CascadePSP: Toward Class-Agnostic and Very High-Resolution Segmentation via Global and Local Refinement

Ho Kei Cheng* Jihoon Chung*
HKUST
{hkchengad,jchungaa}@cs.ust.hk

Yu-Wing Tai
Tencent
yuwingtai@tencent.com

Chi-Keung Tang
HKUST
cktang@cs.ust.hk

Abstract

State-of-the-art semantic segmentation methods were almost exclusively trained on images within a fixed resolution range. These segmentations are inaccurate for very high-resolution images since using bicubic upsampling of low-resolution segmentation does not adequately capture high-resolution details along object boundaries. In this paper, we propose a novel approach to address the high-resolution segmentation problem without using any high-resolution training data. The key insight is our CascadePSP network which refines and corrects local boundaries whenever possible. Although our network is trained with low-resolution segmentation data, our method is applicable to any resolution even for very high-resolution images larger than 4K. We present quantitative and qualitative studies on different datasets to show that CascadePSP can reveal pixel-accurate segmentation boundaries using our novel refinement module without any finetuning. Thus, our method can be regarded as class-agnostic. Finally, we demonstrate the application of our model to scene parsing in multi-class segmentation.

1. Introduction

Resolution of commodity cameras and displays has significantly increased with 4K UHD (3840 × 2160) being the high industry standard. Despite the demand for high-resolution media, many state-of-the-art computer vision algorithms face various challenges with images with high pixel count. Image semantic segmentation is one of these computer vision tasks. Models for semantic segmentation in deep learning designed for low-resolution images (*e.g.* PASCAL or COCO dataset) often fail to generalize to higher resolution scenarios. Specifically, these models typically use GPU memory linear to the number of pixels, making it practically impossible to directly train a 4K UHD segmentation. High-resolution training data for semantic segmentation is difficult to obtain because pixel-accurate annotation is required, much less that even such high-resolution training data are available, to train a model on very high-resolution images, a much larger receptive field is required to capture sufficient semantics. Plausible workarounds include downsampling and cropping, but the former removes details while the latter destroys image context.

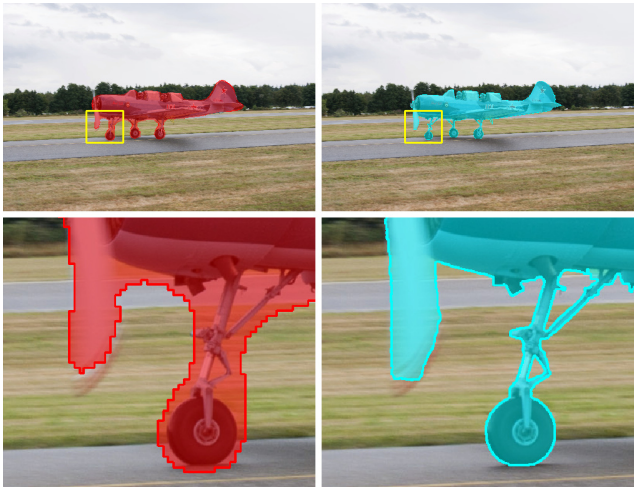


Figure 1. Segmentation of a high-resolution image (3492×2328). **Left**: Produced by Deeplab V3+ [6]. **Right**: Refined by our algorithm.

This paper proposes **CascadePSP**[1], a general segmentation refinement model that refines any given segmentation from low to high resolution. Our model is trained independently and can be easily appended to any existing methods to improve their segmentation, a finer and more accurate segmentation mask of an object can be produced. Our model takes as input an initial mask that can be an output of any algorithm to provide a rough object location. Then our CascadePSP will output a refined mask. Our model is designed in a cascade fashion that generates refined segmentation in a coarse-to-fine manner. Coarse outputs from the early levels predict object structure which will be used as input to the latter levels to refine boundary details. Figure 1 shows that the model not only generates output segmenta-

*Equal contribution. This research is supported in part by Tencent and the Research Grant Council of the Hong Kong SAR under grant no. 1620818.

[1]Source code, pretrained models and dataset are available at https://github.com/hkchengrex/CascadePSP.

tion in very high-resolution but also refines and corrects erroneous boundary to produce more accurate result.

To evaluate on very high-resolution images, we have annotated a high-resolution dataset with 50 validation and 100 test objects with the same semantic classes as in PASCAL, dubbed the BIG dataset. We test our model on PASCAL VOC 2012, BIG, and ADE20K. With a single model *without* using the dataset itself for finetuning, we have achieved consistent improvement over the state-of-the-art methods across these datasets and models. We show that our model does not have to be trained with respect to a specific dataset, or with outputs of a specific model. Rather, performing data augmentation by perturbing the ground truth is sufficient. We also show that our model can be extended to scene parsing for dense multi-class semantic segmentation with straightforward adaptation. Our main contributions can be summarized as:

- We propose CascadePSP, a general cascade segmentation refinement model that can refine any given input segmentations, boosting the performance of state-of-the-art segmentation models without finetuning.
- We further show that our method can be used to produce high-quality and very high-resolution segmentations which has never been achieved by previous deep learning based methods.
- We introduce the BIG dataset that can be used as an accurate evaluation dataset for very high resolution semantic image segmentation task.

