اصول پردازش تصویر

Principles of Image Processing

مصطفي کمالی تبریزی

۱ آذر ۱۳۹۹

جلسه هجدهم

Homework 3

Q1: K-means

```
1.1327 0.19025
0.9439 -0.0044111
0.94656 0.14228
0.92519 0.26221
1.1118 0.4136
0.81575 0.46691
0.96601 0.30631
0.80655 0.45992
0.71181 0.55357
0.83549 0.58476
0.78186 0.64711
0.68142 0.62212
0.74819 0.67477
0.68251 0.63
0.71549 0.72334
0.62308 0.83761
0.45217 1.0251
0.49104 1.0869
0.48987 0.85132
0.40835 1.0926
0.31331 1.0193
0.089689 0.78134
0.15941 1.0996
-0.012438 0.96842
0.022464 1.0159
0.066415 0.8569
-0.047798 0.92105
-0.15017 0.97016
-0.059431 1.0506
-0.10109 1.0371
-0.25395 0.9639
-0.47779 0.88034
-0.42544 0.88133
-0.51394 0.87393
-0.46246 0.9582
-0.76606 0.72096
-0.5518 0.78199
-0.61864 0.85075
-0.56877 0.68247
-0.87477 0.74933
-0.7981 0.75556
-0.77556 0.39027
-0.76083 0.45089
-1.0226 0.44431
-1.0716 0.48753
```

Q2: Mean-shift

Q3: SLIC

Q4: Segmentation

Q5:

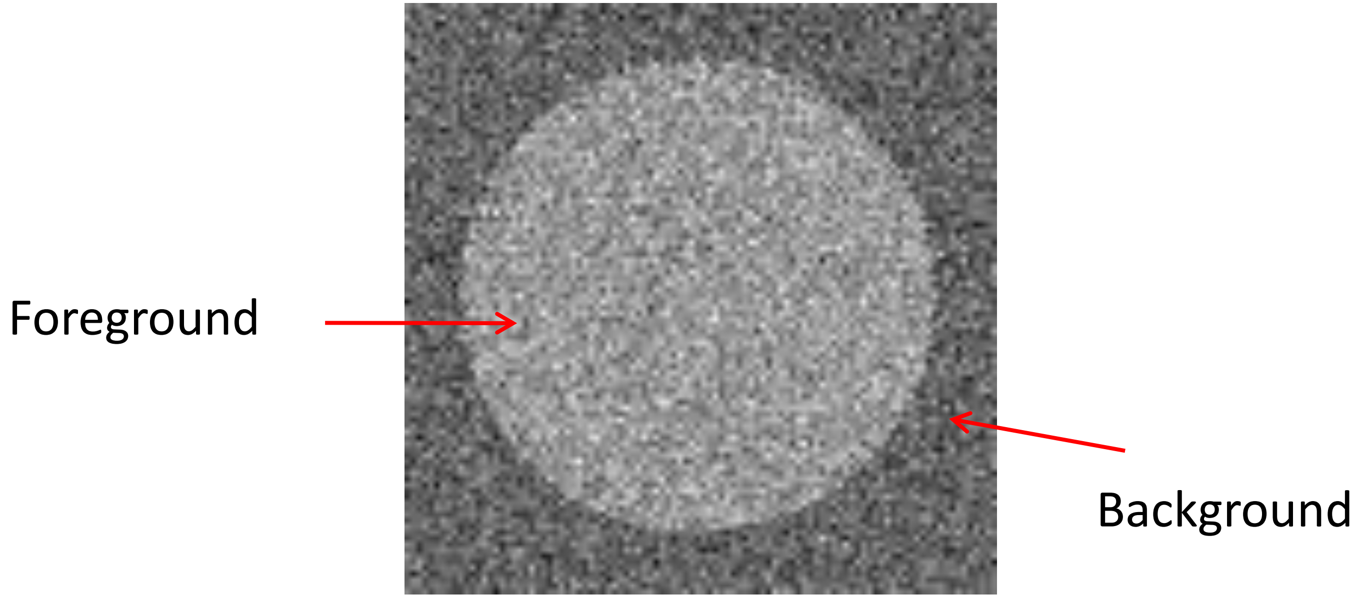
Hidden Variables, the EM Algorithm, and Mixtures of Gaussians

Slides adopted from Derek Hoiem

Missing Data Problems: Segmentation

You are given an image and want to assign foreground/background pixels.

Challenge: Segment the image into figure and ground without knowing what the foreground looks like in advance.

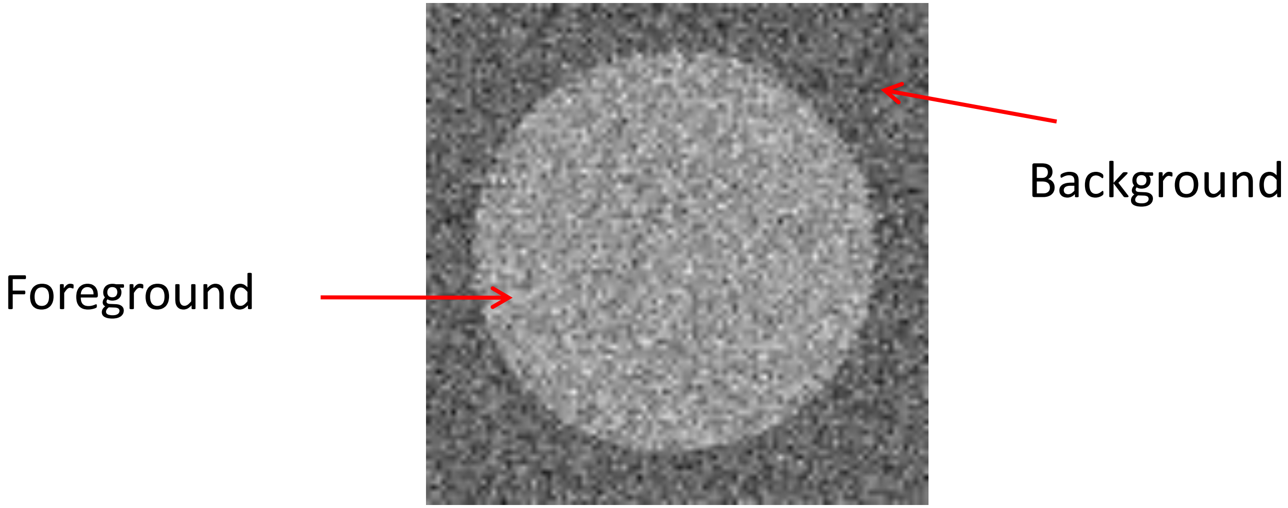


Missing Data Problems: Segmentation

Challenge: Segment the image into figure and ground without knowing what the foreground looks like in advance.

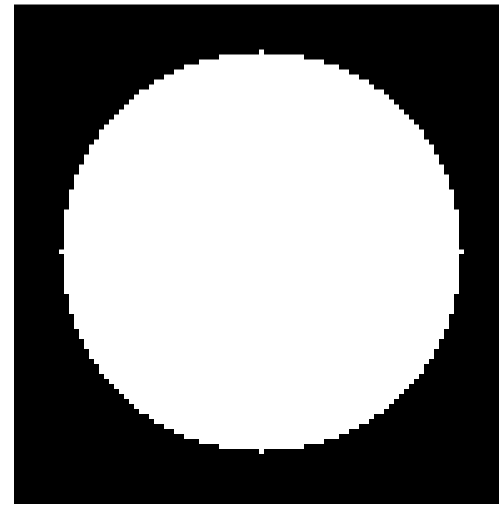
Three steps:

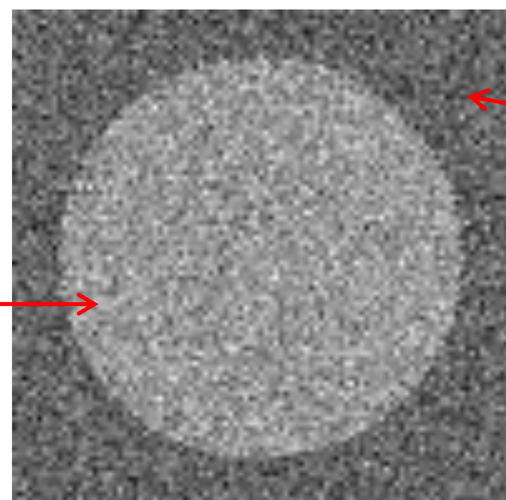
1. If we had labels, how could we model the appearance of foreground and background?
2. Once we have modeled the fg/bg appearance, how do we compute the likelihood that a pixel is foreground?
3. How can we get both labels and appearance models at once?



