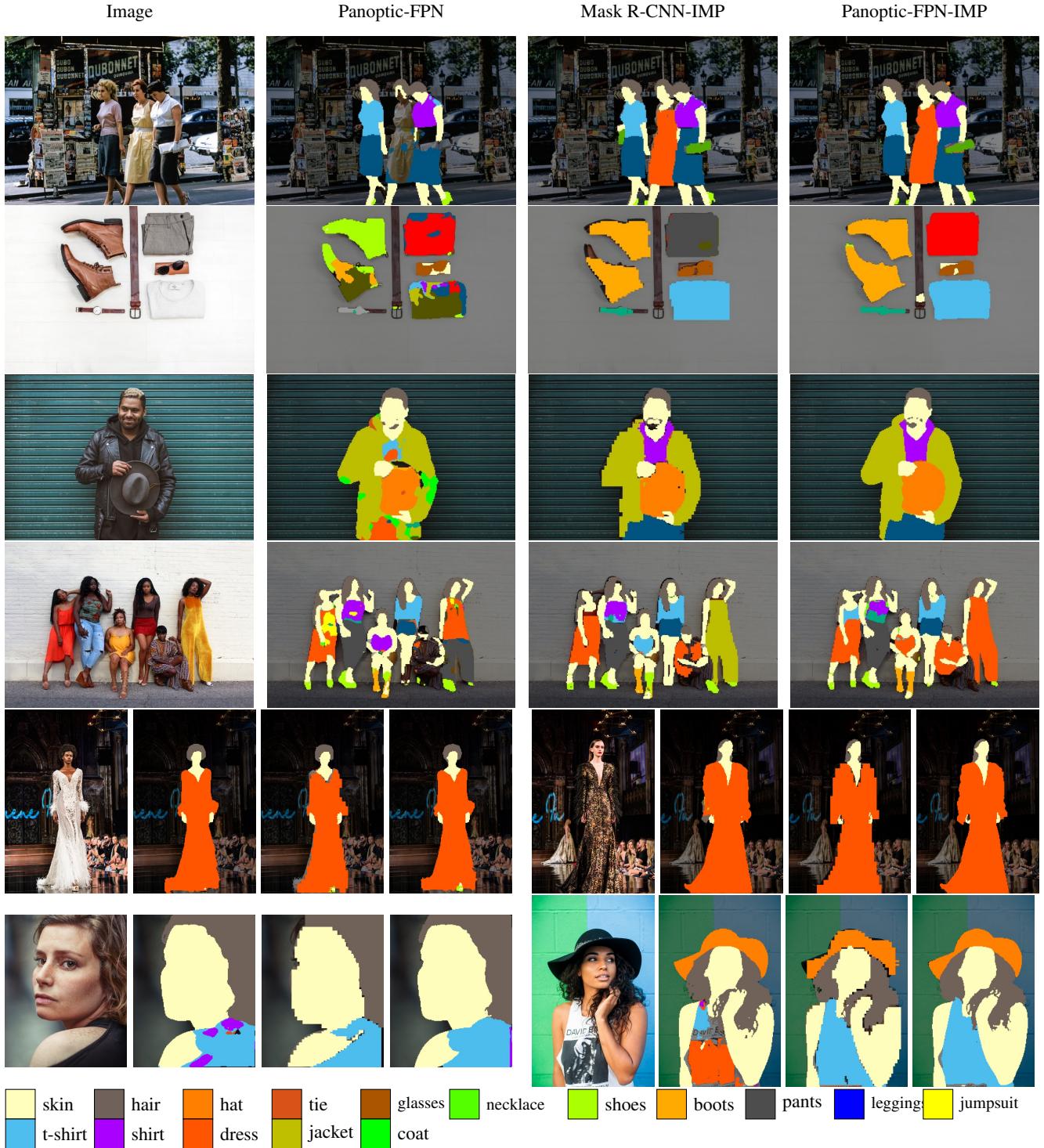


Figure 6: This Figure is an extension of Figure 2. From left to right, images, results of Panoptic-FPN, results of Mask R-CNN-IMP, results of our final model, Panoptic-FPN-IMP. The proposed method, IMP, works well on different types of clothing parsing examples, from vintage images, layflat images, street-fashion examples, fashion-runway photos, and photos with full or partial-bodies visible.