2. Related Works

Semantic Segmentation Fully Convolutional Neural Networks (FCN) was first introduced in semantic segmentation in [31] which achieved remarkable progress at the time of introduction. While FCNs capture information from bottom-up, contextual information with wide field-of-view is also important for pixel labeling tasks and is exploited by many segmentation models [3, 5, 14, 17, 32, 39, 51], including image pyramid methods that use multi-scale inputs [5, 9, 14, 22, 23, 36], or feature pyramid methods that use feature maps of different receptive field sizes by spatial pooling [29, 53] or dilated convolutions with different rates [3, 4, 6, 21, 42, 49]. We choose PSPNet [53] for pyramid pooling in our network because the pertinent module is independent of input resolution, thus providing a simple yet effective method to capture contextual information even when the training and testing resolution significantly differ as in our case.

Encoder-decoder models have also been widely used in segmentation methods [1, 6, 21, 25, 27, 33, 37, 42]. They first reduce the spatial dimension to capture high-level semantics and then recover the spatial extent using a decoder. Skip connections [12, 37, 40] can be added to produce sharper boundaries which we have also employed.

Semantic segmentation models typically have a large output stride such as 4 or 8 [2, 3, 4] due to memory and computational limitation. Outputs with stride are usually bilinearly upsampled to the target size, leading to inaccurate boundary labels. Recently, the authors of [7] have proposed Global-Local Networks (GLNet) to solve this problem using a global information branch with a local fine structure network. However, they still require high-resolution training images which are not available for most tasks.

Our method adopts the encoder-decoder model to obtain better semantic and boundary information with a refinement cascade, which also helps to efficiently generate high-resolution segmentations. This formulation also makes our method highly robust and can generalize to high-resolution data without finetuning.

Refining Segmentation FCN based methods typically do not generate very high-quality segmentation. Researchers have addressed this issue with graphical models such as CRF [2, 3, 23, 20, 30, 54] or region growing [10]. They often adhere to low-level color boundaries without fully leveraging high-level semantic information and cannot fix large error regions. Propagation-based approaches [26] cannot handle very high-resolution data due to computational and memory constraints. Separate refinement modules are also used to increase boundary accuracy [35, 46, 50]. They are trained in an end-to-end fashion. Large models are prone to overfitting [50] while shallow refinement networks [35, 47] have limited refinement capability. Contrary, our method has a high model capacity and can be trained independently to repair segmentation using only objectness. Finetuning with the specific model is not required so our training is not hindered by overfitting.

Cascade Network Multi-scale analysis leverages both large and small scale features in many computer vision tasks, such as edge detection [15, 45], detection [24, 28, 41], and segmentation [7, 22, 52]. In particular, a number of methods [22, 45, 52] predict independent results at each stage and merge them to obtain multi-scale information. Our method not only fuses features from coarse scales but uses them as one of the inputs for the next finer level. We will show that adding coarse outputs as input for the next level does not change our formulation and thus the same network can be used recursively for higher resolution refinement.

3. CascadePSP

In this section, we first describe our single refinement module and then our cascade method which makes use of multiple refinement modules for high-resolution segmentation.

3.1. Refinement Module

As illustrated in Figure 2, our refinement module takes an image and multiple imperfect segmentation masks at different scales to produce a refined segmentation. Multi-scale inputs allow the model to capture different levels of structural and boundary information, which allow the network to learn to adaptively fuse the mask features from different

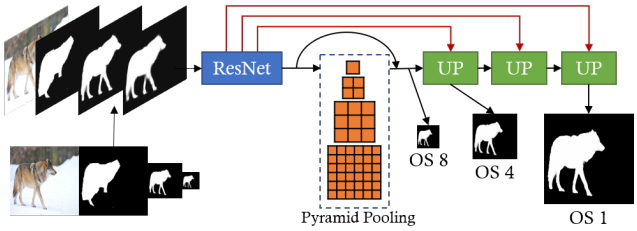


Figure 2. **Refinement module** (RM). Network structure of a single RM, taking three levels of segmentation as inputs to refine the segmentation with different output strides (OS) in different branches. Red lines denote skip-connections. In this paper, we use output strides of 8, 4, and 1.

scales to refine the segmentation at the finest level.

All the input segmentations at lower resolution are bilinearly upsampled to the same size and concatenated with the RGB image. We extract stride 8 feature maps from the inputs using PSPNet [53] with ResNet-50 [16] as the backbone. We follow the pyramid pooling sizes of $[1, 2, 3, 6]$ as in [53] which helps to capture global context. Besides the final stride 1 output, our model also generates intermediate stride 8 and stride 4 segmentations which focus on fixing the overall structure of the input segmentation. We skip stride 2 to provide flexibility to correct local error boundary.

To reconstruct pixel-level image details that are lost in the extraction process, we employ skip-connection from the backbone network and fuse the features using an upsampling block. We concatenate the skip connected features and the bilinearly upsampled features from the main branch, and process them with two ResNet blocks. A segmentation output is generated using a 2-layer 1×1 conv followed by a sigmoid activation.