Maximum Likelihood Estimation (MLE)

1. If we had labels, how could we model the appearance of foreground and background?





Background

Foreground

Maximum Likelihood Estimation (MLE)

data

$$\mathbf{x} = \{x_1, \dots, x_N\}$$

parameters

$$\hat{\theta} = \operatorname*{argmax}_{\theta} p(\mathbf{x}|\theta)$$

$$\hat{\theta} = \operatorname*{argmax}_{\theta} \prod_{n} p(x_n|\theta)$$

Maximum Likelihood Estimation (MLE)

data

$$\mathbf{x} = \{x_1, \dots, x_N\}$$

parameters

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, p(\mathbf{x}|\theta)$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \prod_n p(x_n|\theta)$$

Gaussian Distribution

$$p(x_n|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_n - \mu)^2}{2\sigma^2}\right)$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \prod_n p(x_n|\theta)$$

$$\boxed{p(x_n|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_n-\mu)^2}{2\sigma^2}\right)}$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \log\left(\prod_n p(x_n|\theta)\right)$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_n \log\big(p(x_n|\theta)\big)$$

$$S = \sum_n \log\big(p(x_n|\theta)\big) = \sum_n \left(\log\left(\frac{1}{\sqrt{2\pi}}\right) + \log\left(\frac{1}{\sigma}\right) - \frac{(x_n-\mu)^2}{2\sigma^2}\right)$$

$$\frac{\partial S}{\partial \mu} = 0 \Longrightarrow \sum_n \frac{x_n-\mu}{\sigma^2} = 0 \Longrightarrow \mu = \frac{1}{N}\sum_n x_n$$

$$\frac{\partial S}{\partial \sigma} = 0 \Longrightarrow \sum_n \left(-\frac{1}{\sigma} + \frac{(x_n-\mu)^2}{\sigma^3}\right) = 0 \Longrightarrow N\sigma^2 = \sum_n (x_n-\mu)^2$$

$$\Longrightarrow \sigma^2 = \frac{1}{N}\sum_n (x_n-\mu)^2$$

Maximum Likelihood Estimation (MLE)

data

$$\mathbf{x} = \{x_1, \dots, x_N\}$$

parameters

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, p(\mathbf{x}|\theta)$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \prod_n p(x_n|\theta)$$

Gaussian Distribution

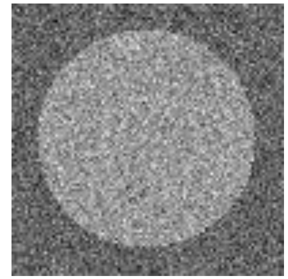
$$p(x_n|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_n - \mu)^2}{2\sigma^2}\right)$$

$$\mu = \frac{1}{N}\sum_n x_n \qquad \sigma^2 = \frac{1}{N}\sum_n (x_n - \mu)^2$$

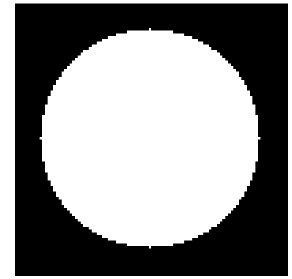
Example: MLE

Parameters used to Generate

```
fg: mu=0.6, sigma=0.1
bg: mu=0.4, sigma=0.1
```



im

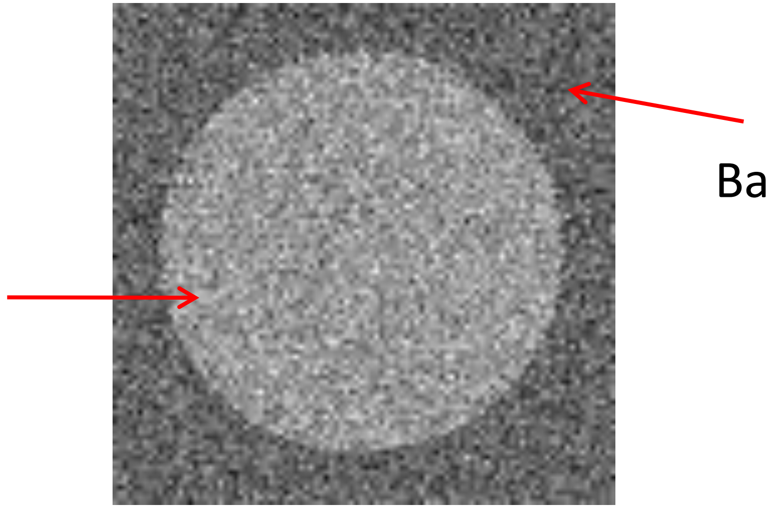


labels

```
>> mu_fg = mean(im(labels))
    mu_fg = 0.6012

>> sigma_fg = sqrt(mean((im(labels)-mu_fg).^2))
    sigma_fg = 0.1007

>> mu_bg = mean(im(~labels))
    mu_bg = 0.4007

>> sigma_bg = sqrt(mean((im(~labels)-mu_bg).^2))
    sigma_bg = 0.1007

>> pfg = mean(labels(:));
```

Probabilistic Inference

2. Once we have modeled the fg/bg appearance, how do we compute the likelihood that a pixel is foreground?



Background

Foreground

Probabilistic Inference

Compute the likelihood that a particular model generated a sample

component or label

$$p(z_n = m | x_n, \theta)$$

Probabilistic Inference

Compute the likelihood that a particular model generated a sample

component or label

$$p(z_n = m | x_n, \theta) = \frac{p(z_n = m, x_n | \theta_m)}{p(x_n | \theta)}$$

$$\boxed{P(A|B) = \frac{P(A, B)}{P(B)}}$$

Probabilistic Inference

Compute the likelihood that a particular model generated a sample

component or label

$$p(z_n = m | x_n, \theta) = \frac{p(z_n = m, x_n | \theta_m)}{p(x_n | \theta)}$$

$$= \frac{p(z_n = m, x_n | \theta_m)}{\sum_k p(z_n = k, x_n | \theta_k)}$$

Probabilistic Inference

Compute the likelihood that a particular model generated a sample

component or label

$$p(z_n = m|x_n, \theta) = \frac{p(z_n{=}m, x_n|\theta_m)}{p(x_n|\theta)}$$

$$= \frac{p(z_n{=}m, x_n|\theta_m)}{\sum_k p(z_n{=}k, x_n|\theta_k)}$$

$$= \frac{p(x_n|z_n{=}m, \theta_m)p(z_n{=}m|\theta_m)}{\sum_k p(x_n|z_n{=}k, \theta_k)p(z_n{=}k|\theta_k)}$$

$$\boxed{P(A|B) = \frac{P(A,B)}{P(B)}}$$

Probabilistic Inference

Compute the likelihood that a particular model generated a sample

component or label

$$p(z_n = m|x_n, \theta) = \frac{p(z_n{=}m, x_n|\theta_m)}{p(x_n|\theta)}$$

$$= \frac{p(z_n{=}m, x_n|\theta_m)}{\sum_k p(z_n{=}k, x_n|\theta_k)}$$

Prior probability

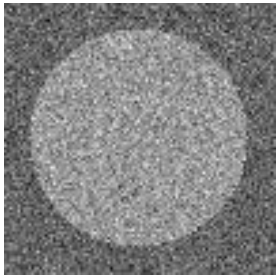
$$= \frac{p(x_n|z_n{=}m, \theta_m)\boxed{p(z_n{=}m|\theta_m)}}{\sum_k p(x_n|z_n{=}k, \theta_k)\boxed{p(z_n{=}k|\theta_k)}}$$

$$\boxed{P(A|B) = \frac{P(A,B)}{P(B)}}$$

Example: Inference

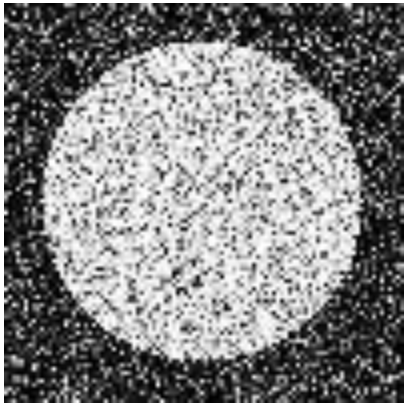
Learned Parameters

```
fg: mu=0.6, sigma=0.1
bg: mu=0.4, sigma=0.1
Pfg = 0.5
```



