ICNet for Real-Time Semantic Segmentation on High-Resolution Images — Supplementary Material

Hengshuang Zhao[1], Xiaojuan Qi[1], Xiaoyong Shen[2], Jianping Shi[3], Jiaya Jia[1,2]

[1]The Chinese University of Hong Kong, [2] Tencent Youtu Lab, [3]SenseTime Research
{hszhao,xjqi,leojia}@cse.cuhk.edu.hk,
dylanshen@tencent.com, shijianping@sensetime.com

1 Detailed Network Structure

Here we detail the network structure of ICNet including three branches with inputs of different image resolutions, *i.e.*, low-, medium- and high resolutions.

Lowest-resolution Branch We denote the size of the full-resolution input as $H \times W$. Thus the input to the lowest branch is of size $H/4 \times W/4$. It is fed to the top branch in Fig. 1, which is an FCN-based architecture with dilated convolution. We adopt PSPNet framework which incorporates dilated convolutions with dilations set to 2 and 4 in stage4 and stage5 of ResNet respectively, thus get downsampled feature map with size 1/8 of the input. The resulting $C \times H/32 \times W/32$ feature map after pyramid pooling module in PSPNet is obtained where C is the output channel size (e.g., 4096). For high efficiency information incorporation with higher-level branches, a reduction convolution is adopted with kernel size $C' \times 1 \times 1$ to reduce the channel dimension from C to C', where C' is much smaller than C (e.g., C'=256). The output of the low-resolution branch is $C' \times H/32 \times W/32$.

Medium-resolution Branch For the 1/2-size input, it is processed in the middle branch in Fig. 1. There are multiple convolution layers in this branch that can be divided into three stages, each with downsampling rate 2 and forming the output feature map downsampled by factor 8 as $H/16 \times W/16$. Compared with the low-resolution branch, this level is important to recover details missing in lower resolution. Thus we share weights and computation of downsampling stages between these two branches. Taking PSPNet50 as an example, 17 convolutional layers in the first three stages (i.e., stage1, stage2 and stage3) are shared, i.e., shrinking the feature map in med-branch by 2 and feeding it to the rest of top branch. In the end, the cascade feature fusion (CFF) unit fuses this 1/16-scale output together with the 1/32 feature map from the lower-res branch, resulting in $H/16 \times W/16$ resolution output feature map.

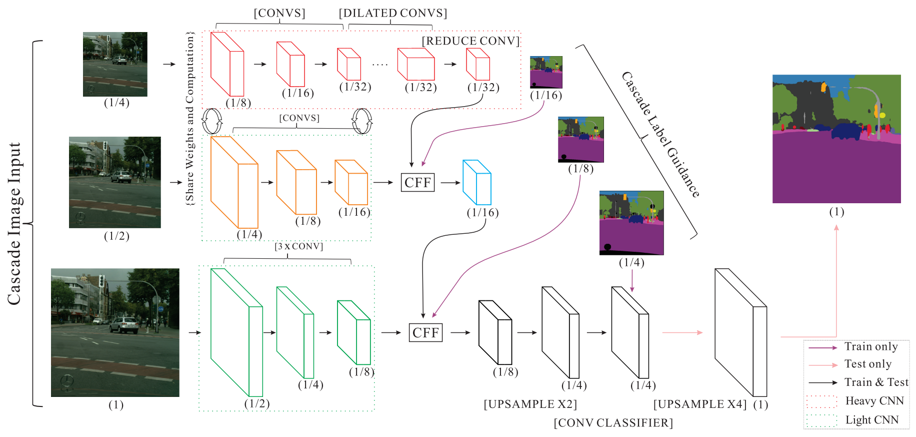

Fig. 1: Network architecture of ICNet. Numbers in parentheses are feature map size ratios to the full-resolution input. Operations are highlighted in brackets. The final $\times 4$ upsampling in the bottom branch is only used during testing.

