Semantic Segmentation and Image Processing with Convnets

Lecture 10
pixels in, pixels out

- **semantic Segmentation**
  - Long et al. 2015

- **monocular depth + normals**
  - Eigen & Fergus 2015

- **optical flow**
  - Fischer et al. 2015

- **boundary prediction**
  - Xie & Tu 2015

- **colorization**
  - Zhang et al. 2016
Overview

• Methods where output is also an image
  • Fully Convolutional Nets [Long et al., CVPR 2015]
  • Depth, normals and semantic labels from a single image [Eigen ICCV 2015]

• Image processing with Convnets
A Fuller Understanding of Fully Convolutional Networks

Evan Shelhamer*   Jonathan Long*    Trevor Darrell

UC Berkeley in CVPR'15, PAMI'16
convnets perform classification

< 1 millisecond

“tabby cat”

1000-dim vector

end-to-end learning
lots of pixels, little time?

end-to-end learning

~1/10 second
a classification network

```
227 x 227  55 x 55  27 x 27  13 x 13
```

"tabby cat"
becoming fully convolutional
becoming fully convolutional
upsampling output

convolution

| H × W | H/4 × W/4 | H/8 × W/8 | H/16 × W/16 | H/32 × W/32 | H × W |
end-to-end, pixels-to-pixels network
end-to-end, pixels-to-pixels network

conv, pool, nonlinearity

convolution

upsampling

pixelwise output + loss
spectrum of deep features

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

image

intermediate layers

fuse features into deep jet

(cf. Hariharan et al. CVPR15 “hypercolumn”)
skip layers

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

interp + sum
skip layer refinement

input image

stride 32
no skips

stride 16
1 skip

stride 8
2 skips

ground truth
skip FCN computation

A multi-stream network that fuses features/predictions across layers
Relative to prior state-of-the-art SDS:

- 30% relative improvement for mean IoU
- 286× faster

*Simultaneous Detection and Segmentation
Hariharan et al. ECCV14
SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation

Max pooling indices transferred to decoder to improve output resolution

https://arxiv.org/abs/1511.00561
How to do the Upsampling?

Also known as Deconvolution
See https://distill.pub/2016/deconv-checkerboard/

Avoid artifacts by doing bilinear interpolation
UNet: Convolutional Networks for Biomedical Image Segmentation

Segmentation of a 512x512 image takes less than a second on a recent GPU

https://arxiv.org/abs/1505.04597
Dilated / Atrous Convolutions

[Multi-Scale Context Aggregation by Dilated Convolutions, Yu and Koltun, 2015]

- No pooling operations
- Constant resolution feature maps
- Integrate increasing spatial context by special kind of dilated convolution

<table>
<thead>
<tr>
<th>Layer</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution</td>
<td>3×3</td>
<td>3×3</td>
<td>3×3</td>
<td>3×3</td>
<td>3×3</td>
<td>3×3</td>
<td>3×3</td>
<td>1×1</td>
</tr>
<tr>
<td>Dilation</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Truncation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Receptive field</td>
<td>3×3</td>
<td>5×5</td>
<td>9×9</td>
<td>17×17</td>
<td>33×33</td>
<td>65×65</td>
<td>67×67</td>
<td>67×67</td>
</tr>
<tr>
<td>Output channels</td>
<td>Basic</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>Large</td>
<td>2C</td>
<td>2C</td>
<td>4C</td>
<td>8C</td>
<td>16C</td>
<td>32C</td>
<td>32C</td>
</tr>
</tbody>
</table>

- Constant 64x64 spatial resolution throughout
Dilated / Atrous Convolutions

[Multiscale Context Aggregation by Dilated Convolutions, Yu and Koltun, 2015]