Loss We produce the best result using cross-entropy loss for the coarser stride 8 output, L1+L2 loss for the finer stride 1 output, and the average of cross-entropy and L1+L2 loss for the intermediate stride 4 output. Different loss functions are applied for different strides because the coarse refinement focuses on the global structure while ignoring local details, while the finest refinement aims to achieve pixel-wise accuracy by relying on local cues. To encourage better boundary refinement, L1 loss on segmentation gradient magnitude is also employed on the stride 1 output. The segmentation gradient is estimated by a 3×3 mean filter followed by a Sobel operator [18]. The gradient loss makes outputs adhere better to the object boundary at the pixel level. As gradient is sparser compared to pixel level loss, we weigh it with α, which is set to 5 in our experiments. The gradient loss can be written as:

$$\mathcal{L}_{grad} = \alpha \cdot \frac{1}{n} \sum_i \|\nabla(f_m(x_i)) - \nabla(f_m(y_i))\|_1$$

where $f_m(\cdot)$ denotes the 3×3 mean filter, ∇ denotes the gradient operator approximated by a Sobel operator, n is the total number of pixels, x_i and y_i are the ith pixel of the ground truth segmentation and output segmentation respectively. Our final loss can be written as:

$$\mathcal{L} = \mathcal{L}^8_{CE} + \frac{1}{2}(\mathcal{L}^4_{L1+L2} + \mathcal{L}^4_{CE}) + \mathcal{L}^1_{L1+L2} + \mathcal{L}^1_{grad}$$

where $\mathcal{L}^s_{CE}$, $\mathcal{L}^s_{L1+L2}$, and $\mathcal{L}^s_{grad}$ denote cross-entropy loss, L1+L2 loss, and gradient loss for output stride s respectively.

Configuration	PASCAL VOC 2012 IoU (%)	PASCAL VOC 2012 mBA (%)
Deeplab V3+	87.13	61.68
Ablation of network structure		
Vanilla FCN	$88.46_{\uparrow 1.33}$	$70.38_{\uparrow 8.70}$
With OS1 only	$88.76_{\uparrow 1.63}$	$71.49_{\uparrow 9.81}$
With OS8 & OS1 only	$88.85_{\uparrow 1.72}$	$71.88_{\uparrow 10.2}$
Ablation of loss function		
CE loss only	$88.73_{\uparrow 1.60}$	$71.07_{\uparrow 9.39}$
L1+L2 loss only	$88.74_{\uparrow 1.61}$	$71.07_{\uparrow 9.39}$
CE and L1+L2 loss only	$88.84_{\uparrow 1.71}$	$71.36_{\uparrow 9.68}$
Ours - Final	$\mathbf{89.01}_{\uparrow \mathbf{1.88}}$	$\mathbf{72.10}_{\uparrow \mathbf{10.4}}$

Table 1. Ablation study of the refinement module. With the proposed 3-level cascade and loss function, we achieve the highest gain over the input segmentation model.

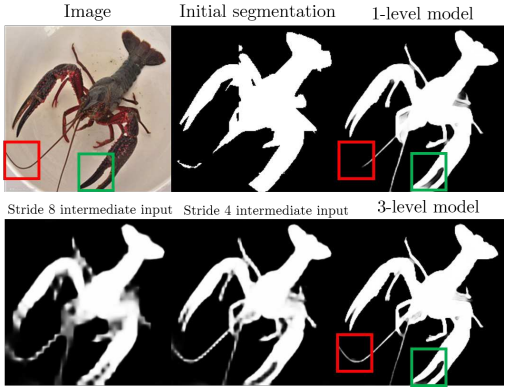


Figure 3. Difference between a 3-level input model and a 1-level input model. The 3-level input model uses small-scale intermediates (bottom row, left two) that, though inaccurate, capture structural information (e.g. the tentacle) to be refined at the later stage.

Ablation Study of Refinement Module We evaluate our method using standard segmentation metric IoU. To highlight the perceptual importance of boundary accuracy, we propose a new **m**ean **B**oundary **A**ccuracy measure (**mBA**). For a robust estimation for images of different sizes, we sample 5 radii in $[3, \frac{w+h}{300}]$ with uniform intervals, compute the segmentation accuracy within each radius from the ground truth boundary, then average these values. Here we perform ablation studies to show the efficacy of our cascade design and loss function. Table 1 shows that our model produces the most significant improvement in IoU and even more significantly in boundary accuracy.

With a multi-level cascade, the module can delegate different stages of refinement to different scales. As shown in Figure 3, the 3-level model uses intermediate small-scale segmentations (will be detailed in Section 3.2) to better capture object structure. Although both models have the same

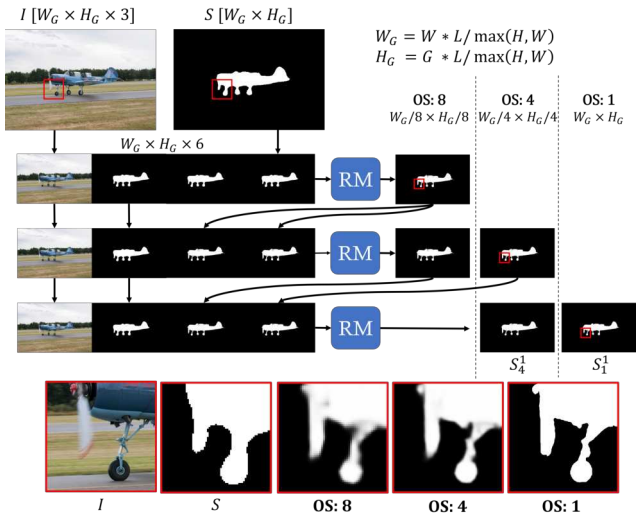


Figure 4. **Global step** refines the whole image using the same refinement module (RM) to perform a 3-level cascade with output strides (OS) of 8, 4, and 1. The cascade is jointly optimized, capturing object structure at large output strides and accurate boundary at small output strides (i.e., with a higher resolution).

receptive field, the 3-level model can better leverage structural cues to produce a more detailed segmentation than the 1-level model.

