CMP717
Image Processing

Semantic Segmentation

Erkut Erdem
Hacettepe University
Computer Vision Lab (HUCVL)

Semantic Segmentation

- Joint recognition & segmentation
  - segmenting all the objects in a given image and identifying their visual categories
- aka scene parsing or image parsing
- Early studies aim at segmenting out a single object of a known category
  - Borenstein & Ullman, 2002, Liebe & Schiele, 2003, etc.
- More recent work depends on CNNs
  - Farabet et al., 2013, Pinheiro and Collobert, 2014, Long et al., 2015, Noh et al., 2015

Computer Vision Tasks

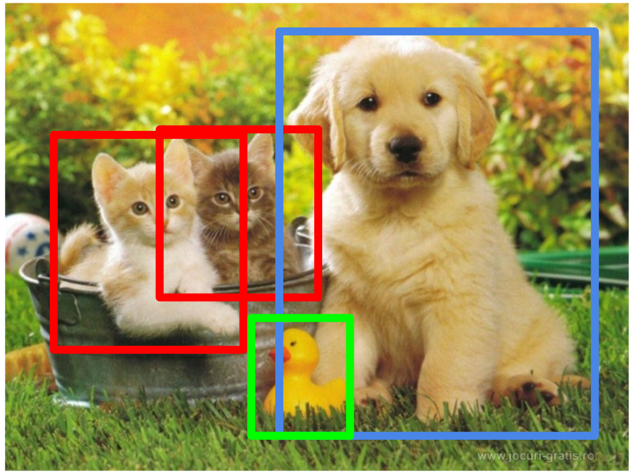
Classification

CAT

Classification + Localization

CAT

Object Detection



CAT, DOG, DUCK

Segmentation

CAT, DOG, DUCK

Single object

Multiple objects

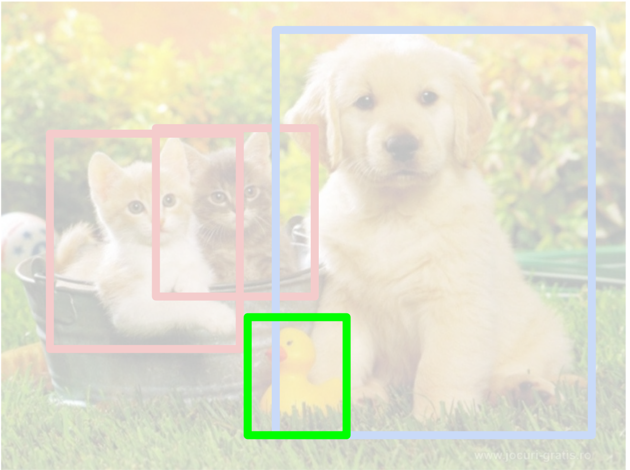
F.-F. Li, A. Karpathy and J. Johnson

Computer Vision Tasks

Classification

Classification + Localization

Object Detection



Segmentation

Today

F.-F. Li, A. Karpathy and J. Johnson

Semantic Segmentation

Label every pixel!

Don’t differentiate instances (cows)