im

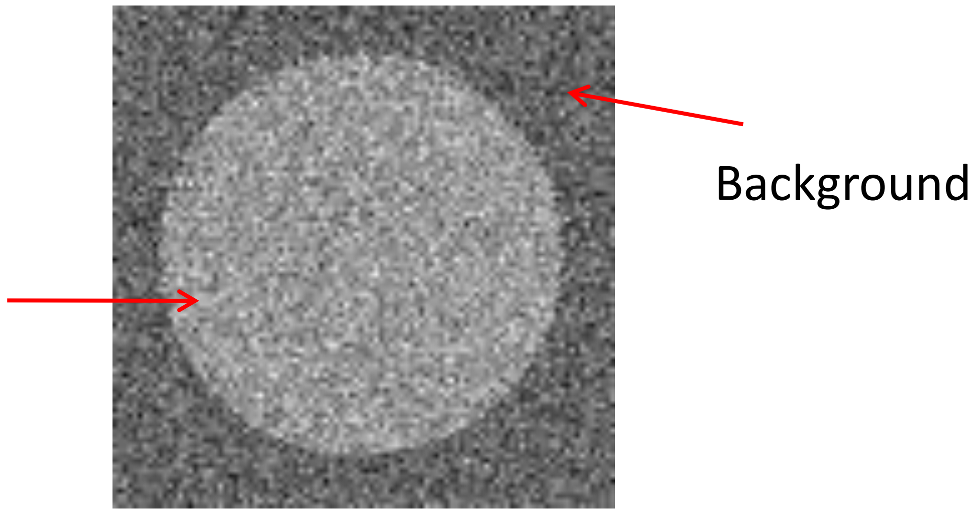
```
>> px_fg = normpdf(im, mu_fg, sigma_fg);
>> px_bg = normpdf(im, mu_bg, sigma_bg);
>> pfg_x = px_fg*pfg ./ (px_fg*pfg + px_bg*(1-pfg));
```



$p(fg \mid im)$

Dealing with Hidden Variables

3. How can we get both labels and appearance parameters at once?



Mixture of Gaussians

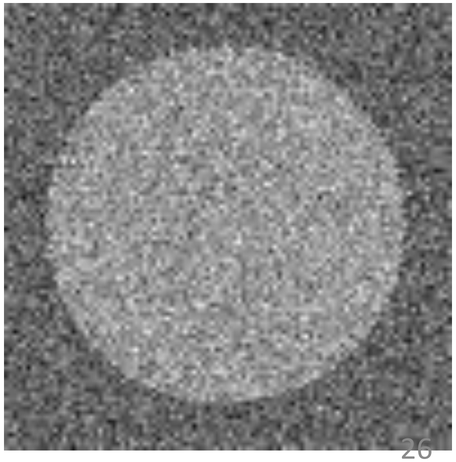
With enough components, can represent any probability density function

- Widely used as general purpose pdf estimator

Segmentation with Mixture of Gaussians

Pixels come from one of several Gaussian components

- We don’t know which pixels come from which components
- We don’t know the parameters for the components



Simple solution

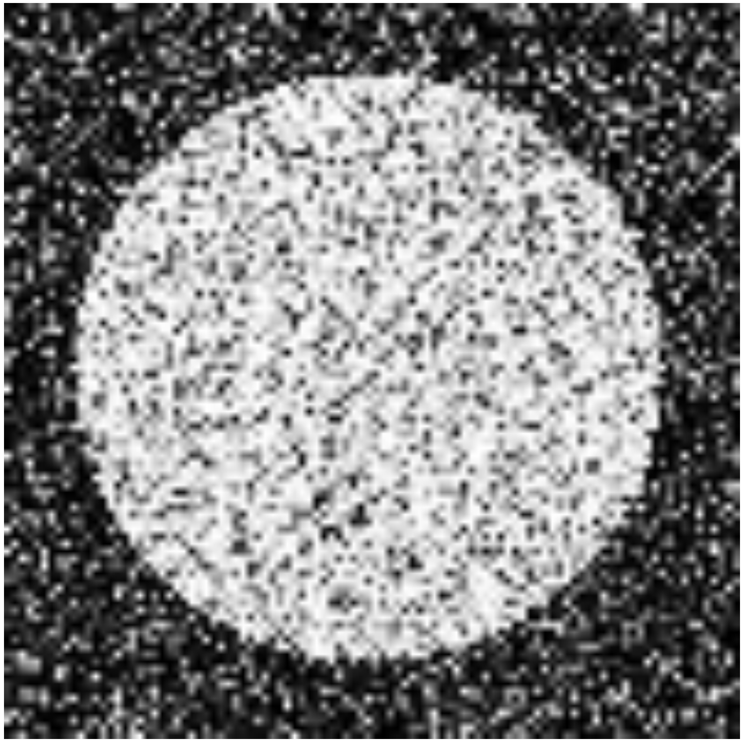
1. Initialize parameters

2. Compute the probability of each hidden variable given the current parameters

3. Compute new parameters for each model, weighted by likelihood of hidden variables

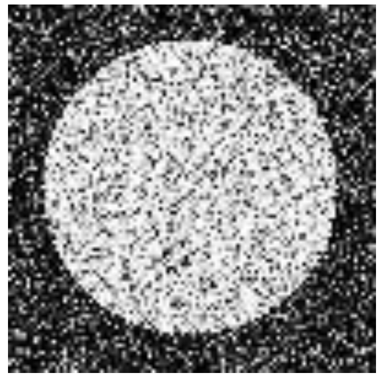
4. Repeat 2-3 until convergence

What’s wrong with this prediction?



$p(foreground \mid image)$

Solution: encode dependencies between pixels



$p(foreground \mid image)$

Normalizing constant called “partition function”

$$p(\mathbf{y}|\theta, image) = \frac{1}{Z} \prod_{i=1}^{n} p_1(z_i|\theta, image) \prod_{i,j \epsilon edges} p_2(z_i, z_j|\theta, image)$$

Labels to be predicted

Individual predictions

Pairwise predictions

Writing Likelihood as an "Energy"

minimizing

$$p(\mathbf{y}|\theta, image) = \frac{1}{Z} \prod_{i=1}^{n} p_1(z_i|\theta, image) \prod_{i,j \epsilon edges} p_2\left(z_i, z_j|\theta, image\right)$$

$$\Big\downarrow -\log(\quad)$$

maximizing

$$Energy(z|\theta, image) = \sum_{i=1}^{n} \psi_1(z_i|\theta, image) + \sum_{i,j \in edges} \psi_2\left(z_i, z_j|\theta, image\right)$$

cost of assignment z_i
(unary term)

cost of pairwise assignment z_i, z_j
(pairwise term)

Notes on energy-based formulation

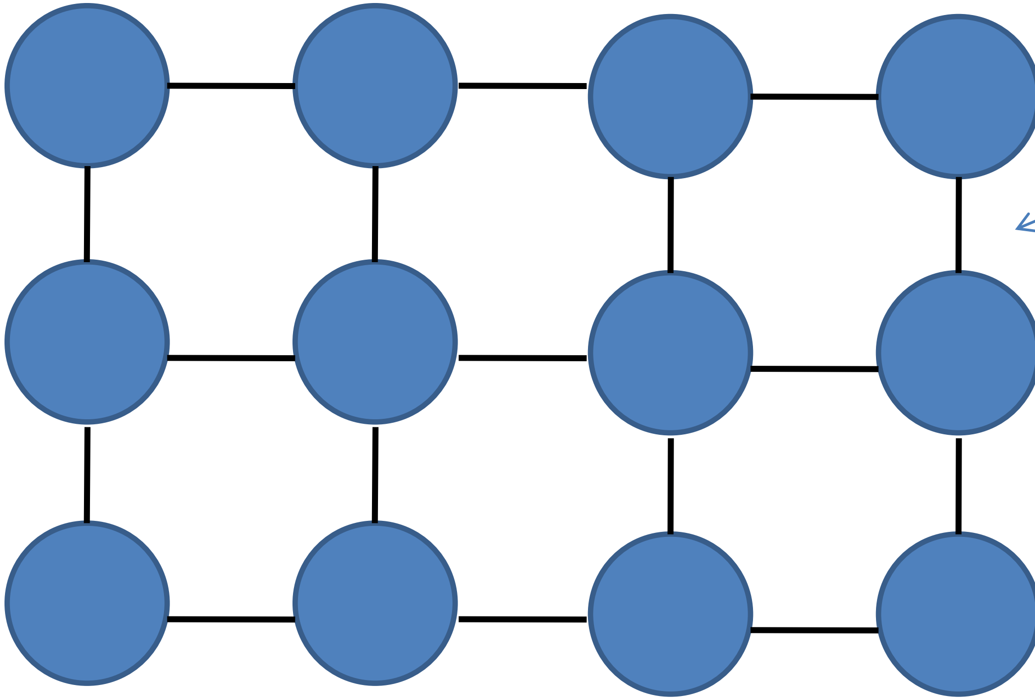
$$Energy(z|\theta, image) = \sum_{i=1}^{n} \psi_1(z_i|\theta, image) + \sum_{i,j \in edges} \psi_2\left(z_i, z_j|\theta, image\right)$$