3.2. Global and Local Cascade Refinement

In testing, we use the **Global step** and the **Local step** to perform high-resolution segmentation refinement by employing the *same* trained refinement module. Specifically, the Global step considers the whole resized image to repair structure while the Local step refines details in full resolution using image crops. The same refinement module can be used recursively for higher resolution refinement.

3.2.1 Global Step

Figure 4 details the design of the Global step which refines the whole image with a 3-level cascade. As the full-resolution image during testing often cannot be fit into the GPU for processing, we downsample the input such that the long-axis has length L while maintaining the same aspect ratio.

Inputs to the cascade are initialized with the input segmentation, which is replicated to keep the input channel dimension constant. After the first level of the cascade, one of the input channels will be replaced with the bilinearly upsampled coarse output. This is repeated until the last level, where the input consists of both the initial segmentation and all outputs from previous levels.

This design enables our network to fix segmentation errors progressively while keeping details present in the initial segmentation. With multiple levels, we can roughly delineate the object and fix larger error in coarse levels, and focus on boundary accuracy in fine levels using more robust features provided by the coarse levels.

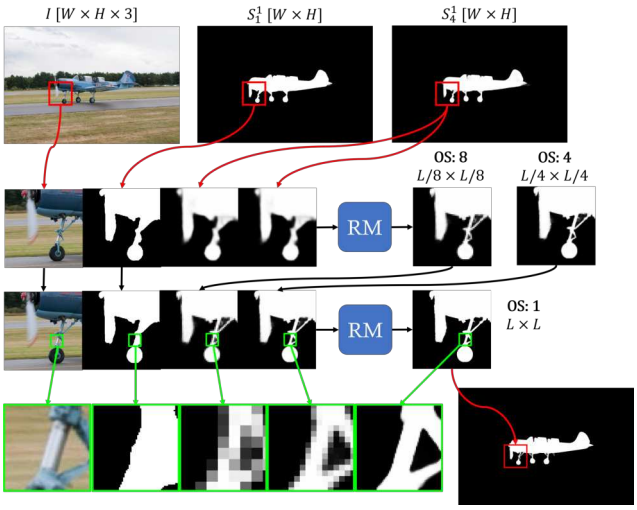


Figure 5. **Local step** takes the outputs from the Global step, and feeds them through a 2-level cascade constructed with the same refinement module with output strides of 4 and 1 respectively. This figure shows the process for a single image crop as shown by the red lines, and green lines show visual improvements of our refinement. Outputs from all the image crops will be fused as the final output.

3.2.2 Local Step

Figure 5 illustrates the details of the Local step. Very high-resolution images cannot be processed in a single pass even with modern GPUs due to the memory constraint. Also, the drastic change of scale between training and testing data will cause poor segmentation quality. We leverage our cascade model to first perform global refinement using a downsampled image, and then perform local refinement using image crops *from a higher resolution image*. These crops enable the Local step to handle high-resolution images without high-resolution training data while taking image context into account due to the Global step.

During the Local step, the model takes the two outputs of the last level of the Global step, denoted as S_4^1 and S_1^1. Both outputs are bilinearly resized to the original size of the image $W \times H$. The model takes image crops of size $L \times L$ and 16 pixels will be chipped away from each side of the crop output to avoid boundary artifacts, with exceptions at the image border. The crops are taken uniformly with a stride of $L/2 - 32$ such that most pixels are covered by four crops, and invalid crops that go beyond image borders are shifted to align with the last row/column of the image. The image crops are then fed into a 2-level cascade with output stride of 4 and 1 respectively. In fusion, the outputs from different patches might disagree with each other due to different image context, and we resolve this by averaging all the output values. For images with even higher resolution, we can apply the local step recursively in a coarse-to-fine manner.

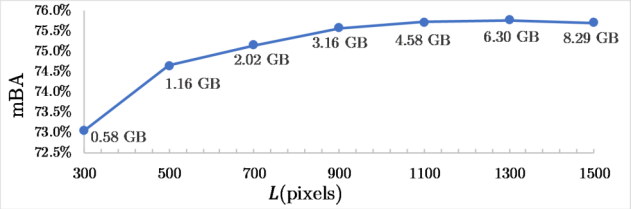


Figure 6. Relationship between the choice of L and mBA in the BIG validation set. With increasing L, GPU memory usage increases with diminishing performance gain.

3.2.3 Choosing L

Figure 6 presents the relationship between GPU memory usage and refinement quality (mBA) during testing when different L is chosen. We have chosen $L = 900$ with 3.16 GB of GPU memory usage in our experiments to balance the tradeoff between increasing GPU memory usage and diminishing performance gain. Using an even higher L is unnecessary and occupies extra memory in our experiments on the BIG validation set. In low-memory settings, a smaller L such as 500 can be used to produce a slightly worse (-0.6% mBA) refinement with a much lower memory footprint (1.16 GB). Note that the GPU memory usage only relates to L but not the image resolution as the fusion step can be easily performed on the CPU.