<table>
<thead>
<tr>
<th></th>
<th>FCN-8s</th>
<th>DeepLab</th>
<th>DeepLab-Msc</th>
<th>Our front end</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean IoU</td>
<td>76.8</td>
<td>72.0</td>
<td>74.9</td>
<td>82.2</td>
</tr>
<tr>
<td>aero</td>
<td>34.2</td>
<td>31.0</td>
<td>34.1</td>
<td>37.4</td>
</tr>
<tr>
<td>bike</td>
<td>68.9</td>
<td>71.2</td>
<td>72.6</td>
<td>72.7</td>
</tr>
<tr>
<td>bird</td>
<td>49.4</td>
<td>53.7</td>
<td>52.9</td>
<td>57.1</td>
</tr>
<tr>
<td>boat</td>
<td>60.3</td>
<td>60.5</td>
<td>61.0</td>
<td>62.7</td>
</tr>
<tr>
<td>bottle</td>
<td>75.3</td>
<td>77.9</td>
<td>77.9</td>
<td>82.8</td>
</tr>
<tr>
<td>bus</td>
<td>74.7</td>
<td>73.0</td>
<td>73.0</td>
<td>77.8</td>
</tr>
<tr>
<td>car</td>
<td>77.6</td>
<td>73.1</td>
<td>73.7</td>
<td>78.9</td>
</tr>
<tr>
<td>cat</td>
<td>21.4</td>
<td>25.2</td>
<td>26.4</td>
<td>28.0</td>
</tr>
<tr>
<td>chair</td>
<td>46.8</td>
<td>49.1</td>
<td>49.3</td>
<td>51.6</td>
</tr>
<tr>
<td>cow</td>
<td>71.8</td>
<td>68.7</td>
<td>68.4</td>
<td>73.1</td>
</tr>
<tr>
<td>mbike</td>
<td>63.9</td>
<td>64.1</td>
<td>64.1</td>
<td>72.8</td>
</tr>
<tr>
<td>person</td>
<td>76.5</td>
<td>73.9</td>
<td>74.0</td>
<td>81.5</td>
</tr>
<tr>
<td>plant</td>
<td>73.9</td>
<td>73.6</td>
<td>75.0</td>
<td>79.1</td>
</tr>
<tr>
<td>sheep</td>
<td>45.2</td>
<td>50.8</td>
<td>51.7</td>
<td>56.6</td>
</tr>
<tr>
<td>sofa</td>
<td>73.9</td>
<td>73.6</td>
<td>75.0</td>
<td>79.1</td>
</tr>
<tr>
<td>train</td>
<td>73.9</td>
<td>73.6</td>
<td>75.0</td>
<td>79.1</td>
</tr>
<tr>
<td>tv</td>
<td>62.2</td>
<td>62.1</td>
<td>62.9</td>
<td>67.6</td>
</tr>
</tbody>
</table>

Our front-end prediction module is simpler and more accurate than prior models. This table reports accuracy on the VOC-2012 test set. Segmentations produced by the different models on images from the VOC-2012 dataset are shown in Figure 2. The accuracy of the models on the VOC-2012 test set is reported in Table 2. Our front-end prediction module is both simpler and more accurate than the prior models. Specifically, our simplified model outperforms both FCN-8s and the DeepLab network by more than 5 percentage points on the test set. Interestingly, our simplified front-end module outperforms the leaderboard accuracy of DeepLab+CRF on the test set by more than a percentage point (67.6% vs. 66.4%).
Further Resources

http://blog.qure.ai/notes/semantic-segmentation-deep-learning-review
Overview

• Methods where output is now an image
  • Fully Convolutional Nets [Long et al., CVPR 2015]
  • Depth, normals and semantic labels from a single image [Eigen ICCV 2015]

• Image processing with Convnets
  • Image colorization [Zhang et al. ECCV 2016]
Beyond Object Classification with Convolutional Networks

David Eigen (NYU -> Clarifai)
Rob Fergus (Facebook / NYU)
Motivation

• Understand input scene
  – Semantic
  – Geometric
Motivation

- Understand input scene
  - Semantic
  - Geometric
• Understand input scene
  – Semantic
  – Geometric
Architecture

Input: 320x240

Output 1: 19x14
Input: 320x240

Output 2: 75x55
Architecture

Input: 320x240

Output: 147x109

conv+pool  concat  convolutions

upsample  upsample
Architecture

Input: 320x240

conv+pool  concat  convolutions

upsample
Architecture

Input: 320x240

conv+pool ← concat ← __________ convolutions ________

upsample

upsample
Losses

Depth: $d = D - D^*$  \[ D = \log \text{predicted depth}, \quad D^* = \log \text{true depth} \]

$L_{depth}(D, D^*) = \frac{1}{n} \sum d_i^2 - \frac{1}{2n^2} \left( \sum d_i \right)^2 + \frac{1}{n} \sum [\nabla_x d_i]^2 + [\nabla_y d_i]^2$
Training

• Pre-train Alexnet/VGGnet scale 1 with Imagenet
• Scale 2 & 3 random initialization
• Joint train layers 1 & 2 for each task
  – Loss on output of layer 2

• Fix layers 1 & 2, train layer 3

• For depth & normals task, share scale 1
  – But separate scale 2 & 3’s
  – 1.6x speedup
Evaluation

• NYU Depth dataset
  – RGB, Depth and per-pixel labels
  – Indoor scenes

• Supervised training of models

• Compare to range of other methods
  – Also on SIFTFlow and PASCAL VOC’11
Depth Comparison