- Primarily used when you only care about the most likely solution (not the confidences)

- Can think of it as a general cost function

- Can have larger “cliques” than 2
  - Clique is the set of variables that go into a potential function

Markov Random Fields



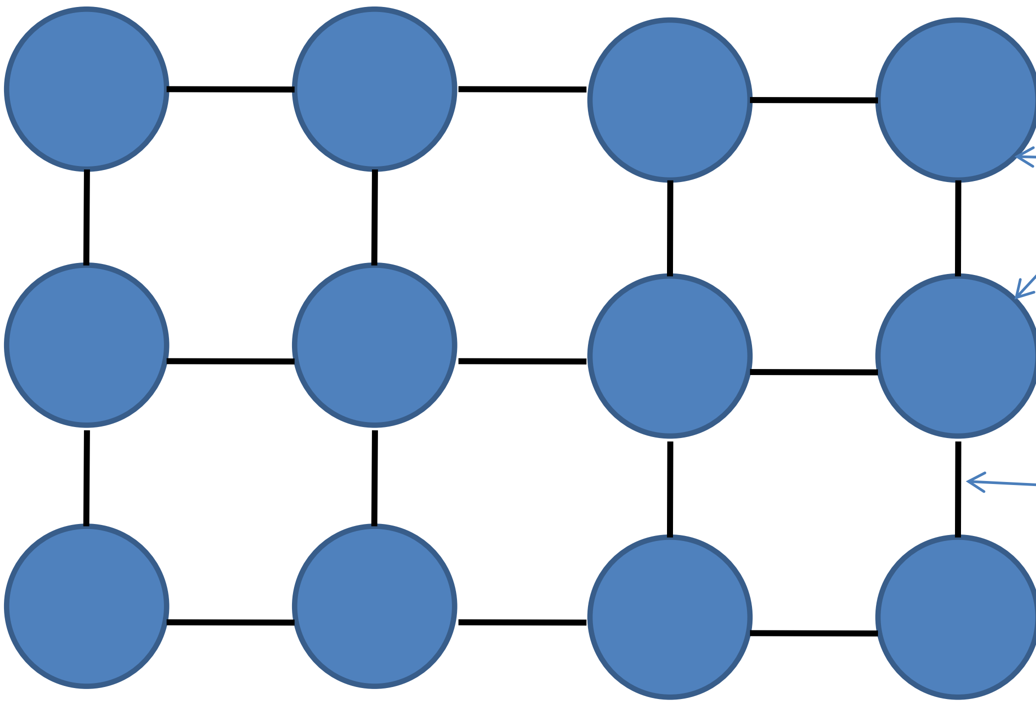
Cost to assign a label to each pixel

Cost to assign a pair of labels to connected pixels

$$Energy(z|\theta, image) = \sum_{i=1}^{n} \psi_1(z_i|\theta, image) + \sum_{i,j \in edges} \psi_2(z_i, z_j|\theta, image)$$