3.2.4 Ablation Study for Global and Local Refinement

Table 2 shows that both the Global step and the Local step are essential to high-resolution segmentation refinement. Note that the IoU drop is much more significant when we remove the Global step, indicating that the Global step is mainly responsible for fixing the overall structure contributing more to IoU boost while the Local step alone cannot achieve due to insufficient image context. Without the Local step, although IoU only decreases slightly, we note that boundary accuracy decreases more significantly since the Global step cannot extract high-resolution details.

Figure 7 studies the importance of the Local step for different resolution inputs: we evaluate our method with and without the Local step in various-sized segmentations generated by resizing the BIG validation set. While the Global step is sufficient for low-resolution inputs, the Local step is crucial for accurate high-resolution refinement with size higher than the switching point 900. We therefore use both the Global and Local step for inputs with $\max(H, W) \geq 900$, and only the Global step for lower resolution inputs.

Configuration	BIG	
	IoU (%)	mBA (%)
Deeplab V3+	89.65	60.94
Global step only	$91.86_{\uparrow 2.21}$	$73.10_{\uparrow 12.2}$
Local step only	$91.35_{\uparrow 1.70}$	$73.06_{\uparrow 12.1}$
Both steps (Ours)	$\mathbf{92.01}_{\uparrow \mathbf{2.36}}$	$\mathbf{75.59}_{\uparrow \mathbf{14.7}}$

Table 2. Ablation experiments for the Global and Local step. Input segmentations are taken from DeepLab V3+ on the BIG validation set. Using both steps show the best results.

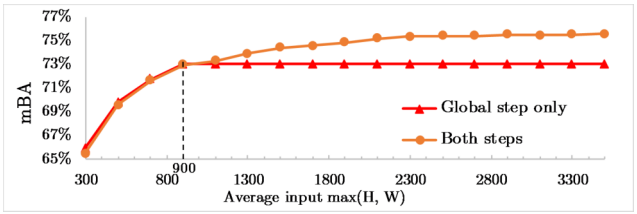


Figure 7. We evaluated our method across different input resolutions. The Global step does not benefit from higher resolution inputs because it is bounded by $L = 900$. The Local step is crucial for high-resolution refinement to handle inputs with a larger size than L with bounded memory cost.

3.3. Training

To learn objectness information, we train our model on a collection of datasets in a class-agnostic manner. We merge MSRA-10K [8], DUT-OMRON [48], ECSSD [38], and FSS-1000 [44] to generate a segmentation dataset of 36,572 with much more diverse semantic classes than common datasets such as PASCAL (20 classes) or COCO (80 classes). Using this dataset (> 1000 classes) makes our model more robust and generalizable to new classes.

During training, we take random 224×224 image crops and generate input segmentations by perturbing the ground truth. The inputs go through a 3-level cascade as in the Global step with the loss computed in every level. Although the crop size is smaller than L which is used in testing, our model design helps bridging this gap. The fully convolutional feature extractor provides translational invariance while the pyramid pooling module provides important image context, allowing our model to be extended to higher resolution without significant performance loss. The use of smaller crop speeds up our training process and makes data preparation much easier as high-resolution training data for segmentation is expensive to obtain.

For generalizability, we avoid training using segmentation outputs generated by existing models which can lead to overfitting to that specific model. Instead, perturbed ground truth should portray various shapes and output of inaccurate segmentations produced by other methods, which helps our algorithm to be more robust to different initial segmentations. We generate such perturbed segmentations by subsampling the contour followed by random dilations and erosions. Examples of such perturbation are shown in Figure 8.

4. Experiments

In this section, we quantitatively evaluate our results with PASCAL VOC 2012 [13], BIG (our high-resolution data set), and ADE20K [55]. We evaluate our model *without any finetuning* in various settings and show the improvements made by our model.

4.1. Dataset and Evaluation Method

Although widely used in image segmentation tasks, PASCAL VOC 2012 dataset does not have pixel-perfect

Figure 8. **Blue**: Ground truth labels of FSS-1000 [44]. **Red**: Perturbed labels that we use as inputs to train our model.

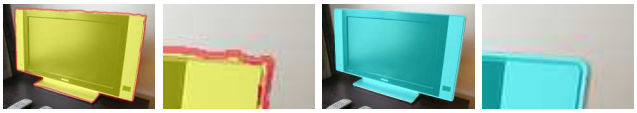


Figure 9. Segmentation results in the PASCAL VOC 2012 validation set. **Left**: An example ground truth label of PASCAL VOC 2012. Red line shows the void boundary label. **Right**: Relabeled segmentation for the same image.

segmentations and the areas near the boundary are labeled as "void". For a more accurate evaluation, we have relabeled 500 segmentations from the PASCAL VOC 2012 validation set, so that the accurate boundary can be found within the void boundary regions. Figure 9 shows a relabeled example.

The lack of a high-resolution image segmentation dataset is one of the difficulties of evaluating an image segmentation model in high-resolution. To solve this issue, we present the BIG dataset, a high-resolution semantic segmentation dataset with 50 validation and 100 test objects. Image resolution in BIG ranges from 2048×1600 to 5000×3600. Every image in the dataset has been carefully labeled by a professional while keeping the same guidelines as PASCAL VOC 2012 without the void region. Both the relabeled PASCAL validation set and the BIG dataset are available on our project website. Other datasets used in the evaluation are not modified. We evaluate our method using standard segmentation metric IoU and our boundary metric mBA.

