CNNs for dense image labeling

- Image classification
- Object detection
- Semantic segmentation
- Instance segmentation
Outline

• Operations and architectures for dense prediction: U-Net
• Instance segmentation: Mask R-CNN
• Other dense prediction problems
Dense prediction architectures

- To make predictions for all pixels at once, we can design a network with only stride-1 convolutions and element-wise operations
- What are the pros and cons of this approach?

Source: Stanford CS231n
Dense prediction architectures

- Practical solution: first downsample, then upsample

Source: Stanford CS231n
Dense prediction architectures

Problem: no way to recover information lost to downsampling

Fuse encoder and decoder feature maps at the same resolution

Figure source
Recall: Feature pyramid networks

- Improve predictive power of lower-level feature maps by adding contextual information from higher-level feature maps.
- Predict different sizes of bounding boxes from different levels of the pyramid (but share parameters of predictors).

U-Net

Feature map upsampling

Step 1: increase the resolution of the feature grid
Feature map upsampling

Step 2: **interpolate** to get the missing values
Bilinear interpolation

\[ \frac{1}{2} (A + B) \]
Bilinear interpolation

\[ \frac{1}{2} (A + C) \]
Bilinear interpolation

\[ \frac{1}{4}(A + B + C + D) \]
Feature map upsampling

- For 2x upsampling, dilate the input by inserting rows and columns of zeros between adjacent entries, convolve with upsampling filter

V. Dumoulin and F. Visin, *A guide to convolution arithmetic for deep learning*, arXiv 2018
Feature map upsampling: Max unpooling

- Can be used when max pooling is used to downsample

Remember pooling indices (which element was max)

Output is sparse, so unpooling is typically followed by a transposed convolution layer
DeconvNet

H. Noh, S. Hong, and B. Han, Learning Deconvolution Network for Semantic Segmentation, ICCV 2015
DeconvNet

H. Noh, S. Hong, and B. Han, Learning Deconvolution Network for Semantic Segmentation, ICCV 2015
SegNet

Drop the FC layers, get better results

Dense prediction: Outline

- Operations and architectures for dense prediction: U-Net
- Instance segmentation: Mask R-CNN
Instance segmentation

Object Detection  Semantic Segmentation  Instance Segmentation

Source: Kaiming He
Mask R-CNN

- Mask R-CNN = Faster R-CNN + FCN on RoIs

K. He, G. Gkioxari, P. Dollar, and R. Girshick, Mask R-CNN, ICCV 2017 (Best Paper Award)
RoIAlign vs. RoIPool

- RoIPool: nearest neighbor quantization

Source: K. He, R. Girshick
RoIAlign vs. RoIPool

- RoIPool: nearest neighbor quantization
- RoIAlign: bilinear interpolation

Source: K. He, R. Girshick
Bilinear interpolation

\[ f(x, y) = w_{11}A + w_{21}B + w_{12}C + w_{22}D \]

\[
w_{11} = \frac{(x_2 - x)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} \quad \quad w_{21} = \frac{(x - x_1)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)}
\]
\[
w_{12} = \frac{(x_2 - x)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} \quad \quad w_{22} = \frac{(x - x_1)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)}
\]

http://en.wikipedia.org/wiki/Bilinear_interpolation
Mask R-CNN

Classification+regression branch

Mask branch: separately predict segmentation for each possible class
Mask R-CNN

- From RoIAlign features, predict class label, bounding box, and segmentation mask

Classification/regression head from an established object detector (e.g., FPN)

Separately predict binary mask for each class with per-pixel sigmoids, use average binary cross-entropy loss
Mask R-CNN

Validation image with box detection shown in red
Example results
Example results
## Instance segmentation results on COCO

<table>
<thead>
<tr>
<th>Model</th>
<th>backbone</th>
<th>AP</th>
<th>AP$_{50}$</th>
<th>AP$_{75}$</th>
<th>AP$_S$</th>
<th>AP$_M$</th>
<th>AP$_L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNC [10]</td>
<td>ResNet-101-C4</td>
<td>24.6</td>
<td>44.3</td>
<td>24.8</td>
<td>4.7</td>
<td>25.9</td>
<td>43.6</td>
</tr>
<tr>
<td>FCIS [26] +OHEM</td>
<td>ResNet-101-C5-dilated</td>
<td>29.2</td>
<td>49.5</td>
<td>-</td>
<td>7.1</td>
<td>31.3</td>
<td>50.0</td>
</tr>
<tr>
<td>FCIS+++ [26] +OHEM</td>
<td>ResNet-101-C5-dilated</td>
<td>33.6</td>
<td>54.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Mask R-CNN</td>
<td>ResNet-101-C4</td>
<td>33.1</td>
<td>54.9</td>
<td>34.8</td>
<td>12.1</td>
<td>35.6</td>
<td>51.1</td>
</tr>
<tr>
<td>Mask R-CNN</td>
<td>ResNet-101-FPN</td>
<td>35.7</td>
<td>58.0</td>
<td>37.8</td>
<td>15.5</td>
<td>38.1</td>
<td>52.4</td>
</tr>
<tr>
<td>Mask R-CNN</td>
<td>ResNeXt-101-FPN</td>
<td><strong>37.1</strong></td>
<td><strong>60.0</strong></td>
<td><strong>39.4</strong></td>
<td><strong>16.9</strong></td>
<td><strong>39.9</strong></td>
<td><strong>53.5</strong></td>
</tr>
</tbody>
</table>

AP at different IoU thresholds

AP for different size instances
Keypoint prediction

- Given $K$ keypoints, train model to predict $K \times m \times m$ one-hot maps with cross-entropy losses over $m^2$ outputs
Outline

• Operations and architectures for dense prediction: U-Net
• Instance segmentation: Mask R-CNN
• Other dense prediction problems
Panoptic segmentation

A. Kirillov et al. Panoptic segmentation. CVPR 2019
Panoptic feature pyramid networks

A. Kirillov et al. Panoptic feature pyramid networks. CVPR 2019
Panoptic feature pyramid networks

Figure 2: Panoptic FPN results on COCO (top) and Cityscapes (bottom) using a single ResNet-101-FPN network.

A. Kirillov et al. Panoptic feature pyramid networks. CVPR 2019
Amodal instance segmentation

K. Li and J. Malik. Amodal instance segmentation. ECCV 2016
Promptable segmentation

A. Kirillov et al. Segment anything, ICCV 2023
Depth and normal estimation

D. Eigen and R. Fergus, Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture, ICCV 2015
Depth and normal estimation

D. Eigen and R. Fergus, Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture, ICCV 2015
Colorization

R. Zhang, P. Isola, and A. Efros, Colorful Image Colorization, ECCV 2016
Image-to-image translation (paired)

Image-to-image translation (unpaired)

Image generation

Encoder-decoder

random seed or latent vector

\( z \rightarrow y \)