Markov Random Fields

- Example: “label smoothing” grid



Unary potential

$0: -\log\big(p(z_i = 0 \mid data)\big)$

$1: -\log\big(p(z_i = 1 \mid data)\big)$

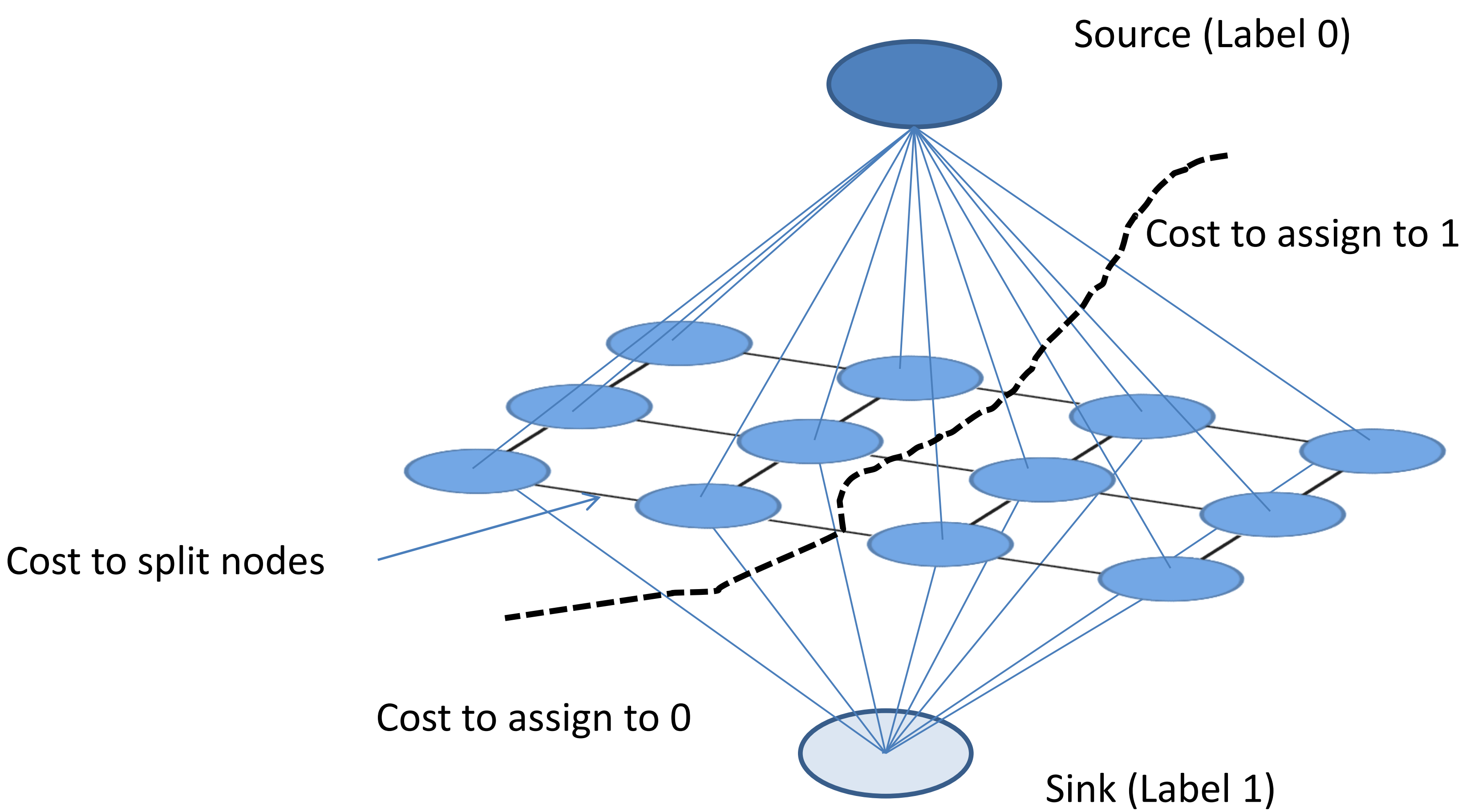
Pairwise Potential

	0	1
0	0	K
1	K	0

$K > 0$

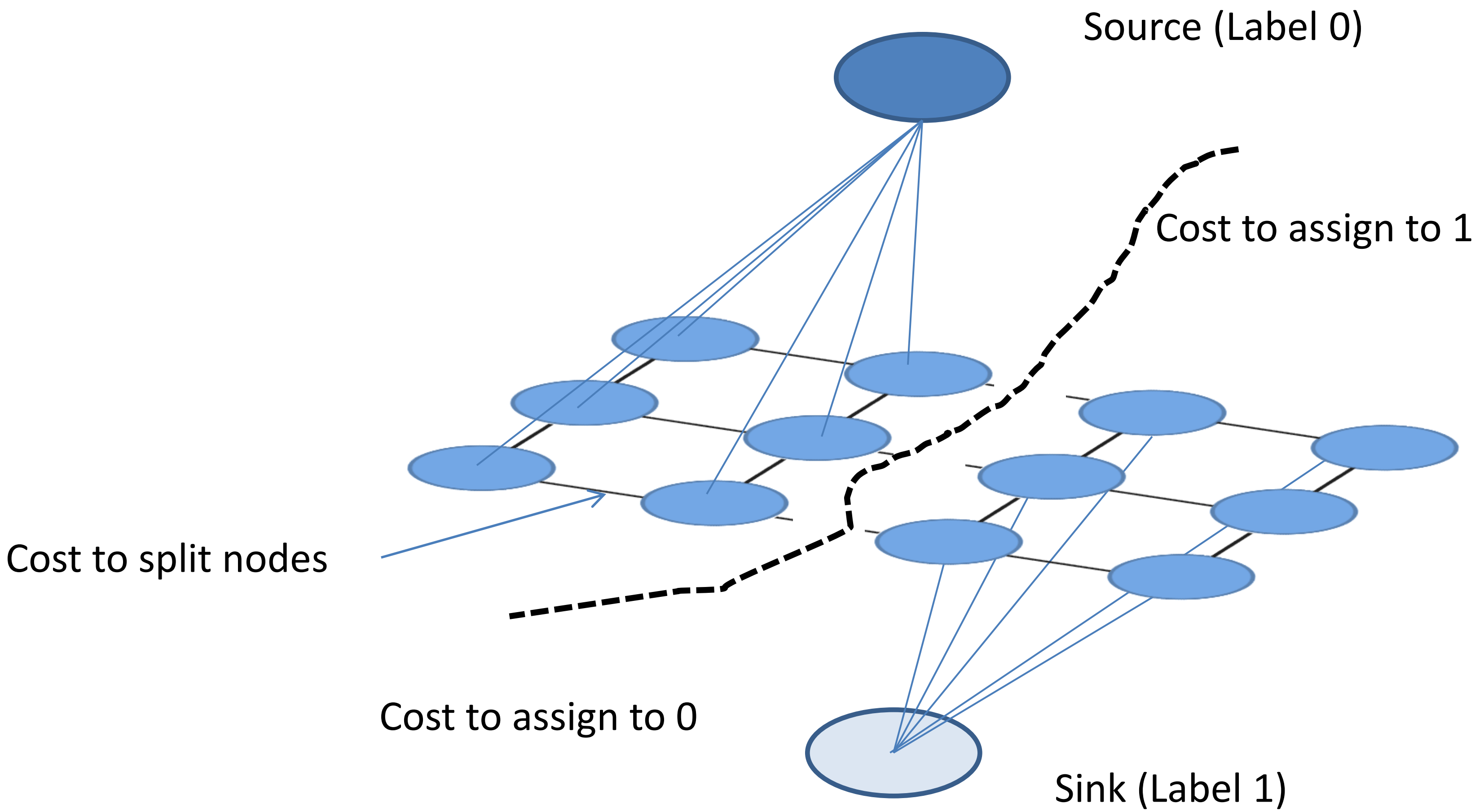
$$Energy(z|\theta, image) = \sum_{i=1}^{n} \psi_1(z_i|\theta, image) + \sum_{i,j \in edges} \psi_2\big(z_i, z_j|\theta, image\big)$$

Solving MRFs with graph cuts



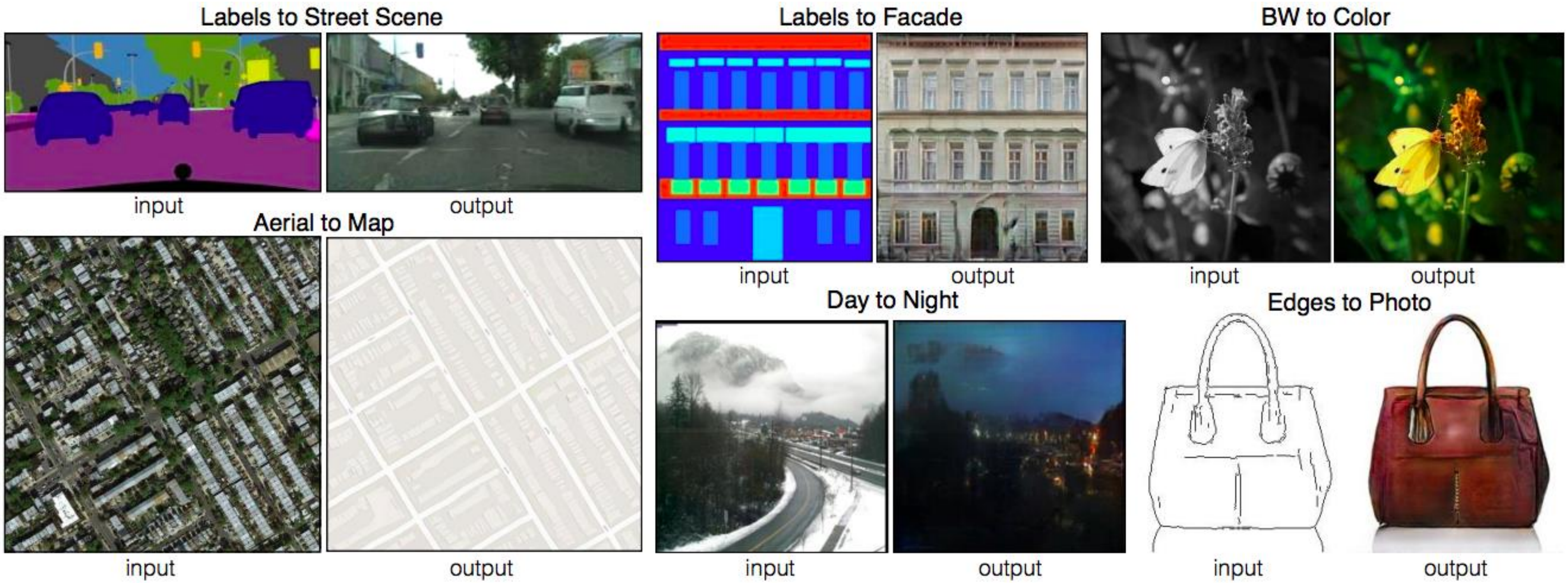
$$Energy(z|\theta, image) = \sum_{i=1}^{n} \psi_1(z_i|\theta, image) + \sum_{i,j \in edges} \psi_2(z_i, z_j|\theta, image)$$

Solving MRFs with graph cuts



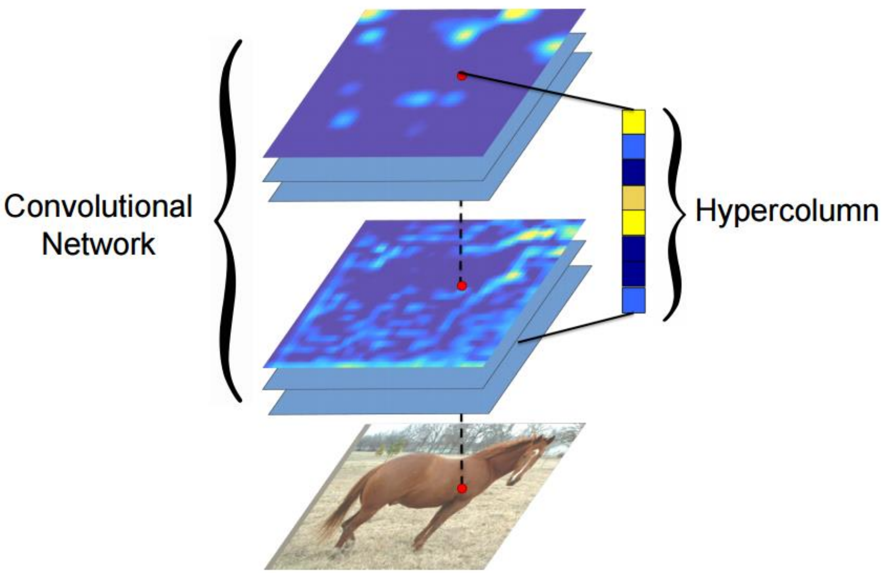
$$Energy(z|\theta, image) = \sum_{i=1}^{n} \psi_1(z_i|\theta, image) + \sum_{i,j \in edges} \psi_2(z_i, z_j|\theta, image)$$

