Fully Convolutional Networks for Semantic Segmentation

Jonathan Long*    Evan Shelhamer*    Trevor Darrell

UC Berkeley

Presented by: Gordon Christie

Slide credit: Jonathan Long
Overview

• Reinterpret standard classification convnets as “Fully convolutional” networks (FCN) for semantic segmentation
• Use AlexNet, VGG, and GoogleNet in experiments
• Novel architecture: combine information from different layers for segmentation
• State-of-the-art segmentation for PASCAL VOC 2011/2012, NYUDv2, and SIFT Flow at the time
• Inference less than one fifth of a second for a typical image
pixels in, pixels out

monocular depth estimation (Liu et al. 2015)

semantic segmentation

boundary prediction (Xie & Tu 2015)

Slide credit: Jonathan Long
convnets perform classification

< 1 millisecond

1000-dim vector

end-to-end learning

“tabby cat”

Slide credit: Jonathan Long
R-CNN does detection

many seconds

R-CNN

“dog”

“cat”
R-CNN

1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

figure: Girshick et al.
Slide credit: Jonathan Long
< 1/5 second

end-to-end learning

Slide credit: Jonathan Long
a classification network

Slide credit: Jonathan Long
becoming fully convolutional

Slide credit: Jonathan Long
becoming fully convolutional
upsampling output

Slide credit: Jonathan Long
end to end, pixels to pixels network

conv, pool, nonlinearity

upsampling

pixelwise output + loss

Slide credit: Jonathan Long
Dense Predictions

- Shift-and-stitch: trick that yields dense predictions without interpolation
- Upsampling via deconvolution
- Shift-and-stitch used in preliminary experiments, but not included in final model
- Upsampling found to be more effective and efficient
Classifier to Dense FCN

- Convolutionalize proven classification architectures: AlexNet, VGG, and GoogLeNet (reimplementation)
- Remove classification layer and convert all fully connected layers to convolutions
- Append 1x1 convolution with channel dimensions and predict scores at each of the coarse output locations (21 categories + background for PASCAL)
## Classifier to Dense FCN

Cast ILSVRC classifiers into FCNs and compare performance on validation set of PASCAL 2011

<table>
<thead>
<tr>
<th></th>
<th>FCN-AlexNet</th>
<th>FCN-VGG16</th>
<th>FCN-GoogLeNet$^4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean IU</td>
<td>39.8</td>
<td><strong>56.0</strong></td>
<td>42.5</td>
</tr>
<tr>
<td>forward time</td>
<td>50 ms</td>
<td>210 ms</td>
<td>59 ms</td>
</tr>
<tr>
<td>conv. layers</td>
<td>8</td>
<td>16</td>
<td>22</td>
</tr>
<tr>
<td>parameters</td>
<td>57M</td>
<td>134M</td>
<td>6M</td>
</tr>
<tr>
<td>rf size</td>
<td>355</td>
<td>404</td>
<td>907</td>
</tr>
<tr>
<td>max stride</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
</tbody>
</table>
spectrum of deep features

combine *where* (local, shallow) with *what* (global, deep)

Slide credit: Jonathan Long
skip layers

end-to-end, joint learning of semantics and location

Slide credit: Jonathan Long
skip layers

Figure 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Layers are shown as grids that reveal relative spatial coarseness. Only pooling and prediction layers are shown; intermediate convolution layers (including our converted fully connected layers) are omitted. Solid line (FCN-32s): Our single-stream net, described in Section 4.1, upsamples stride 32 predictions back to pixels in a single step. Dashed line (FCN-16s): Combining predictions from both the final layer and the pool4 layer, at stride 16, lets our net predict finer details, while retaining high-level semantic information. Dotted line (FCN-8s): Additional predictions from pool3, at stride 8, provide further precision.
Comparison of skip FCNs

