Semantic segmentation

- Goal: classify each pixel into one of semantic classes
- Does not distinguish objects of the same class

http://jamie.shotton.org
Related problems

- Classification
- Classification + localization
- Semantic segmentation
- Object detection
- Instance segmentation

Single object
Just pixels (no objects)
Multiple objects

CAT
CAT
CAT
CAT
DOG
DOG
DOG
DOG

CS231N (Stanford)
Autonomous driving

- Differentiate road, pedestrians, traffic signs, ...

Cordts et al., *The Cityscapes Dataset for Semantic Urban Scene Understanding*
Image/video editing, visual effects

- Sky replacement

Tsai et al., *Sky is Not the Limit: Semantic-Aware Sky Replacement*
Map extraction from satellite image

- Distinguish streets from houses, parks etc.

Marmanis et al., *Semantic segmentation of aerial images with an ensemble of fully convolutional neural networks*
Outline

- Semantic segmentation and related problems
- Convolutional neural networks for image classification
- Fully convolutional networks
- Upsampling methods
- Refinement methods
- Evaluation
Image classification

- Goal: classify the main object in a given image
- Basis for solving other problems

Krizhevsky, Sutskever, Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*
ImageNet

- ImageNet 1K challenge
  - 1000 classes
  - 1.28 million training images
  - 50,000 test images

http://www.image-net.org/challenges/LSVRC/

Convolutional neural networks

- Feedforward neural networks suitable for image data
- Convolutional, max-pooling, and fully connected layers

Inception v3 (colors denote layer types)

Convolutional neural networks (cont.)

- Activations can be viewed as multichannel images
- Number of channels increases, spatial size decreases
Convolutional filter

- 3D tensor of weights which “slides” across input tensor
- Followed by a nonlinear activation function
- Generates one output channel ("feature map")
Example: edge detection

- Sobel operator is implemented as convolution

By Simpsons contributor, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=8904663
Convolutional layer

- Consists of multiple filters acting on the same input, each producing one output channel
Max-pooling layer

- Output is the max value within a sliding window
- Invariance to translation

![Max pooling diagram]
Fully connected layer

- Like a convolutional layer with filter size equal to its input size
  - Output spatial size 1 x 1

- Same as hidden layer in multilayer perceptrons
  - Product between matrix of weight and vector of inputs
Stride

- Distance between consecutive sliding window positions
- Horizontal and vertical stride usually equal
- Used for downsampling
Outline

- Semantic segmentation and related problems
- Intro to convolutional neural networks for image classification
  - Fully convolutional networks
- Upsampling methods
- Refinement methods
- Evaluation
Fully convolutional networks (FCNs)

- Family of architectures for pixelwise predictions

- Any architecture for image classification can be converted and finetuned for a pixelwise task

- Contain only “sliding window” layers (convolution, pooling)
  - Do not contain fully connected layers

Long, Shelhamer, Darrell, *Fully Convolutional Networks for Semantic Segmentation*
Reduction to image classification

- Apply image classifier to sliding window
- Apply result to the central part of each patch
Problem: inefficiency

- Patches classified independently even though they overlap
  - Not reusing computation
- Large number of heavily overlapping patches
  - Patch size much bigger than labeled region per patch
  - Need context to reliably classify

Patch size ~ 100s of pixels
Labeled region ~ a few pixels
Idea for reusing computation

- Convolution and pooling layers accept input of any size
  - Get proportionally bigger output
- Apply first layer to the whole image
  - Contains activations for all invocations of the classifier
- Apply second layer to the whole output of the first layer etc.
Converting fully connected layers

- Fully connected layer is just a convolutional layer applied only once
- Convert it to a convolutional layer acting on a bigger input
Recipe for vanilla fully convolutional network

- Start from a (pretrained) image classification network
  - AlexNet, VGG, Inception, ResNet, ...