- $m3d = \text{Make3D}\ [\text{Saxena & Ng} \ 2006]$
# Surface Normals

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
<td><img src="image16.png" alt="Image" /></td>
<td><img src="image17.png" alt="Image" /></td>
<td><img src="image18.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image19.png" alt="Image" /></td>
<td><img src="image20.png" alt="Image" /></td>
<td><img src="image21.png" alt="Image" /></td>
<td><img src="image22.png" alt="Image" /></td>
<td><img src="image23.png" alt="Image" /></td>
<td><img src="image24.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Results: Normals

Angle from Ground Truth

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours + VGG</td>
<td>0.2</td>
<td>0.97</td>
</tr>
<tr>
<td>Ours</td>
<td>0.75</td>
<td>15.2</td>
</tr>
<tr>
<td>Wang &amp; al</td>
<td>18.5</td>
<td>22.5</td>
</tr>
<tr>
<td>Fouhey &amp; al '14</td>
<td>27.5</td>
<td>30.5</td>
</tr>
<tr>
<td>Ladicky &amp; al</td>
<td>36.5</td>
<td>37.5</td>
</tr>
<tr>
<td>3DP</td>
<td>8.5</td>
<td>15.0</td>
</tr>
</tbody>
</table>
Output from each scale

Input

depth

coarse → fine

normals
Semantic Labels: NYUD
Results: NYUD 40 Classes

- Use RGB + ground truth depth & normals as inputs

![Bar chart showing per-pixel and per-class accuracy for different methods.

- Ours (VGG)
- Long & al
- Ours
- Gupta & al '14
- Gupta & al '13

Per-Pixel Acc.

<table>
<thead>
<tr>
<th>Method</th>
<th>54.25</th>
<th>56.00</th>
<th>57.75</th>
<th>59.50</th>
<th>61.25</th>
<th>63.00</th>
<th>64.75</th>
<th>66.50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours (VGG)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Long &amp; al</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gupta &amp; al '14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gupta &amp; al '13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Per-Class Acc.

<table>
<thead>
<tr>
<th>Method</th>
<th>20.00</th>
<th>27.00</th>
<th>34.00</th>
<th>41.00</th>
<th>48.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours (VGG)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Long &amp; al</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gupta &amp; al '14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gupta &amp; al '13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Semantic Labels: Pascal VOC’11

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Long &amp; al [19]</td>
<td>90.3</td>
<td>75.9</td>
<td>83.2</td>
<td>62.7</td>
</tr>
<tr>
<td>Ours (VGG)</td>
<td>90.3</td>
<td>72.4</td>
<td>82.9</td>
<td>62.2</td>
</tr>
</tbody>
</table>
Contribution from different scales

- On NYU Depth

![Contributions of Scales table]

- Depth & normals: scale 1 most important
- Semantic labels: scale 2 most important
  (if D & N are available)
Using Predicted Depths

• Use **predicted** depth/normals as input?

![Bar chart showing Per-Pixel Acc. and Per-Class Acc. for different datasets and combinations.]

- **Per-Pixel Acc.**
  - Scales 1+2: RGB only (53.1), RGB + Pred D&N (58.7), RGB + GT D&N (65.1)
  - Scale 2 only: RGB only (63.2), RGB + Pred D&N (65.8), RGB + GT D&N (69.8)

- **Per-Class Acc.**
  - Scales 1+2: RGB only (38.3), RGB + Pred D&N (43.8), RGB + GT D&N (50.6)
  - Scale 2 only: RGB only (43.3), RGB + Pred D&N (49.5), RGB + GT D&N (58.9)

• **NYU Depth 13-class**
  - RGB only
  - RGB + Pred D&N
  - RGB + GT D&N
Overview

• Methods where output is also an image
  – Fully Convolutional Nets [Long et al., CVPR 2015]
  – Depth, normals and semantic labels from a single image [Eigen ICCV 2015]

• Image processing with Convnets
Denoising with ConvNets

- Burger et al. “Can plain NNs compete with BM3D?” CVPR 2012
Learning to See in the Dark

[Chen et al., arXiv 1805.01934]

(a) Camera output with ISO 8,000
(b) Camera output with ISO 409,600
(c) Our result from the raw data of (a)

(a) Traditional pipeline
(b) ... followed by BM3D denoising
(c) Our result
Learning to See in the Dark

[Chen et al., arXiv 1805.01934]
Deblurring with ConvNets

- Blind deconvolution
  - Learning to Deblur, Schuler et al., arXiv 1406.7444, 2014
Inpainting with Convnets

• Image Denoising and Inpainting with Deep Neural Networks, Xie et al. NIPS 2012.