Results on subset of validation set of PASCAL VOC 2011

<table>
<thead>
<tr>
<th></th>
<th>pixel acc.</th>
<th>mean acc.</th>
<th>mean IU</th>
<th>f.w. IU</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCN-32s-fixed</td>
<td>83.0</td>
<td>59.7</td>
<td>45.4</td>
<td>72.0</td>
</tr>
<tr>
<td>FCN-32s</td>
<td>89.1</td>
<td>73.3</td>
<td>59.4</td>
<td>81.4</td>
</tr>
<tr>
<td>FCN-16s</td>
<td>90.0</td>
<td>75.7</td>
<td>62.4</td>
<td>83.0</td>
</tr>
<tr>
<td>FCN-8s</td>
<td>90.3</td>
<td>75.9</td>
<td>62.7</td>
<td>83.2</td>
</tr>
</tbody>
</table>
skip layer refinement

input image  stride 32  stride 16  stride 8  ground truth

no skips  1 skip  2 skips

Slide credit: Jonathan Long
training + testing

- train full image at a time \textit{without patch sampling}
- reshape network to take input of any size
- forward time is \(\sim150\text{ms}\) for \(500 \times 500 \times 21\) output
Results – PASCAL VOC 2011/12

VOC 2011: 8498 training images (from additional labeled data

<table>
<thead>
<tr>
<th></th>
<th>mean IU VOC2011 test</th>
<th>mean IU VOC2012 test</th>
<th>inference time</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-CNN [12]</td>
<td>47.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SDS [16]</td>
<td>52.6</td>
<td>51.6</td>
<td>~ 50 s</td>
</tr>
<tr>
<td>FCN-8s</td>
<td>62.7</td>
<td>62.2</td>
<td>~ 175 ms</td>
</tr>
</tbody>
</table>
Results – NYUDv2

1449 RGB-D images with pixelwise labels → 40 categories

<table>
<thead>
<tr>
<th></th>
<th>pixel acc.</th>
<th>mean acc.</th>
<th>mean IU</th>
<th>f.w. IU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gupta et al. [14]</td>
<td>60.3</td>
<td>-</td>
<td>28.6</td>
<td>47.0</td>
</tr>
<tr>
<td>FCN-32s RGB</td>
<td>60.0</td>
<td>42.2</td>
<td>29.2</td>
<td>43.9</td>
</tr>
<tr>
<td>FCN-32s RGBD</td>
<td>61.5</td>
<td>42.4</td>
<td>30.5</td>
<td>45.5</td>
</tr>
<tr>
<td>FCN-32s HHA</td>
<td>57.1</td>
<td>35.2</td>
<td>24.2</td>
<td>40.4</td>
</tr>
<tr>
<td>FCN-32s RGB-HHA</td>
<td>64.3</td>
<td>44.9</td>
<td>32.8</td>
<td>48.0</td>
</tr>
<tr>
<td>FCN-16s RGB-HHA</td>
<td>65.4</td>
<td>46.1</td>
<td>34.0</td>
<td>49.5</td>
</tr>
</tbody>
</table>
Results – SIFT Flow

2688 images with pixel labels
→ 33 semantic categories, 3 geometric categories
Learn both label spaces jointly
→ learning and inference have similar performance and computation as independent models

<table>
<thead>
<tr>
<th></th>
<th>pixel acc.</th>
<th>mean acc.</th>
<th>mean IU</th>
<th>f.w. IU</th>
<th>geom. acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liu et al. [23]</td>
<td>76.7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Tighe et al. [33]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>90.8</td>
</tr>
<tr>
<td>Tighe et al. [34]1</td>
<td>75.6</td>
<td>41.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Tighe et al. [34]2</td>
<td>78.6</td>
<td>39.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Farabet et al. [8]1</td>
<td>72.3</td>
<td>50.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Farabet et al. [8]2</td>
<td>78.5</td>
<td>29.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Pinheiro et al. [28]</td>
<td>77.7</td>
<td>29.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>FCN-16s</td>
<td><strong>85.2</strong></td>
<td><strong>51.7</strong></td>
<td><strong>39.5</strong></td>
<td><strong>76.1</strong></td>
<td><strong>94.3</strong></td>
</tr>
</tbody>
</table>
Relative to prior state-of-the-art SDS:

- 20% relative improvement for mean IoU
- 286× faster

*Simultaneous Detection and Segmentation
Hariharan et al. ECCV14
== segmentation with Caffe

Slide credit: Jonathan Long
conclusion

fully convolutional networks are fast, end-to-end models for pixelwise problems

- **code** in Caffe branch (merged soon)
- **models** for PASCAL VOC, NYUDv2, SIFT Flow, PASCAL-Context

[Link to Caffe](caffe.berkeleyvision.org)
[Link to github](github.com/BVLC/caffe)

fcn.berkeleyvision.org

Slide credit: Jonathan Long