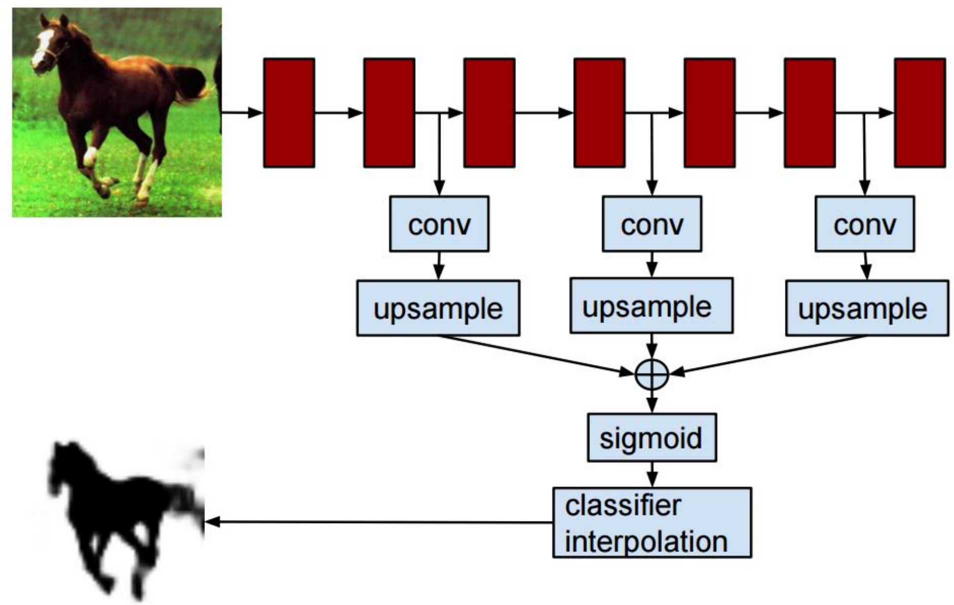
CNNs for segmentation



Slides: Svetlana Lazebnik

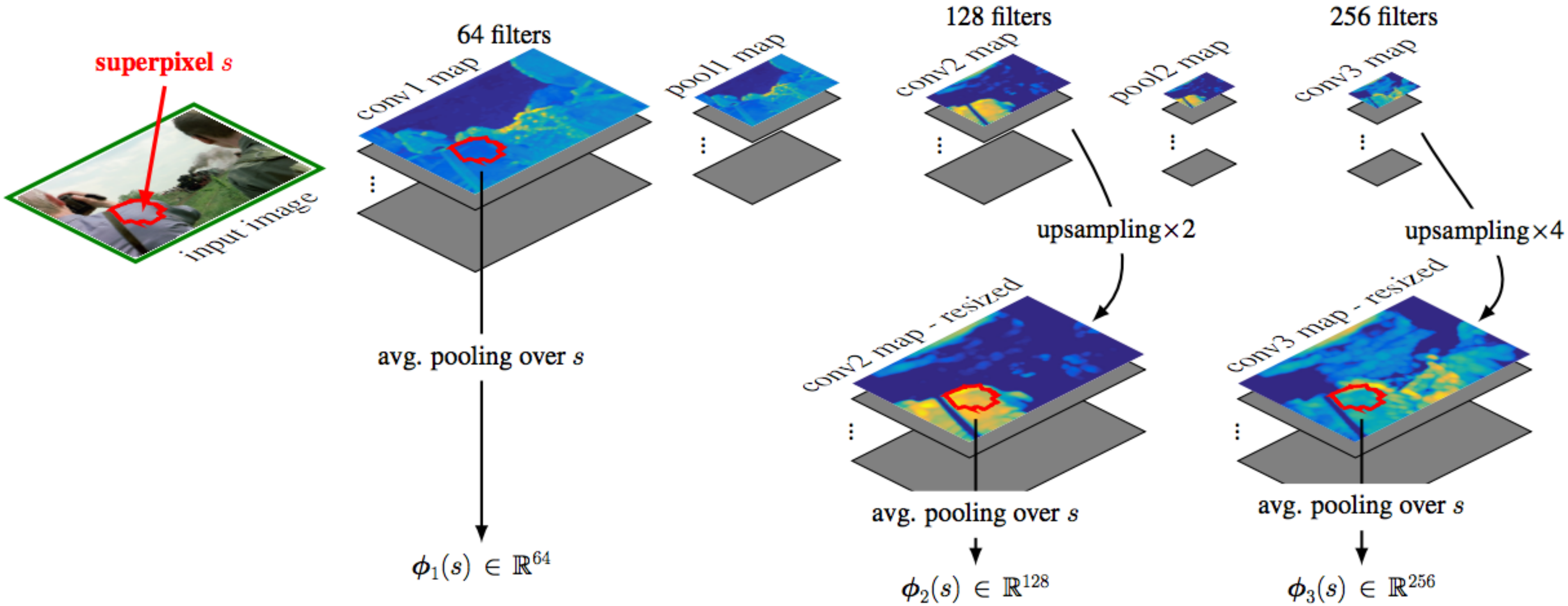
Hypercolumns

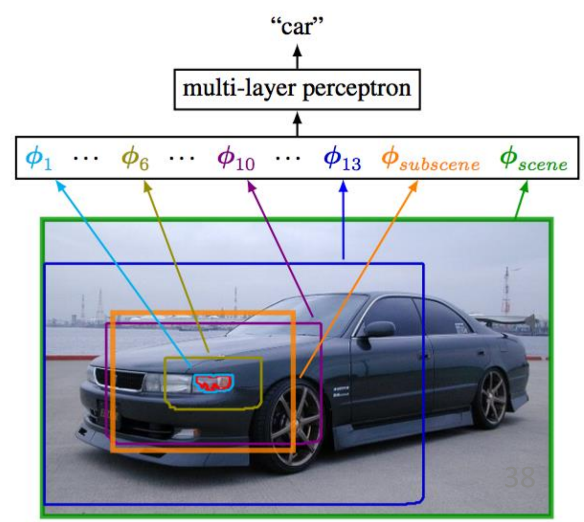




B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik,
[Hypercolumns for Object Segmentation and Fine-grained Localization](), CVPR 2015