- Convert any fully connected layers to convolutional layers

- Converted network can be applied to image of any size
  - Each layer’s output size changes proportionally to the input size
Downsampling effect

- Compared to the sliding window approach, we get a downsampled version of output
Downsampling effect: example

- Input image: 512 x 512
- Classifier network: input 256 x 256, output 1 x 1
- Comparison of output sizes

Sliding window approach
\[ 1 + (512 - 256) = 257 \]

Fully convolutional approach
\[ \approx 1 \cdot 512 / 256 = 2 \]
Outline

- Semantic segmentation and related problems
- Intro to convolutional neural networks for image classification
- Fully convolutional networks
- Upsampling methods
- Refinement methods
- Evaluation
Obtaining high-resolution output

- Two approaches
  - Eliminating existing downsampling
  - Appending explicit upsampling layers

- Usually a hybrid, dictated by computational efficiency
Eliminating strided convolution/pooling

- Replacing stride $s > 1$ by stride 1
- All subsequent feature maps are upsampled $s$ times
- Effectively, each subsequent kernel is $s$ times smaller
Dilated convolution

- Insert $s - 1$ zeros between every two consecutive entries of the original kernel.
Dilated convolution (cont.)

- Does not increase parameter count
- Can keep the same weights
  - Important when starting from a pretrained network
- Zero-optimization helps
- Can eliminate all downsampling in principle
  - Typically not done in practice due to computational complexity
Bilinear upsampling

- Each input multiplies the kernel
- Arrange kernels as sliding window
- Add overlapping elements

Bilinear kernel

Input

Output

3 x 3

5 x 5

7 x 7
Transposed convolution

- Generalization of bilinear upsampling
  - Weights can be learned
  - Channels can be “mixed”

Long, Shelhamer, Darrell, *Fully Convolutional Networks for Semantic Segmentation*
Max-unpooling

- Store indices of “winning” pixels in max-pooling
- Upsample by restoring to the same position

Noh et al., Learning Deconvolution Network for Semantic Segmentation
Outline

- Semantic segmentation and related problems
- Intro to convolutional neural networks for image classification
- Fully convolutional networks
- Upsampling methods
- Refinement methods
- Evaluation
Fusing high-res and low-res information

Shallow layers
Appearance information
High resolution
Poor classification

Deep layers
Semantic information
Low resolution
Good classification

Long, Shelhamer, Darrell, *Fully Convolutional Networks for Semantic Segmentation*
Fusion by simple addition

"Backbone" network

ConvNet → ConvNet → ConvNet → ConvNet

1/8x features → 1/16x features → 1/32x features

ConvNet → ConvNet → ConvNet → 2x

1/16x features + 1/32x features → 16x

ConvNet → ConvNet → ConvNet → 2x

1/32x features + 2x → 8x

High-res features + Low-res features → Upsample

Long, Shelhamer, Darrell, *Fully Convolutional Networks for Semantic Segmentation*
Effects of fusion

Long, Shelhamer, Darrell, *Fully Convolutional Networks for Semantic Segmentation*
Refinement modules

- More sophisticated way of fusing information from “shallower” layers

Lin et al., *RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation*
Refinement modules (cont.)

Lin et al., *RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation*
Conditional random field (CRF) postprocessing

- Encourage consistent labels for “similar” pixels
- Energy function
  - Nodes $x_i$: original network outputs
  - Nodes $y_j$: postprocessed outputs
  - Edges: energy terms (unary and pairwise)
- CRF inference
  - Given $x_i$ minimize energy over $y_i$
  - Practical algorithms exist only in special cases

Chen et al., *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*

Krähenbühl and Koltun, *Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials*
CRF postprocessing (cont.)