• Mask-specific inpainting with deep neural networks, Köhler et al., Pattern Recognition 2014

Original

Schmid CVPR’10

Köhler et al. ‘14
Removing Local Corruption

- Restoring An Image Taken Through a Window Covered with Dirt or Rain, Eigen et al., ICCV 2013.
Removing Local Corruption

Restoring An Image Taken Through a Window Covered with Dirt or Rain

Rain Sequence
Each frame processed independently

David Eigen, Dilip Krishnan and Rob Fergus
ICCV 2013
Enhanced Deep Residual Networks for Single Image Super-Resolution, Bee Lim Sanghyun Son Heewon Kim Seungjun Nah Kyoung Mu Le, CVPR 2017 workshop

Figure 2: Comparison of residual blocks in original ResNet, SRResNet, and ours.

Figure 3: The architecture of the proposed single-scale SR network (EDSR).
The 2018 PIRM Challenge on Perceptual Image Super-resolution

Yochai Blau*, Roey Mechrez*, Radu Timofte², Tomer Michaeli, and Lihi Zelnik-Manor¹

¹ Technion–Israel Institute of Technology, Haifa, Israel
² ETH Zurich, Switzerland
{yochai,roey}@campus.technion.ac.il

Figure 4. Submissions on the perception-distortion plane.
(a) Each submission is a point on the perception-distortion plane, whose axes are RMSE (²) and the PI (¹).

The perceptual quality of the challenge submissions exceeds that of the EDSR [19], EnhanceNet [31] and CX [24] baselines (plotted in red). Notice the tradeoff between perceptual quality and distortion, i.e. as the perceptual quality of the submissions improved (lower PI), their RMSE increased. (b) The mean-opinion score of 35 human raters vs. the mean perceptual index (PI) on the 10 top submission. The PI is highly-correlated with human opinion scores (Spearmans correlation of 0.83), as visualized by the least squares fit. This validates our choice of definition of the PI. A thorough analysis of other images quality measures appears in Section 5.

The mean human-opinion-scores are shown in Fig. 6. The human-opinion study validates that the challenge submissions surpassed the performance of state-of-the-art baselines by significant margins. Region 3 submissions, and even Region 2 submissions, are considered notably better than EnhanceNet by human raters. Region 1 submissions were rated far better in visual quality compared to EDSR (with only a slight increase in RMSE). The tradeoff between perceptual quality and distortion is once more revealed, as the best attainable perceptual quality increases with the increase in RMSE. Note that while the PI is well correlated with the human-opinion-scores on a coarse scale (in between regions), it is not always well-correlated with these scores on a finer scale (rankings within the regions), which can be seen when comparing the rankings in Table 1 and Fig. 6. This highlights the urgent need for better perceptual quality metrics, a point which is further analyzed in Section 5.

Figure 7 shows the normalized histogram of votes per method. Notice that all methods fail to achieve a large percentage of “definitely real” votes, indicating that there is still much to be done in perceptual super-resolution. In all submitted results, there tend to appear unnatural features in the reconstructions (at 4× magnification), which degrade the perceptual quality. Notice that the outputs of EDSR, a state-of-the-art algorithm in terms of distortion, are mostly voted as “definitely fake”. This is due to the aggressive averaging causing bluriness as a consequence of optimizing for distortion.
Fig. 5. Visual results.

SR results of several top methods in each region, along with the EDSR \[19\] and EnhanceNet\[31\] baselines. The attainable perceptual quality becomes higher as the allowed RMSE increases.
Class Project Admin

• Report
• 2 min summary video
• Deadline is Thursday Dec 17th midnight
  • Feel free to turn in earlier
  • Will *try* to grade them and compute final grades by Christmas
• Will post all of this to Piazza
Project Expectations

• Grading (45% of total grade for class)
  • Novelty / Technical difficulty of problem [15%]
  • Quality of Results [15%]
  • Quality of implementation [5%]
  • Quality of writeup [5%]
  • Presentation [5%]
  • How many people in your group
Project Expectations

• Report
  • 4-8 page conference paper style report on your project
  • Intro (with refs to related work)
  • Method (be sure to cite any code/pre-trained models)
  • Experiments (must have plots/results figures; also should have baselines; ideally some kind of ablation experiments too)
  • Discuss (brief)
  • See examples: http://openaccess.thecvf.com/CVPR2018.py
• Zip of source code or link to Github (please ensure you give access to robfergus)
• For videos:
  • 2mins PPT slides only.
Project Expectations

• Generalities
  • Please make sure you have *something* working, even if you don’t achieve overall goal
  • Even a small part of an ambitious project can be OK
  • So please have a safe plan B option in mind
  • Expect all projects to train something, i.e. must use b-prop at some point
  • Just evaluating existing models is NOT OK.
• Cluster gets busy -- please don’t leave it all to last moment.