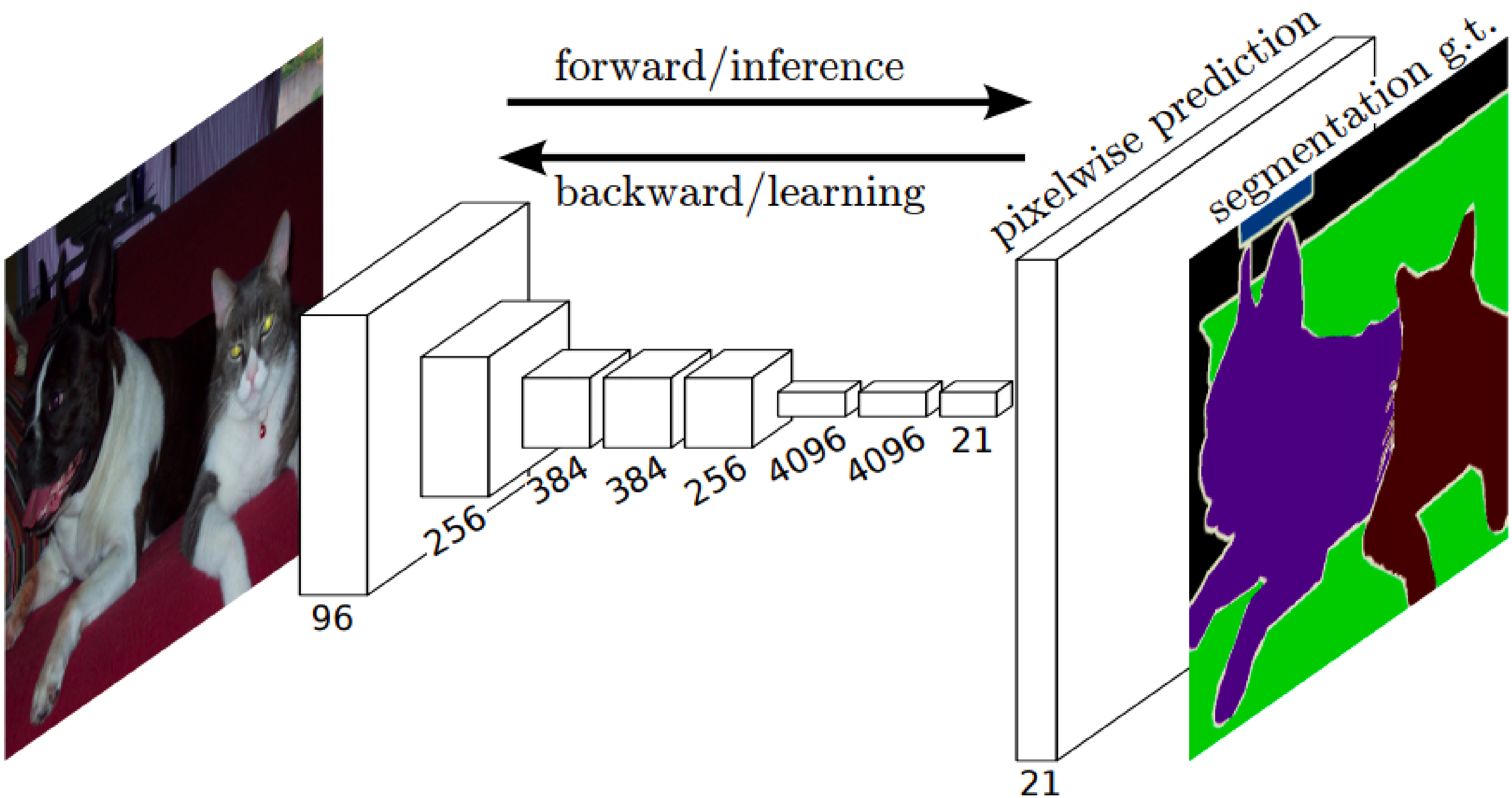
Zoom-out features





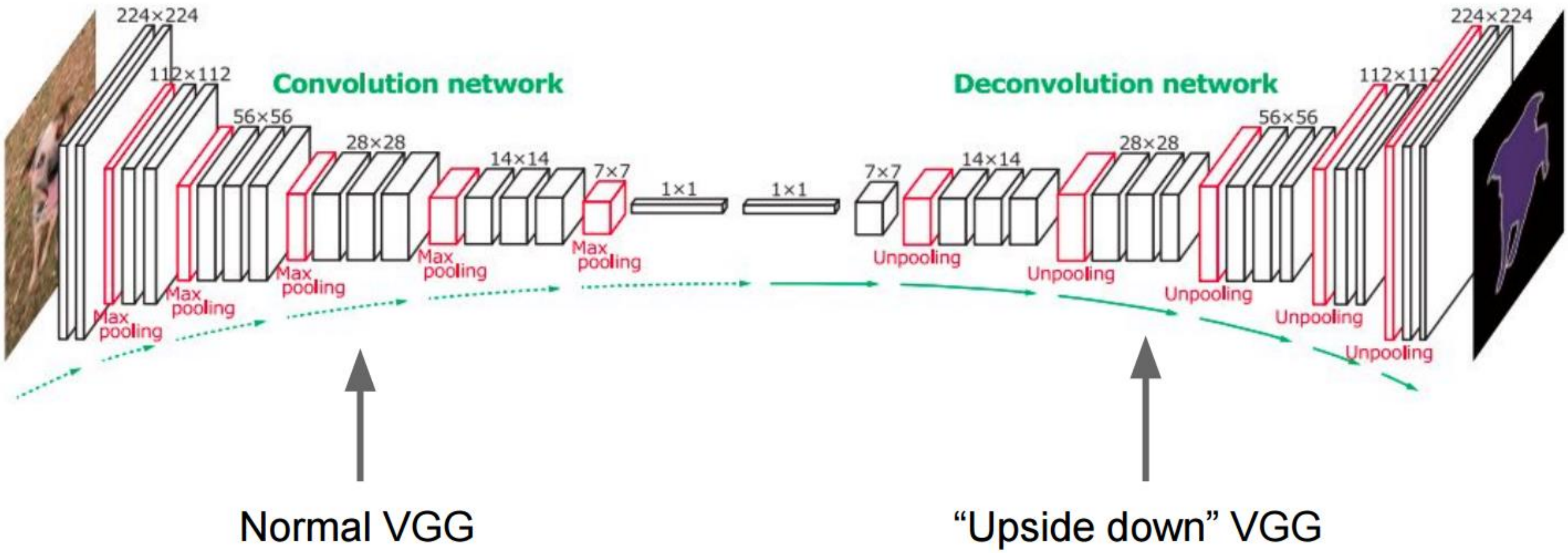
M. Mostajabi, P. Yadollahpour and G. Shakhnarovich, [Feedforward semantic segmentation with zoom-out features](), CVPR 2015

Fully Convolutional Networks (FCN)



J. Long, E. Shelhamer, and T. Darrell, [Fully Convolutional Networks for Semantic Segmentation](), CVPR 2015

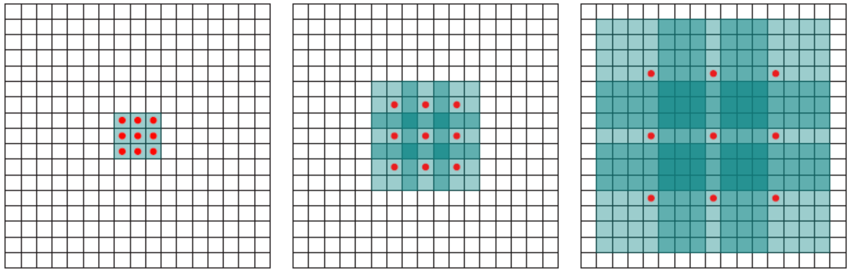
Learned upsampling architectures



H. Noh, S. Hong, and B. Han,
[Learning Deconvolution Network for Semantic Segmentation](), ICCV 2015

Dilated Convolutions

- Idea: instead of reducing spatial resolution of feature maps, use a large sparse filter
- Can aggregate contextual information across large scales without loss of resolution

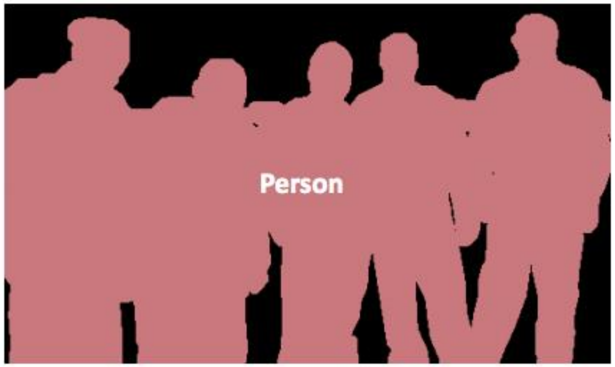


F. Yu and V. Koltun, [Multi-scale context aggregation by dilated convolutions](), ICLR 2016

Instance segmentation

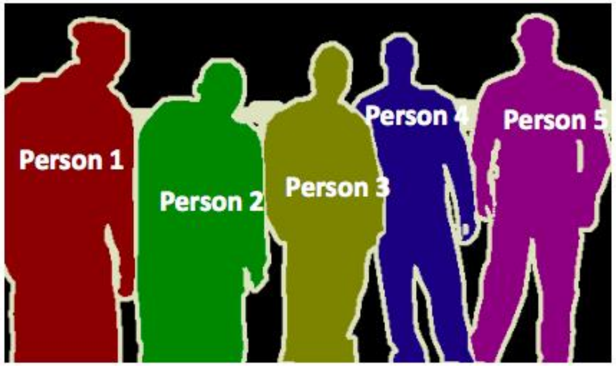
Object Detection

✓



Semantic Segmentation

✓



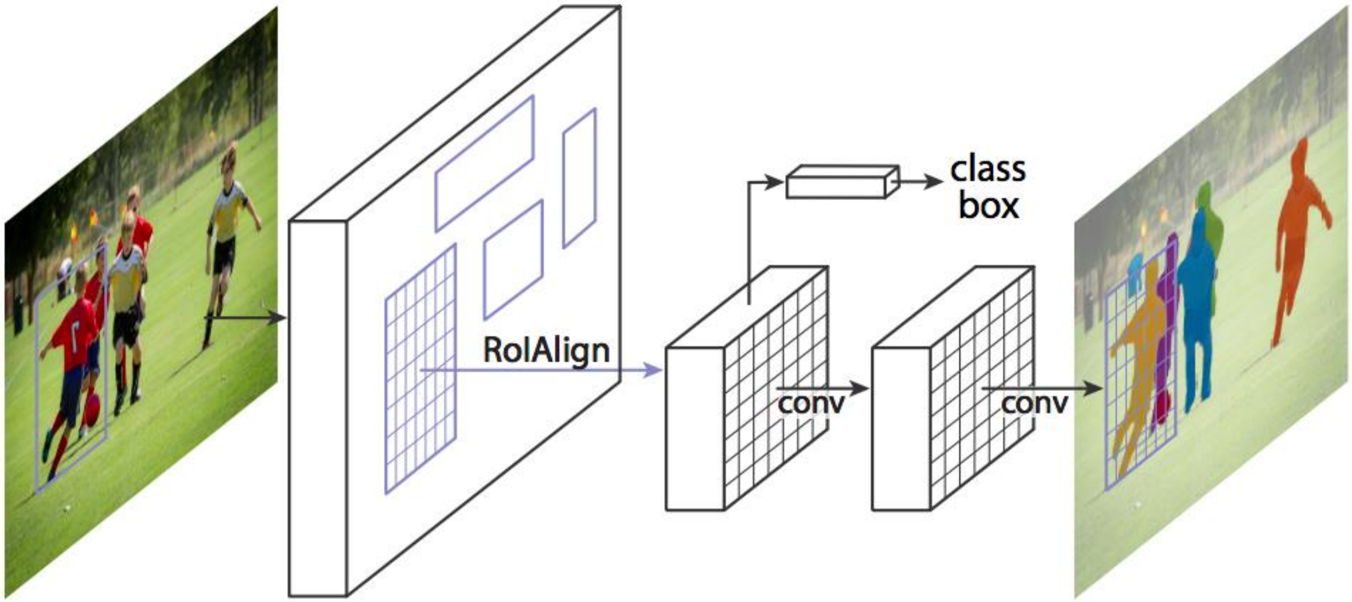
Instance Segmentation

?