- Unary potential is high for labels which the network assigns low probability
- Pairwise potential is high if corresponding pixels in the original input image have similar color or spatial position
- Inference can be approximated by a convolutional network
  - Enables end-to-end training

Zheng et al., *Conditional Random Fields as Recurrent Neural Networks*
Outline

- Semantic segmentation and related problems
- Intro to convolutional neural networks for image classification
- Fully convolutional networks
- Upsampling methods
- Refinement methods
- Evaluation
Datasets

- Classic, relatively small
  - MSRC
  - SiftFlow
  - Stanford
  - LabelMe

- Newer & bigger
  - Pascal VOC
  - Microsoft COCO

- Including depth channel
  - NYU
  - Sun RGBD

- Autonomous driving
  - KITTI
  - CamVid
  - Cityscapes
PASCAL VOC challenge

- Image classification, object detection, semantic segmentation...
- 21 classes
- A few thousand images for training/test

Everingham et al., *The PASCAL Visual Object Classes (VOC) Challenge*
http://host.robots.ox.ac.uk/pascal/VOC/
Semantic segmentation metrics

- Intersection-over-union (IoU)
  - Intersection and union computed per class, and over whole set

- Mean intersection-over-union (mIOU)
  - Average over classes

- Standard accuracy, precision, and recall also used
Semantic segmentation metrics

- Accuracy
  \[
  \frac{TP}{TP + FN}
  \]

- Mean per-class accuracy
  \[
  \frac{1}{C} \sum_{i=1}^{C} \frac{TP_i}{TP_i + FN_i}
  \]

- Mean intersection-over-union (mIOU)
  \[
  \frac{1}{C} \sum_{i=1}^{C} \frac{TP_i}{TP_i + FN_i + FP_i}
  \]
## Pascal VOC leaderboard

[View the leaderboard on host.robots.ox.ac.uk:8080/leaderboard/](http://host.robots.ox.ac.uk:8080/leaderboard/)