4.2. Implementation Details

We implement our model with PyTorch [34]. We use PSPNet with ResNet-50 backbone [53] as our base network. Data augmentations, including perturbation of ground truths, image flipping and cropping are done on-the-fly to further increase data variety. We use Adam optimizer [19] with a weight decay of 10^{-4}, learning rate of 3×10^{-4} for 30K iterations followed by a learning rate of 3×10^{-5} for another 30K iterations with a batch size of 9. The total training time is around 16h with two 1080Ti. The Local step is only performed in the region of interest, and the complete refinement process takes about 6.6s for Figure 1. Unless otherwise specified, we use the same trained model for all the experiments.

4.3. Segmentation Input

Our method can refine input segmentations using only objectness information. Note that our model has *never* seen any of the following datasets in training. In this section, we focus on the refinement effect of individual objects, meaning that class competition is not introduced.

Here, we compare and evaluate the effect of our refinement model on the output of various semantic segmentation models [6, 53] trained on the PASCAL VOC 2012 dataset. Our method is more effective than commonly used multi-scale testing, with experimental results further shown in the supplementary material.

PASCAL VOC 2012 As the input models are trained in the PASCAL VOC 2012 dataset, resizing is not needed to obtain their outputs which are then fed into our refinement model. These images are of low resolution, so we can refine them directly using the Global step only. We report the overall class-agnostic IoU and boundary accuracy in the upper half of Table 3. Results show that our method can improve segmentation quality in all cases, especially along the boundary region.

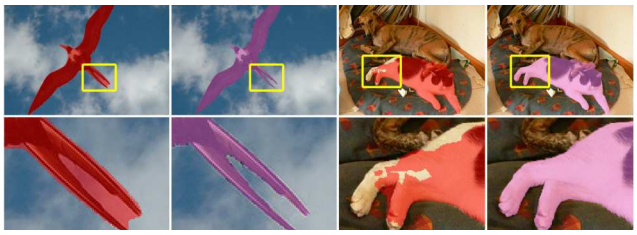


Figure 10. **Red**: Output produced by Deeplab V3+. **Purple**: Segmentation refined by our algorithm.

Methods	IoU (%)	mBA (%)
PASCAL VOC 2012		
FCN-8s [31]	68.85	54.05
(+) Ours	**72.70**$_{\uparrow 3.85}$	**65.36**$_{\uparrow 11.3}$
RefineNet [22]	86.21	62.61
(+) Ours	**87.48**$_{\uparrow 1.27}$	**71.34**$_{\uparrow 8.73}$
DeepLabV3+ [6]	87.13	61.68
(+) Ours	**89.01**$_{\uparrow 1.88}$	**72.10**$_{\uparrow 10.4}$
PSPNet [53]	90.92	60.51
(+) Ours	**92.86**$_{\uparrow 1.94}$	**72.24**$_{\uparrow 11.7}$
BIG		
FCN-8s [31]	72.39	53.63
(+) Ours	**77.87**$_{\uparrow 5.48}$	**67.04**$_{\uparrow 13.4}$
RefineNet [22]	90.20	62.03
(+) Ours	**92.79**$_{\uparrow 2.59}$	**74.77**$_{\uparrow 12.7}$
DeepLabV3+ [6]	89.42	60.25
(+) Ours	**92.23**$_{\uparrow 2.81}$	**74.59**$_{\uparrow 14.3}$
PSPNet [53]	90.49	59.63
(+) Ours	**93.93**$_{\uparrow 3.44}$	**75.32**$_{\uparrow 15.7}$

Table 3. Comparison between different semantic segmentation methods with and without our refinement. Their results are produced using their respective official implementations with the best provided model. Low-resolution outputs from the original model are bicubic-upsampled to the original resolution for evaluation.

BIG dataset Most existing segmentation methods cannot be directly evaluated on the full-resolution BIG dataset,

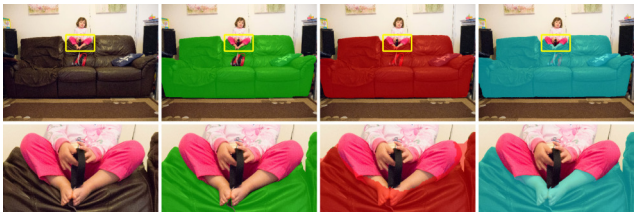


Image **GT** **DeeplabV3+** **Ours**

Figure 11. A failure case of our method. DeeplabV3+ incorrectly labels a large region of the feet as foreground. Although our refinement still adheres well to the color boundary, it produces a wrong segmentation due to the lack of semantic information.

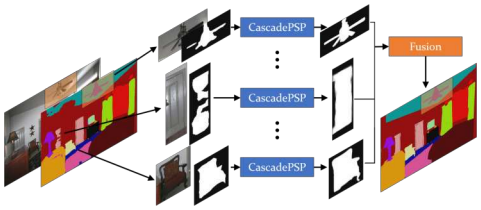


Figure 12. Divide-and-conquer strategy in applying CascadePSP to scene parsing.

due to the memory constraint. Therefore, we obtained initial segmentations by feeding resized images to the existing models. We downsampled the input image such that the long-axis is 512-pixel while maintaining the aspect ratio, and bicubic-upsampled the output segmentation to the original resolution.

In the lower half of Table 3, we show our results on the BIG test set with high-resolution segmentation. Note that even we have never seen any high-resolution training images, we are able to produce high-quality refinements at these scales. Figure 14 shows the visual improvement of our refinement. Although super-resolution models [11, 43] may seem plausible for upsampling segmentation masks, inputs with erroneous segmentation (*e.g.* the missing table leg in Figure 13 and the baby's hand in Figure 14) cannot be corrected by super-resolution.