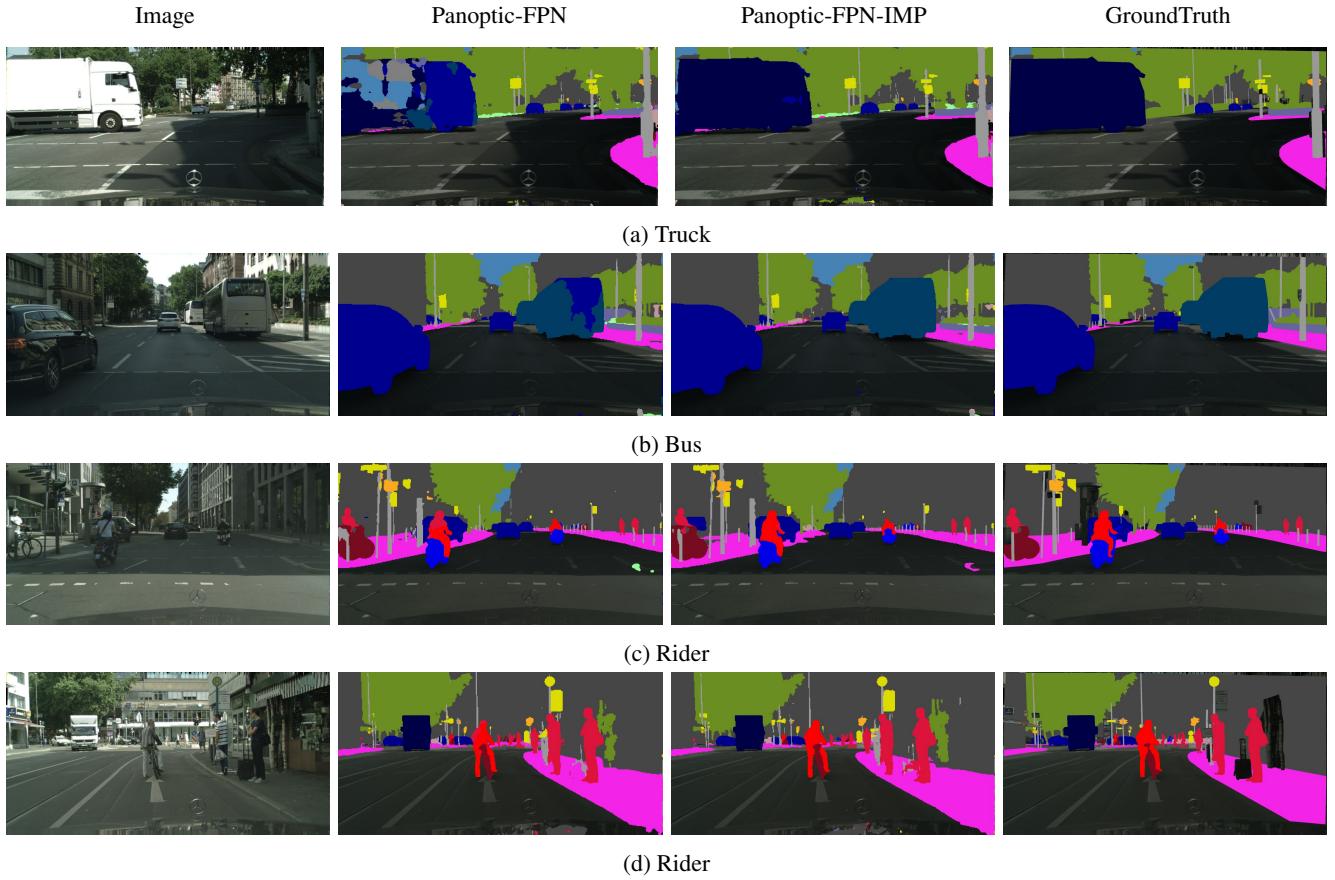


Figure 7: From left to right, images, results of Panoptic-FPN, Panoptic-FPN-IMP and GroundTruth. With the Instance Mask Projection, our final model, shows cleaner results on Truck(a), Bus(b), and Rider(c,d) classes.