<table>
<thead>
<tr>
<th>Method</th>
<th>mean</th>
<th>aircraft</th>
<th>bicycle</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>dining table</th>
<th>dog</th>
<th>horse</th>
<th>motorbike</th>
<th>person</th>
<th>potted plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>monitor</th>
<th>submission date</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDW-CNN [7]</td>
<td>96.3</td>
<td>94.8</td>
<td>67.3</td>
<td>53.4</td>
<td>74.8</td>
<td>84.6</td>
<td>55.3</td>
<td>86.6</td>
<td>93.6</td>
<td>54.1</td>
<td>94.9</td>
<td>79.0</td>
<td>93.3</td>
<td>55.5</td>
<td>91.7</td>
<td>89.2</td>
<td>77.5</td>
<td>93.7</td>
<td>79.2</td>
<td>94.0</td>
<td>80.8</td>
<td>30-Jun-2017</td>
</tr>
<tr>
<td>CASTA_IVA_SDN [7]</td>
<td>86.0</td>
<td><strong>96.6</strong></td>
<td><strong>76.4</strong></td>
<td>95.6</td>
<td>79.3</td>
<td>81.4</td>
<td>97.0</td>
<td>91.7</td>
<td>96.4</td>
<td>45.5</td>
<td>91.4</td>
<td>78.7</td>
<td>92.4</td>
<td>95.4</td>
<td>92.5</td>
<td>90.5</td>
<td>73.7</td>
<td>93.8</td>
<td>64.9</td>
<td>90.2</td>
<td><strong>83.0</strong></td>
<td>15-Jun-2017</td>
</tr>
<tr>
<td>DeepLabv3 [7]</td>
<td>85.7</td>
<td>96.4</td>
<td><strong>76.6</strong></td>
<td>92.7</td>
<td>77.8</td>
<td>87.6</td>
<td>96.7</td>
<td>90.2</td>
<td>95.4</td>
<td>47.5</td>
<td>93.4</td>
<td>76.3</td>
<td>91.4</td>
<td><strong>97.2</strong></td>
<td>91.0</td>
<td>92.1</td>
<td>71.3</td>
<td>90.9</td>
<td>68.9</td>
<td>90.6</td>
<td>79.3</td>
<td>20-Jun-2017</td>
</tr>
<tr>
<td>PSPNet [7]</td>
<td>85.4</td>
<td>95.8</td>
<td>72.7</td>
<td>95.0</td>
<td>78.9</td>
<td>84.4</td>
<td>94.7</td>
<td>92.0</td>
<td>95.7</td>
<td>43.1</td>
<td>91.0</td>
<td>80.3</td>
<td>91.3</td>
<td>96.3</td>
<td>92.3</td>
<td>90.1</td>
<td>71.5</td>
<td>94.4</td>
<td>66.9</td>
<td>88.8</td>
<td>82.0</td>
<td>08-Dec-2016</td>
</tr>
<tr>
<td><strong>ResNet-38_COCO</strong> [7]</td>
<td>84.9</td>
<td>96.2</td>
<td>75.2</td>
<td>95.4</td>
<td>74.4</td>
<td>81.7</td>
<td>93.7</td>
<td>89.9</td>
<td>92.5</td>
<td>48.2</td>
<td>92.0</td>
<td>79.0</td>
<td>90.1</td>
<td>95.5</td>
<td>91.8</td>
<td>91.2</td>
<td>73.0</td>
<td>90.5</td>
<td>65.4</td>
<td>88.7</td>
<td>80.6</td>
<td>22-Jan-2017</td>
</tr>
<tr>
<td>Multipath-RefineNet [7]</td>
<td>84.2</td>
<td>95.0</td>
<td>73.2</td>
<td>93.5</td>
<td>78.1</td>
<td>84.8</td>
<td>95.6</td>
<td>89.9</td>
<td>94.1</td>
<td>43.7</td>
<td>92.0</td>
<td>77.2</td>
<td>90.8</td>
<td>93.4</td>
<td>98.6</td>
<td>88.1</td>
<td>70.1</td>
<td>92.9</td>
<td>64.3</td>
<td>67.7</td>
<td>78.8</td>
<td>17-Jan-2017</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Method</th>
<th>mean</th>
<th>aircraft</th>
<th>bicycle</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>dining table</th>
<th>dog</th>
<th>horse</th>
<th>motorbike</th>
<th>person</th>
<th>potted plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>monitor</th>
<th>submission date</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIST_GDN_FCN [7]</td>
<td>62.2</td>
<td>74.5</td>
<td>31.9</td>
<td>66.7</td>
<td>49.7</td>
<td>60.5</td>
<td>76.9</td>
<td>75.9</td>
<td>76.0</td>
<td>22.9</td>
<td>57.6</td>
<td>54.5</td>
<td>73.0</td>
<td>59.4</td>