High-resolution Branch For the full-resolution input in bottom branch of Fig. 1, it is processed by several convolutions with downsampling rate 8, similar to above operations. This results in a $H/8 \times W/8$ sized feature map. Since the med-resolution image already restores most semantically meaningful details, it is safe to limit the number of convolutions when processing the high-res input. Here we use only three convolutional layers each with kernel size 3×3 and stride 2 to downsize the resolution to 1/8 of the original input. Although the resolution is high, the inference time is only 9ms from this branch. Similar to the fusion strategy mentioned above, we fuse the output of two neighboring higher-res branches, resulting in a $H/8 \times W/8$ output. The 1/8 sized output in the high-res branch is upsampled by 2 through bilinear interpolation, resulting 1/4 sized feature map. Then a per pixel convolutional classifier with 1×1 kernel is utilized to get the final prediction. This prediction is trained with 1/4 sized label guidance. And during testing, we directly upsample the 1/4 sized predicted score map by 4 through bilinear interpolation to get the full sized prediction as illustrated in the bottom right part of Fig. 1.

2 Other Combinations and Segmentation Backbone

ICNet can be applied to various backbones. ICNet with different feature downsampling factors and kernel keeping rates (e.g. 8–0.5 adopted in our paper) is evaluated in Table 1. We have also experimented with using other segmentation framework like DeepLabv3 [1] as ICNet backbone and the results are shown in Table 2. In all ablated settings, ICNet can improve efficiency 5× times on average without sacrificing much accuracy. It is thus a very general framework.

Table 1: Performance of ICNet on PSPNet50 backbone with different combinations.

Architecture	Featrue downsampling size – Kernel keeping rate											
	8–1				16-1				16-0.5			
	Baseline	sub4	sub24	sub124	Baseline	sub4	sub24	sub124	Baseline	sub4	sub24	sub124
mIoU (%)	71.71	66.70	70.37	71.40	70.17	61.59	69.14	70.48	67.54	59.45	65.96	66.95
Time (ms)	446	64	67	92	177	41	45	68	106	22	24	32
Frame (fps)	2.2	15.6	14.9	10.9	5.6	24.4	22.2	14.7	9.4	45.5	41.7	31.3
Speedup	1×	7.1×	6.8×	5.0×	1×	4.4×	4.0×	2.6×	1×	4.8×	4.4×	3.3×

Table 2: Performance of ICNet with segmentation framework DeepLabv3 (based on ResNet50 without compression) as backbone architecture.

Items	Baseline	sub4	sub24	sub124
mIoU (%)	72.88	66.87	70.06	71.04
Time (ms)	800	80	83	107
Frame (fps)	1.3	12.5	12.0	9.3
Speedup	1×	9.6×	9.2×	7.2×

3 Detailed Time of Several Approaches

The inference mIoU and time are highly correlated. While we are not able to normalize different approaches for variance of deep learning framework, hardware platform and testing tricks. Here we list detailed information of several approaches in Table 3 for readers as reference. We thank corresponding authors for providing the related information. Inference speed is reported with single network forward while accuracy of several mIoU aimed approaches may contain testing tricks like multi-scale and flipping, resulting much more time. Note that the image/feature scaling in our Caffe platform is achieved through a CPU version of 'Interp' layer, which may effect the inference speed a little.

4 More Visual Results

We include more visual prediction improvements of ICNet in Fig. 2 and Fig. 3. And show the visual comparisons of ICNet on CamVid and COCO-Stuff datasets in Fig. 4 and Fig. 5.

Table 3: Detailed time of several approaches.