Our method relies on the input segmentation and low-level cues and does not have the specific semantic capability. Figure 11 shows one failure case where the input error is too large for our method to eliminate.

4.4. Scene parsing

To extend CascadePSP to scene parsing, in the presence of dense classes where class competition may be problematic, we propose a divide-and-conquer approach to independently refine each semantic object using our pretrained network, followed by integrating the results using a fusion function. Figure 12 overviews our strategy.

We refine sufficiently large connected components for each semantic object independently by taking ROIs with 25% padding. To handle overlapping regions, the naïve approach would be to use argmax on the output confidence which would lead to noisy results in regions where all the classes have low scores. Instead, our fusion function is a modified argmax where if all the input class confidence have values lower than 0.5, we fall back to the original segmentation.

Methods	mIoU (%)	mBA (%)
ADE20K		
RefineNet [22]	41.47	55.60
(+) Ours	$\mathbf{42.20}_{\uparrow \mathbf{0.73}}$	$\mathbf{56.67}_{\uparrow \mathbf{1.07}}$
EncNet [51]	42.20	55.29
(+) Ours	$\mathbf{43.19}_{\uparrow \mathbf{0.99}}$	$\mathbf{57.29}_{\uparrow \mathbf{2.00}}$
PSPNet [53]	43.10	57.03
(+) Ours	$\mathbf{43.83}_{\uparrow \mathbf{0.73}}$	$\mathbf{58.13}_{\uparrow \mathbf{1.10}}$

Table 4. Comparison between different methods with and without our refinement on the ADE20K validation set.

Here, we evaluate our model on the validation set of ADE20K [55]. As the ADE20K dataset contains "stuff" background classes (see supplementary material) that are not strong in objectness and too different from our training data, we have attenuated their output scores to focus on foreground refinement. Note that refining the foreground objects can still help with background refinement since the argmax operation takes both confidence scores into consideration. Table 4 tabulates the results which show that our model produces higher quality segmentation. Figure 13 shows sample qualitative evaluation.