<td>75.0</td>
<td>73.7</td>
<td>51.0</td>
<td>67.5</td>
<td>43.3</td>
<td>70.0</td>
<td>56.4</td>
<td>27-Jul-2016</td>
</tr>
<tr>
<td>FCN-8s [7]</td>
<td>62.2</td>
<td>76.8</td>
<td>34.2</td>
<td>68.9</td>
<td>49.4</td>
<td>65.3</td>
<td>75.3</td>
<td>74.7</td>
<td>77.6</td>
<td>21.4</td>
<td>63.5</td>
<td>46.8</td>
<td>71.8</td>
<td>63.9</td>
<td>76.5</td>
<td>73.9</td>
<td>45.2</td>
<td>72.4</td>
<td>37.4</td>
<td>70.9</td>
<td>55.1</td>
<td>12-Nov-2014</td>
</tr>
<tr>
<td>MSRA_CFM [7]</td>
<td>61.8</td>
<td>75.7</td>
<td>25.7</td>
<td>69.5</td>
<td>48.8</td>
<td>65.6</td>
<td>81.0</td>
<td>59.2</td>
<td>73.3</td>
<td>30.0</td>
<td>68.7</td>
<td>51.5</td>
<td>69.1</td>
<td>68.1</td>
<td>71.7</td>
<td>67.5</td>
<td>50.4</td>
<td>66.5</td>
<td>44.4</td>
<td>58.9</td>
<td>52.5</td>
<td>17-Dec-2014</td>
</tr>
<tr>
<td><strong>SegNet</strong> [7]</td>
<td>59.9</td>
<td>73.6</td>
<td>37.6</td>
<td>62.0</td>
<td>46.8</td>
<td>58.6</td>
<td>79.1</td>
<td>70.1</td>
<td>65.4</td>
<td>23.6</td>
<td>60.4</td>
<td>45.6</td>
<td>61.8</td>
<td>63.5</td>
<td>75.3</td>
<td>74.9</td>
<td>42.6</td>
<td>63.7</td>
<td>42.5</td>
<td>67.8</td>
<td>52.7</td>
<td>10-Nov-2015</td>
</tr>
<tr>
<td>TTI_zoomout [7]</td>
<td>58.4</td>
<td>70.3</td>
<td>31.9</td>
<td>66.3</td>
<td>46.4</td>
<td>52.1</td>
<td>75.3</td>
<td>68.4</td>
<td>75.3</td>
<td>19.2</td>
<td>58.4</td>
<td>49.9</td>
<td>65.6</td>
<td>63.0</td>
<td>70.1</td>
<td>67.6</td>
<td>41.5</td>
<td>64.0</td>
<td>34.9</td>
<td>64.2</td>
<td>47.3</td>
<td>17-Nov-2014</td>
</tr>
<tr>
<td>SDS [7]</td>
<td>51.6</td>
<td>63.3</td>
<td>25.7</td>
<td>63.0</td>
<td>39.8</td>
<td>59.2</td>
<td>70.9</td>
<td>61.4</td>
<td>54.9</td>
<td>16.8</td>
<td>45.0</td>
<td>48.2</td>
<td>50.5</td>
<td>51.0</td>
<td>57.7</td>
<td>63.3</td>
<td>31.8</td>
<td>58.7</td>
<td>31.2</td>
<td>55.7</td>
<td>48.5</td>
<td>21-Jul-2014</td>
</tr>
<tr>
<td>NUS_UDS [7]</td>
<td>50.0</td>
<td>67.0</td>
<td>24.5</td>
<td>47.2</td>
<td>45.0</td>
<td>47.9</td>
<td>65.3</td>
<td>60.6</td>
<td>58.5</td>
<td>15.5</td>
<td>50.8</td>
<td>37.4</td>
<td>45.8</td>
<td>59.0</td>
<td>62.0</td>
<td>52.7</td>
<td>40.8</td>
<td>48.2</td>
<td>35.8</td>
<td>53.1</td>
<td>45.6</td>
<td>29-Oct-2014</td>
</tr>
<tr>
<td>TTIC-divmbest-rerank [7]</td>
<td>48.1</td>
<td>62.7</td>
<td>25.6</td>
<td>46.9</td>
<td>43.0</td>
<td>54.8</td>
<td>58.4</td>
<td>58.6</td>
<td>55.6</td>
<td>14.6</td>
<td>47.5</td>
<td>31.2</td>
<td>44.7</td>
<td>51.0</td>
<td>60.9</td>
<td>53.5</td>
<td>36.6</td>
<td>50.9</td>
<td>30.1</td>
<td>50.2</td>
<td>46.8</td>
<td>15-Nov-2012</td>
</tr>
<tr>
<td>BONN_02PCPMC_FGT_SEG [7]</td>
<td>47.8</td>
<td>64.0</td>
<td>27.3</td>
<td>54.1</td>
<td>39.2</td>
<td>48.7</td>
<td>56.6</td>
<td>57.7</td>
<td>52.5</td>
<td>14.2</td>
<td>54.8</td>
<td>29.6</td>
<td>42.2</td>
<td>58.0</td>
<td>54.8</td>
<td>50.2</td>
<td>36.6</td>
<td>58.6</td>
<td>31.6</td>
<td>48.4</td>
<td>38.6</td>
<td>08-Aug-2013</td>
</tr>
</tbody>
</table>
Summary

- Fully convolutional neural networks can be used for pixelwise classification on images
- Can start from pretrained network for image classification
- Need to adapt architecture to upsample and refine output