Method	DR	mIoU	Time (ms)	Frame (fps)	Platform	GPU
FRRN [2]	2	71.8	469	2.1	Theano	TitanX Pascal
RefineNet [3]	no	73.6	2517	0.40	MatConvNet	TitanX Pascal
RefineNet [3]	no	73.6	1174	0.85	MxNet	TitanX Pascal
DUC [4]	no	80.1	879	1.14	MxNet	1080Ti
ResNet38 [5]	no	80.6	1150	0.87	MxNet	TitanX Pascal
PSPNet [6]	no	81.2	1288	0.78	Caffe	TitanX Maxwell
PSPNet [6]	no	81.2	680	1.5	Caffe	TitanX Pascal
PSPNet* [6]	no	81.2	51.0×1e3	0.02	Caffe	TitanX Maxwell
PSPNet* [6]	no	81.2	26.7×1e3	0.04	Caffe	TitanX Pascal
ICNet	no	70.6	33	30.3	Caffe	TitanX Maxwell
ICNet	no	70.6	24	41.7	Caffe	TitanX Pascal

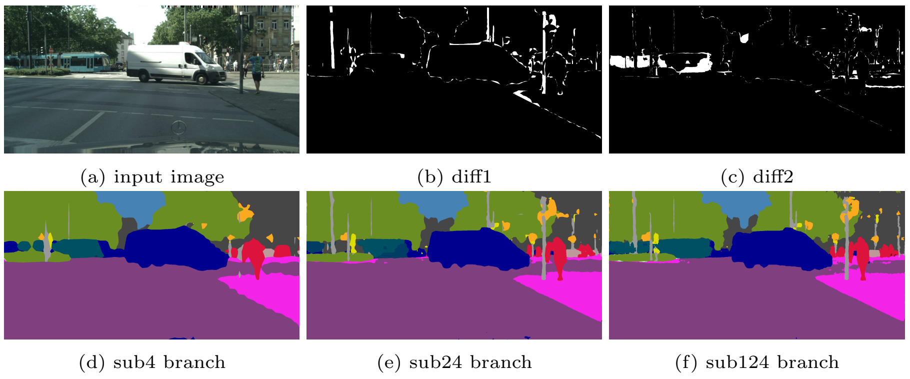
(a) input image (b) diff1 (c) diff2
(d) sub4 branch (e) sub24 branch (f) sub124 branch

Fig. 2: Visual prediction improvement of ICNet.

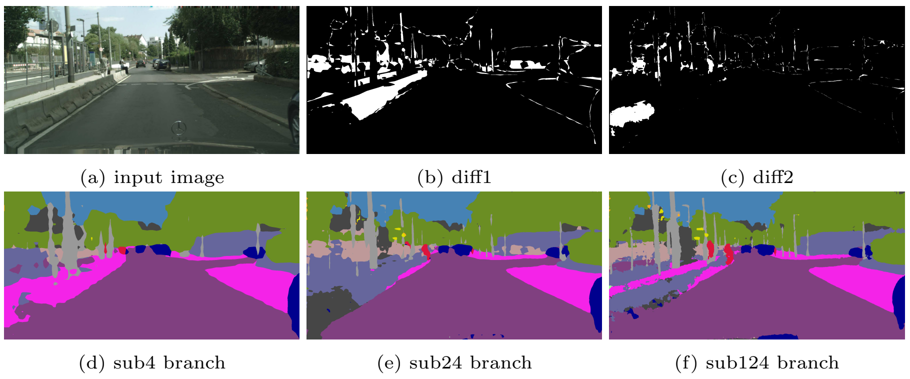
(a) input image (b) diff1 (c) diff2
(d) sub4 branch (e) sub24 branch (f) sub124 branch

Fig. 3: Visual prediction improvement of ICNet.