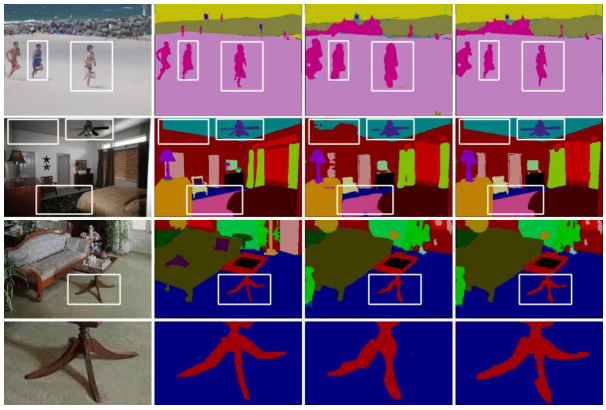


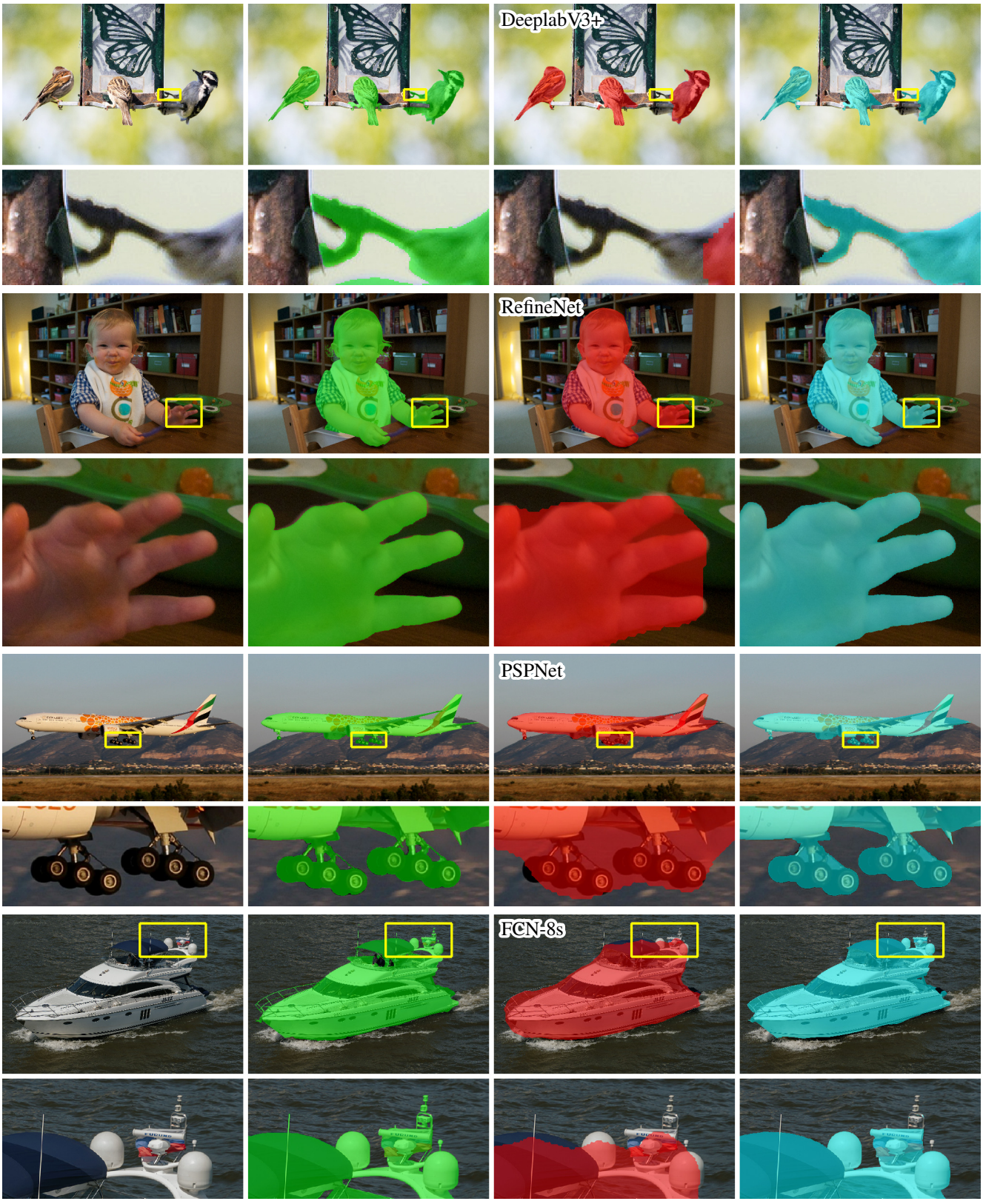
Image **GT** **Input** **Ours**

Figure 13. Refinement results in the ADE20K validation set. Top two rows: PSPNet. Bottom two rows: RefineNet.

5. Conclusion

We propose CascadePSP, a general segmentation refinement framework for refining any input segmentations and achieve a higher accuracy without any finetuning afterward. CascadePSP performs high-resolution (up to 4K) segmentation refinement even our model has *never* seen any high-resolution training images. With a single refinement module trained on low-resolution data without any finetuning, the proposed Global step refines the entire image and provides sufficient image context for the subsequent Local step to perform full-resolution high-quality refinement. We hope this work can contribute to more high-resolution computer vision tasks in the future.

Acknowledgements We thank Gary Jing Yang Zhang for fruitful discussion during his exchange semester at HKUST.



(a) Image **(b) Ground Truth** **(c) Input** **(d) Our Refinement**

Figure 14. Visual comparison on the BIG test set. Odd rows show the whole image and even rows show the zoomed-in crop. Inputs are from DeeplabV3+, RefineNet, PSPNet, and FCN-8s, top to bottom.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *TPAMI*, 2017.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

[3] 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. *TPAMI*, 2017.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[5] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016.

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

[7] Wuyang Chen, Ziyu Jiang, Zhangyang Wang, Kexin Cui, and Xiaoning Qian. Collaborative global-local networks for memory-efficient segmentation of ultra-high resolution images. In *CVPR*, 2019.

[8] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip Hilaire Sean Torr, and Shi-Min Hu. Global contrast based salient region detection. *TPAMI*, 2015.

[9] Jifeng Dai, Kaiming He, and Jian Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, 2015.

[10] Philipe Ambrozio Dias and Henry Medeiros. Semantic segmentation refinement by monte carlo region growing of high confidence detections. In *ACCV*, 2018.

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. In *PAMI*, 2015.

[12] Michal Drozdzal, Eugene Vorontsov, Gabriel Chartrand, Samuel Kadoury, and Chris Pal. The importance of skip connections in biomedical image segmentation. In *Deep Learning and Data Labeling for Medical Applications*, 2016.

[13] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge – a retrospective. In *IJCV*, 2014.

[14] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *TPAMI*, 2012.

[15] Jianzhong He, Shiliang Zhang, Ming Yang, Yanhu Shan, and Tiejun Huang. Bi-directional cascade network for perceptual edge detection. In *CVPR*, 2019.

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

[17] Xuming He, Richard S Zemel, and Miguel Á Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *CVPR*, 2004.

[18] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 1988.

[19] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.

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

[21] Hanchao Li, Pengfei Xiong, Jie An, and Lingxue Wang. Pyramid attention network for semantic segmentation. In *BMVC*, 2018.

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

[23] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016.

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

[25] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *CVPR*, 2019.

[26] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. In *NIPS*, 2017.

[27] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018.

[28] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[29] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. In *ICLR*, 2016.

[30] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015.

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

[32] Mohammadreza Mostajabi, Payman Yadollahpour, and Gregory Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *CVPR*, 2015.

[33] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015.

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

[35] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *CVPR*, 2017.

[36] Pedro HO Pinheiro and Ronan Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, 2014.

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[38] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *TPAMI*, 2015.

[39] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. In *IJCV*, 2009.

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

[41] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *CVPR*, 2013.

[42] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. In *WACV*, 2018.

[43] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCV Workshop*, 2018.

[44] Tianhan Wei, Xiang Li, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. FSS-1000: A 1000-class dataset for few-shot segmentation. *CoRR*, abs/1907.12347, 2019.

[45] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *ICCV*, 2015.

[46] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *CVPR*, 2017.

[47] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas Huang. Deep grabcut for object selection. In *BMVC*, 2017.

[48] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, 2013.

[49] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[50] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *CVPR*, 2019.

[51] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018.

[52] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *ECCV*, 2018.

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

[54] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip Hilaire Sean Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015.

[55] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.