RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation

– Author: Guosheng Lin, Anton Milan, Chunhua Shen, Ian Reid
– Published: CVPR 2017

Jialin Yuan
Date: May. 8 2017
Content

2. Existing methods to obtain High-Resolution Semantic Segmentation.
4. Experiments.
5. Ablation Experiments.
1. Why need Refinement Process?

1. Many Computer Vision task needs dense prediction result.

2. Because CNNs' methods integrates repeated sub-sampling operations, like pooling, the outputs are in low-resolution.

Example: FCN-32s, if input is 32x32, output would be 1x1.

2. Existing methods to obtain High-Resolution Result:

1. FCN-8s.

- In-network up-sampling is fast and effective.

- It only use one deconvolution layer for each skip fusion, makes it difficult to reconstruct highly non-linear structures of object boundaries accurately.

2. Existing methods to obtain High-Resolution Result:

2. Deconvolution

- It up-sample the low-resolution predictions by learning deconvolution layers.
- It can not recover the low-level visual features, which had been lost after down-sampling operation.

2. Existing methods to obtain High-Resolution Result:

3. Dilated Convolution.

- It could obtained large receptive field without losing resolution.

- It is computational expensive considering of its large number of high-dimensional and high-resolution feature maps.

2. Existing methods to obtain High-Resolution Result:

4. interpolation + CRF

– It refine boundaries by leveraging color contrast information.

3. The Proposed RefineNet Algorithm:

- Make use of both low layer, middle layer, high layer information.
- This is one possible arrangement of RefineNet Units, it's called as 4-cascaded.

3. The Proposed RefineNet Algorithm:

- **RCU:** fine-tune feature map for fusion task.

- **Multi-resolution Fusion:** scale feature maps from different layers to the same resolution and summarize them.

- **CRP:** capture background context from a large image region.

- **OC:** another RCU.

** There are 3 RCU in each unit (in this case).

** Residual connections: enables effective BP and facilitate end-to-end training.
4. Experiments:

6 Semantic Segmentation Dataset:
NYUDv2, PASCAL-VOC 2012, SUN-RGBD, PASCAL-Context, Cityscapes, ADE20K MIT

4. Experiments:

Figure 5. Our prediction examples on VOC 2012 dataset.

Figure 6. Our prediction examples on Cityscapes dataset.
4. Experiments:

1 Object Parsing Dataset: Person-Part

    Training: 1717 images (augment it with scale / crop / horizontal flip).

    Testing: 1818 images.

<table>
<thead>
<tr>
<th>Method</th>
<th>IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attention [7]</td>
<td>56.4</td>
</tr>
<tr>
<td>HAZN [45]</td>
<td>57.5</td>
</tr>
<tr>
<td>LG-LSTM [29]</td>
<td>58.0</td>
</tr>
<tr>
<td>Graph-LSTM [28]</td>
<td>60.2</td>
</tr>
<tr>
<td>DeepLab [3]</td>
<td>62.8</td>
</tr>
<tr>
<td>DeepLab-v2 (Res101) [6]</td>
<td>64.9</td>
</tr>
<tr>
<td>RefineNet-Res101 (ours)</td>
<td>68.6</td>
</tr>
</tbody>
</table>

Table 1. Object parsing results on the Person-Part dataset. Our method achieves the best performance (bold).

Figure 4. Our prediction examples on Person-Parts dataset.
5. Ablation Experiment:

1. to qualify the influence of Network depth, chained residual pooling, and multi-scale evaluation.

<table>
<thead>
<tr>
<th>Initialization</th>
<th>Chained pool.</th>
<th>Msc Eva</th>
<th>NYUDv2</th>
<th>Person-Parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-30</td>
<td>no</td>
<td>no</td>
<td>40.4</td>
<td>64.1</td>
</tr>
<tr>
<td>ResNet-30</td>
<td>yes</td>
<td>no</td>
<td>42.5</td>
<td>65.7</td>
</tr>
<tr>
<td>ResNet-30</td>
<td>yes</td>
<td>yes</td>
<td>43.8</td>
<td>67.1</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>yes</td>
<td>no</td>
<td>43.6</td>
<td>67.6</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>yes</td>
<td>yes</td>
<td>44.7</td>
<td>68.6</td>
</tr>
<tr>
<td>ResNet-152</td>
<td>yes</td>
<td>yes</td>
<td>46.5</td>
<td>68.8</td>
</tr>
</tbody>
</table>

2. to qualify several variants of the RefineNet.

Table 9. Evaluations of 4 variants of cascaded RefineNet: single RefineNet, 2-cascaded RefineNet, 4-cascaded RefineNet, 4-cascaded RefineNet with 2-scale RefineNet on the NYUDv2 dataset. We use the 4-cascaded version as our main architecture throughout all experiments in the paper because this turns out to be the best compromise between accuracy and efficiency.

<table>
<thead>
<tr>
<th>Variant</th>
<th>Initialization</th>
<th>Msc Eva</th>
<th>IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>single RefineNet</td>
<td>ResNet-50</td>
<td>no</td>
<td>40.3</td>
</tr>
<tr>
<td>2-cascaded RefineNet</td>
<td>ResNet-50</td>
<td>no</td>
<td>40.9</td>
</tr>
<tr>
<td>4-cascaded RefineNet</td>
<td>ResNet-50</td>
<td>no</td>
<td>42.5</td>
</tr>
<tr>
<td>4-cascaded 2-scale RefineNet</td>
<td>ResNet-50</td>
<td>no</td>
<td>43.1</td>
</tr>
</tbody>
</table>
Thank you!