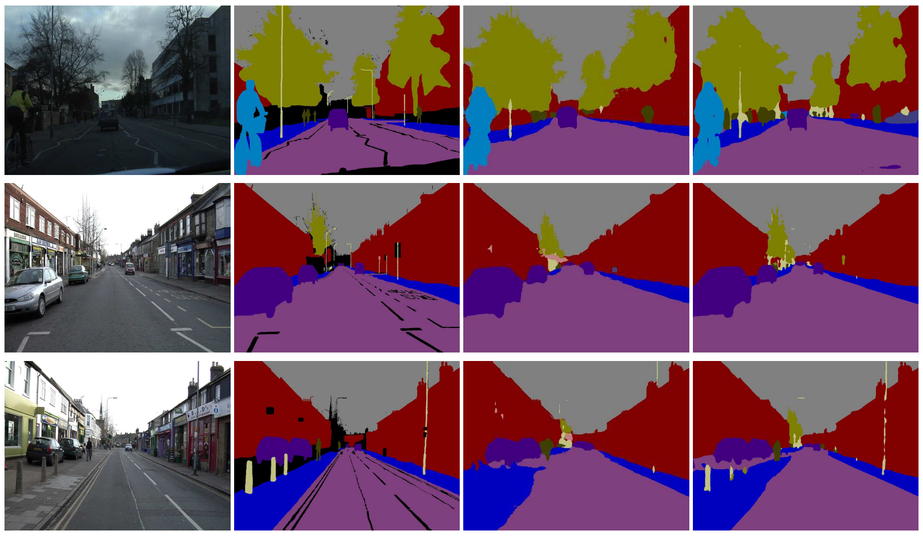
(a) Image (b) Ground Truth (c) DPN (d) ICNet

Fig. 4: Visual comparison of ICNet on CamVid dataset.

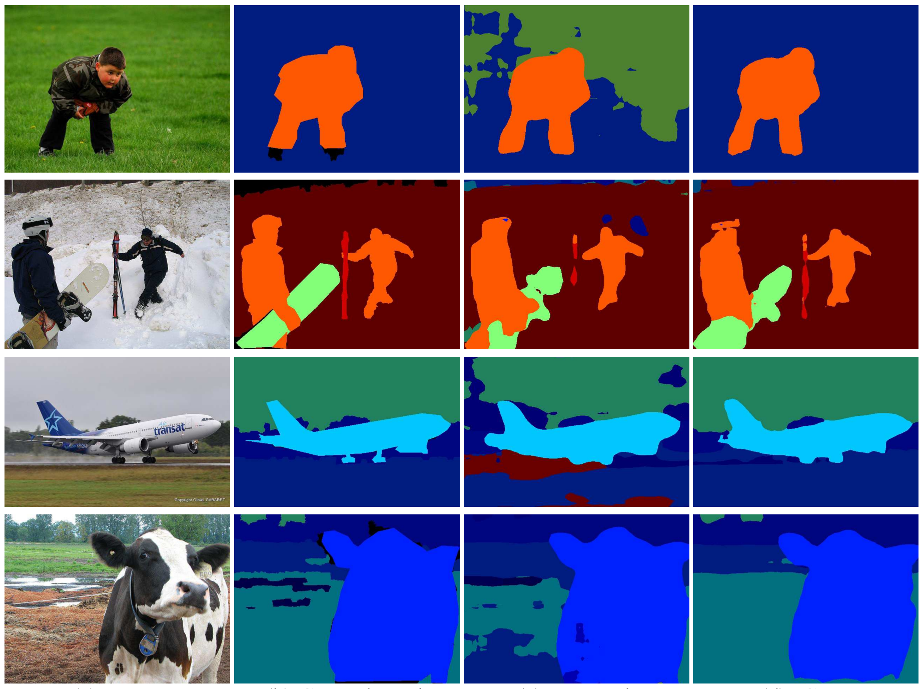
(a) Image (b) Ground Truth (c) DeepLab (d) ICNet

Fig. 5: Visual comparison of ICNet on COCO-Stuff dataset.

References

1. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv:1706.05587 (2017)
2. Pohlen, T., Hermans, A., Mathias, M., Leibe, B.: Full-resolution residual networks for semantic segmentation in street scenes. In: CVPR. (2017)
3. Lin, G., Milan, A., Shen, C., Reid, I.D.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: CVPR. (2017)
4. Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., Cottrell, G.W.: Understanding convolution for semantic segmentation. In: WACV. (2018)
5. Wu, Z., Shen, C., van den Hengel, A.: Wider or deeper: Revisiting the resnet model for visual recognition. arXiv:1611.10080 (2016)
6. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR. (2017)