Source: [Kaiming He]()

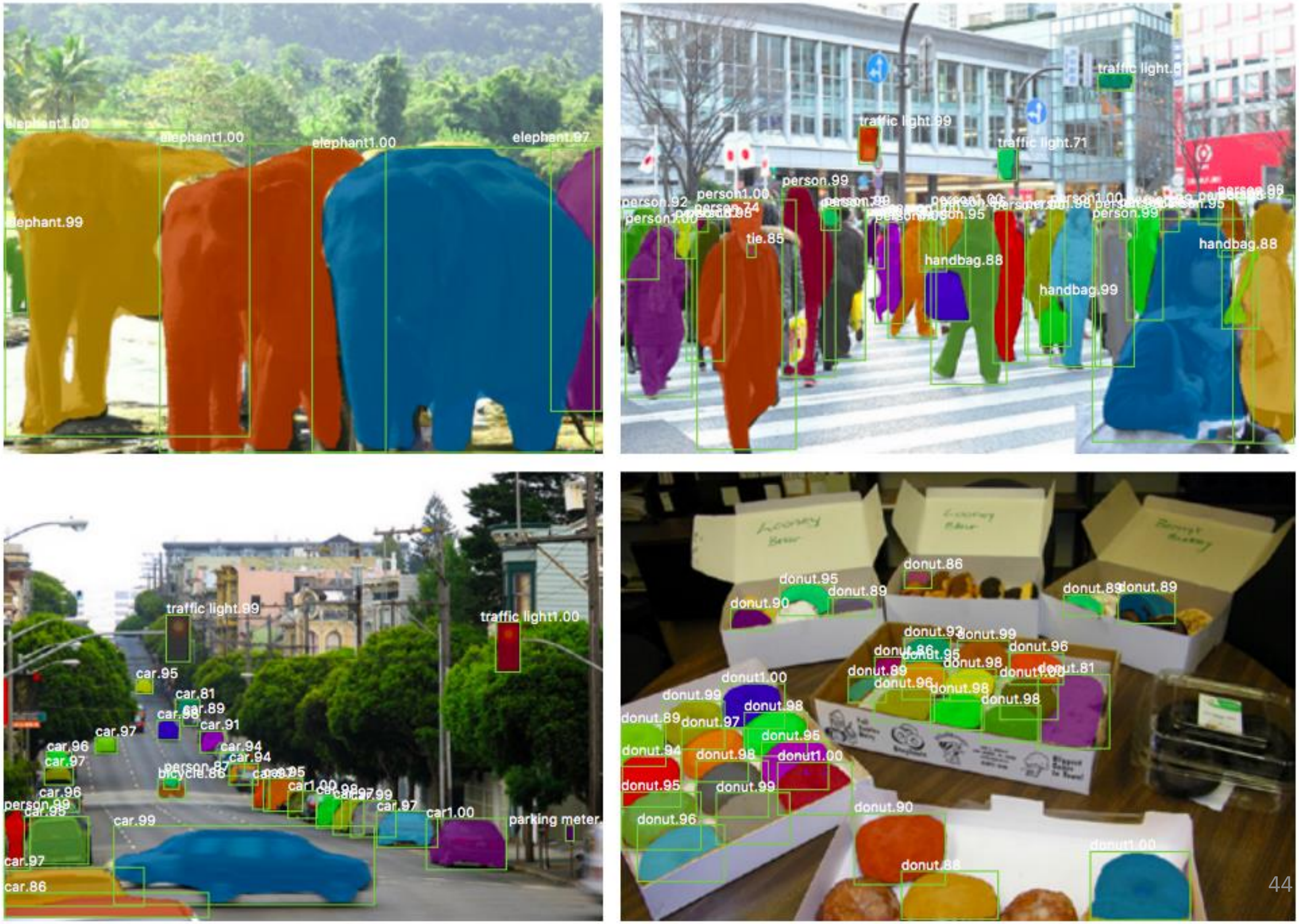
Mask R-CNN

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

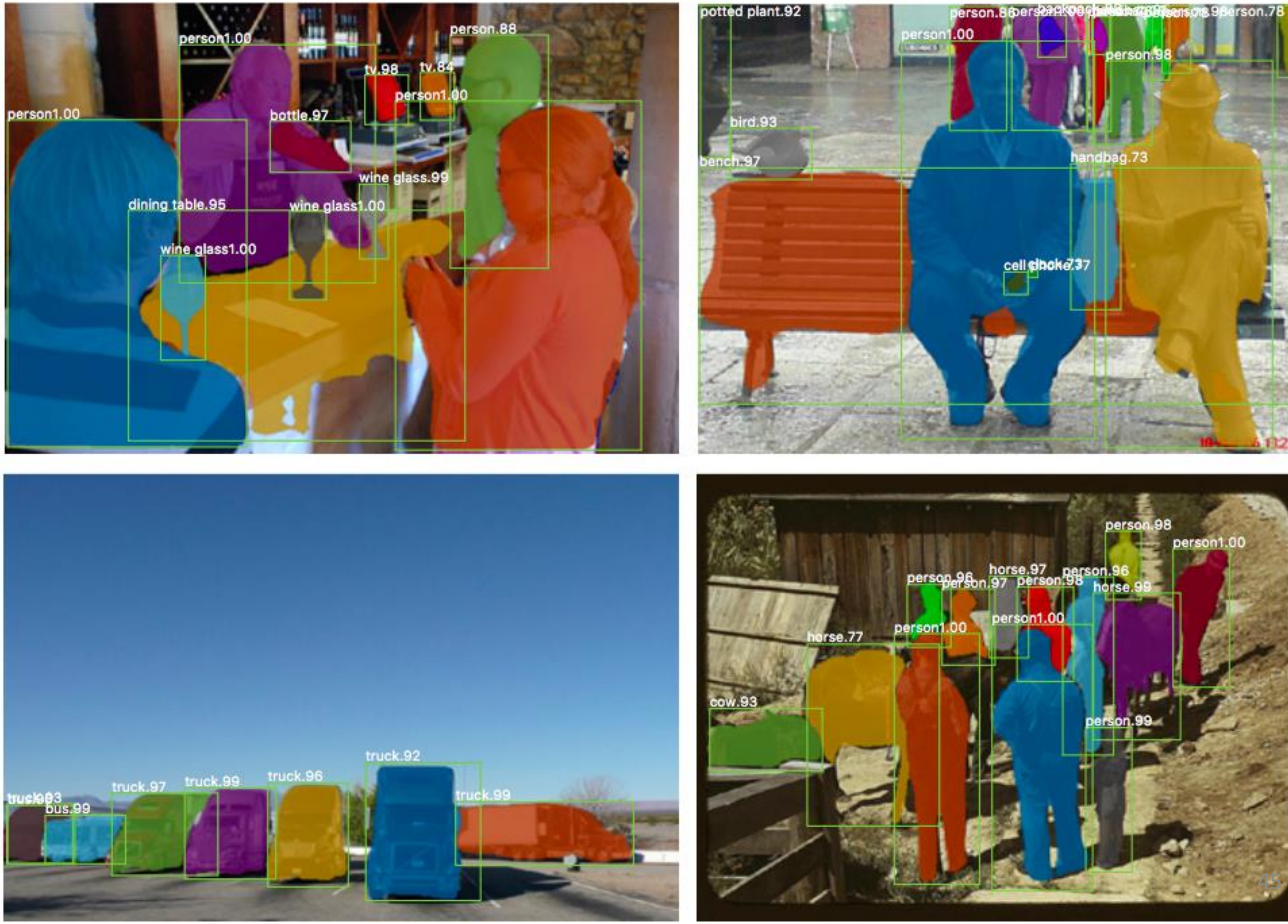


K. He, G. Gkioxari, P. Dollar, and R. Girshick,
[Mask R-CNN](), ICCV 2017 (Best Paper Award)

Example results



Example results



Some Segmentation Methods

- Bottom-up vs. Top-down
- Supervised vs. Unsupervised
- Thresholding
- K-means
- Histogram-based
- Region growing
- PDE-based
  - Parametric
  - Level-sets
- Variational methods
- Edge Detection
- Line Detection
- Watershed
- Split & Merge
- Color Based
- Region Moments
- Motion Based
  - Optical Flow
- Similar Depth
- Active Contours
  - Snakes
  - Intelligent Scissors
- Graph Based
  - Min Cut
  - Normalized Cut
  - Graph Cut
  - Grab Cut
  - Object Cut
- Oversegmentation
  - SLIC
  - Turbo superpixels
  - Felzenswalb
- Deep Networks
  - Hypercolumns
  - Zoom-out features
  - Fully Convolutional Networks
  - Dilated Convolutions
  - Mask R-CNN