References

- [1] Y. Aksoy, T.-H. Oh, S. Paris, M. Pollefeys, and W. Matusik. Semantic Soft Segmentation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 2018. 3
- [2] H. Caesar, J. Uijlings, and V. Ferrari. COCO-Stuff: Thing and Stuff Classes in Context. In *CVPR*, 2018. 3
- [3] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille. Semantic Image Segmentation with Task-Specific Edge Detection Using CNNs and a Discriminatively Trained Domain Transform. 2016. 3
- [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. *PAMI*, 2018. 3
- [5] L.-C. Chen, K. I. Papandreou, George *, K. Murphy, and A. e. c. L. Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. 2015. 3
- [6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *ECCV*, 2018. 3, 7, 8
- [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. 1, 5, 7
- [8] J. Dai, K. He, and J. Sun. Instance-aware Semantic Segmentation via Multi-task Network Cascades. In *CVPR*, 2016. 2
- [9] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In *NeurIPS*, 2016. 2
- [10] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable Convolutional Networks. *ICCV*, 2017. 3
- [11] N. Dvornik, K. Shmelkov, J. Mairal, and C. Schmid. BlitzNet: A Real-Time Deep Network for Scene Understanding. In *ICCV*, 2017. 3
- [12] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. DSSD : Deconvolutional Single Shot Detector. *arXiv:1701.06659*, 2017. 2, 3
- [13] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 2
- [14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2
- [15] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *arXiv:1706.02677*, 2017. 5
- [16] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017. 1, 2, 3
- [17] S. Honari, J. Yosinski, P. Vincent, and C. Pal. Recombinator Networks: Learning Coarse-to-Fine Feature Aggregation. In *CVPR*, 2016. 3
- [18] S. R. Kaiming He, Xiangyu Zhang and J. Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015. 2
- [19] A. Kirillov, R. Girshick, K. He, and P. Dollár. Panoptic Feature Pyramid Networks. *arXiv preprint arXiv:1901.02446*, 2019. 1, 3, 4, 5, 7, 8
- [20] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic Segmentation. *arXiv preprint arXiv:1801.00868*, 2017. 3
- [21] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully Convolutional Instance-aware Semantic Segmentation. In *CVPR*, 2017. 2
- [22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, 2017. 2, 3, 4
- [23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal Loss for Dense Object Detection. In *ICCV*, 2017. 2
- [24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014. 5
- [25] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path Aggregation Network for Instance Segmentation. In *CVPR*, 2018. 2
- [26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single Shot MultiBox Detector. In *ECCV*, 2016. 2
- [27] F. Massa and R. Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. <https://github.com/facebookresearch/maskrcnn-benchmark>, 2018. Accessed: [03/22/2019]. 5
- [28] A. Newell, K. Yang, and J. Deng. Stacked Hourglass Networks for Human Pose Estimation. In *ECCV*, 2016. 3
- [29] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *NeurIPS-W*, 2017. 5
- [30] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollr. Learning to Refine Object Segments. In *ECCV*, 2016. 3
- [31] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *CVPR*, 2016. 2
- [32] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 2015. 2
- [33] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*, 2015. 3
- [34] S. Rota Bulò, L. Porzi, and P. Kontschieder. In-Place Activated BatchNorm for Memory-Optimized Training of DNNs. In *CVPR*, 2018. 3, 8
- [35] E. Shelhamer, J. Long, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. *PAMI*, 2016. 3, 7
- [36] A. K. Vijay Badrinarayanan and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *PAMI*, 2017. 3
- [37] Y. Wu and K. He. Group Normalization. In *ECCV*, 2018. 4
- [38] Y. Xiong*, R. Liao*, H. Zhao*, R. Hu, M. Bai, E. Yumer, and R. Urtasun. UPSNet: A Unified Panoptic Segmentation Network. 2019. 3
- [39] F. Yu and V. Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. In *ICLR*, 2016. 3
- [40] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid Scene Parsing Network. In *CVPR*, 2017. 3

- [41] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia. PSANet: Point-wise Spatial Attention Network for Scene Parsing. In *ECCV*, 2018. [3](#), [8](#)
- [42] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional Random Fields as Recurrent Neural Networks. In *ICCV*, 2015. [7](#)
- [43] S. Zheng, F. Yang, M. H. Kiapour, and R. Piramuthu. ModaNet: A Large-Scale Street Fashion Dataset with Polygon Annotations. In *ACM Multimedia*, 2018. [1](#), [2](#), [6](#), [9](#)