Classic computer vision problem

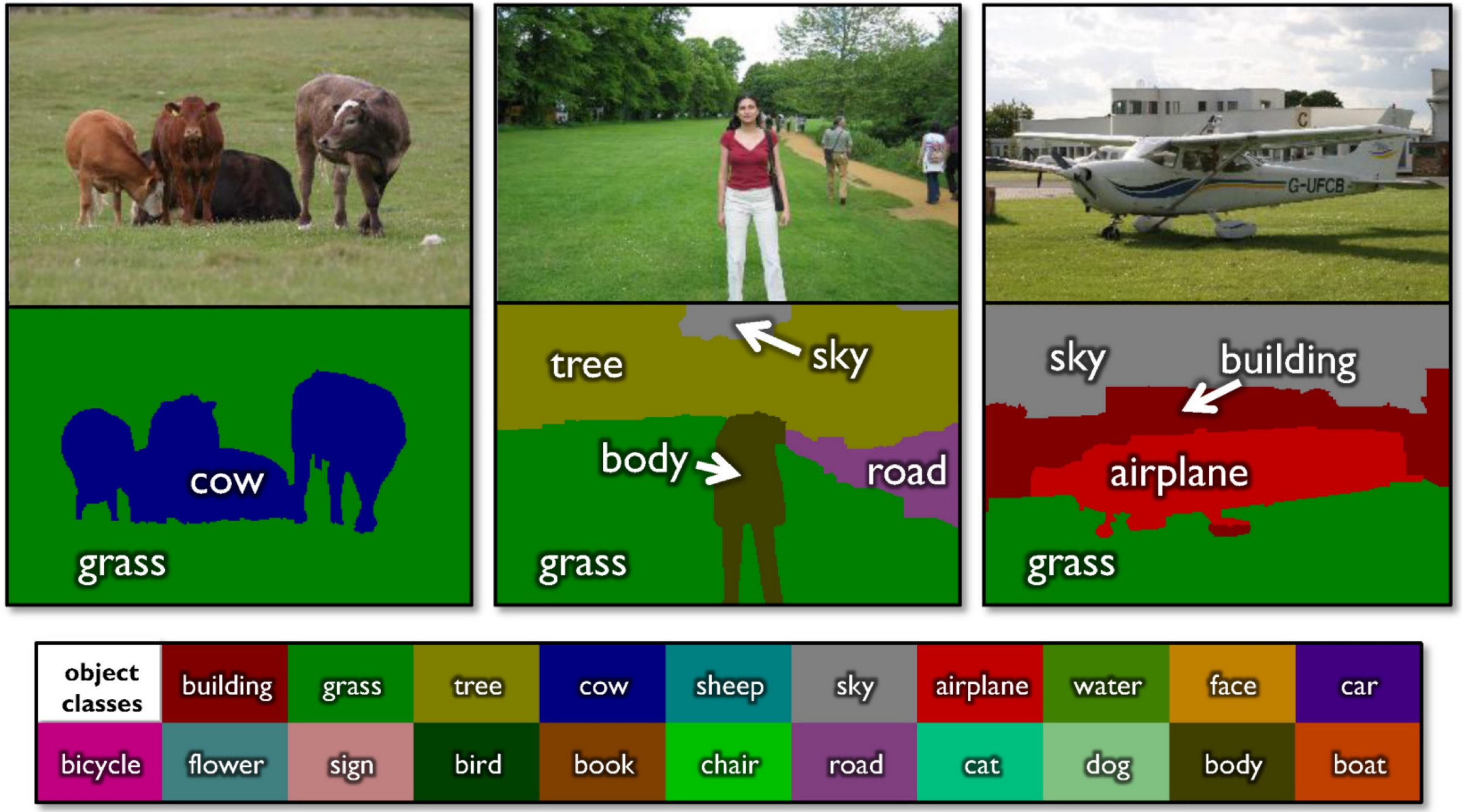


Figure credit: Shotton et al, “TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context”, IJCV 2007

Instance Segmentation

Detect instances, give category, label pixels

“simultaneous detection and segmentation” (SDS)

Lots of recent work (MS-COCO)

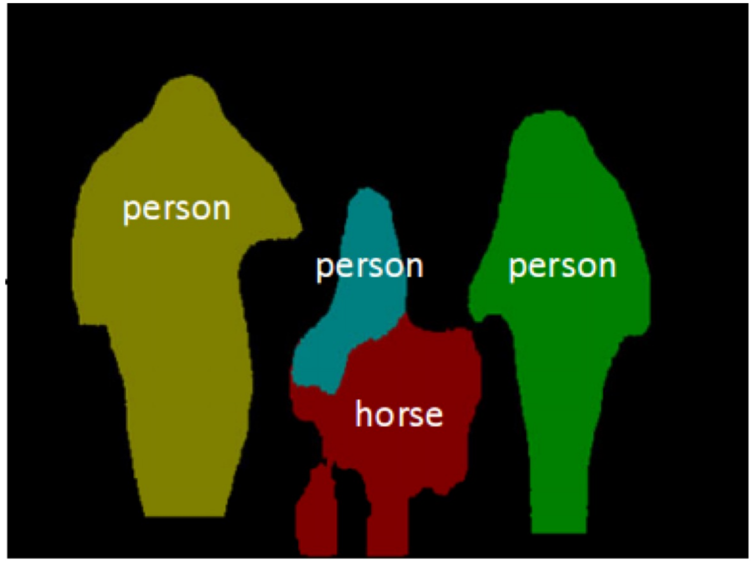
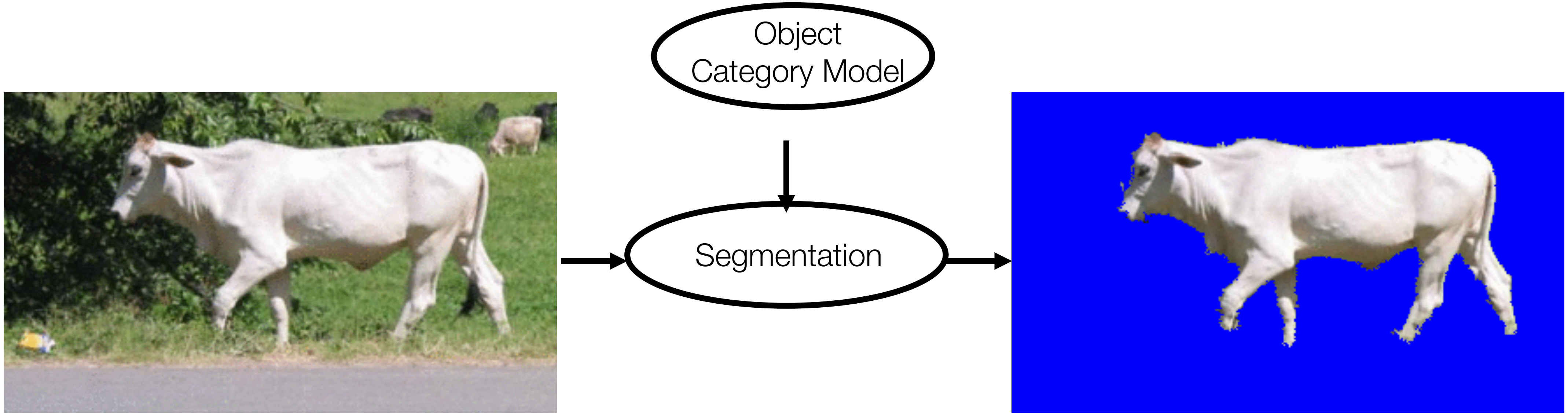


Figure credit: Dai et al, “Instance-aware Semantic Segmentation via Multi-task Network Cascades”, arXiv 2015

Early Studies of Semantic Segmentation

- Given an image and object category, to segment the object



Cow Image

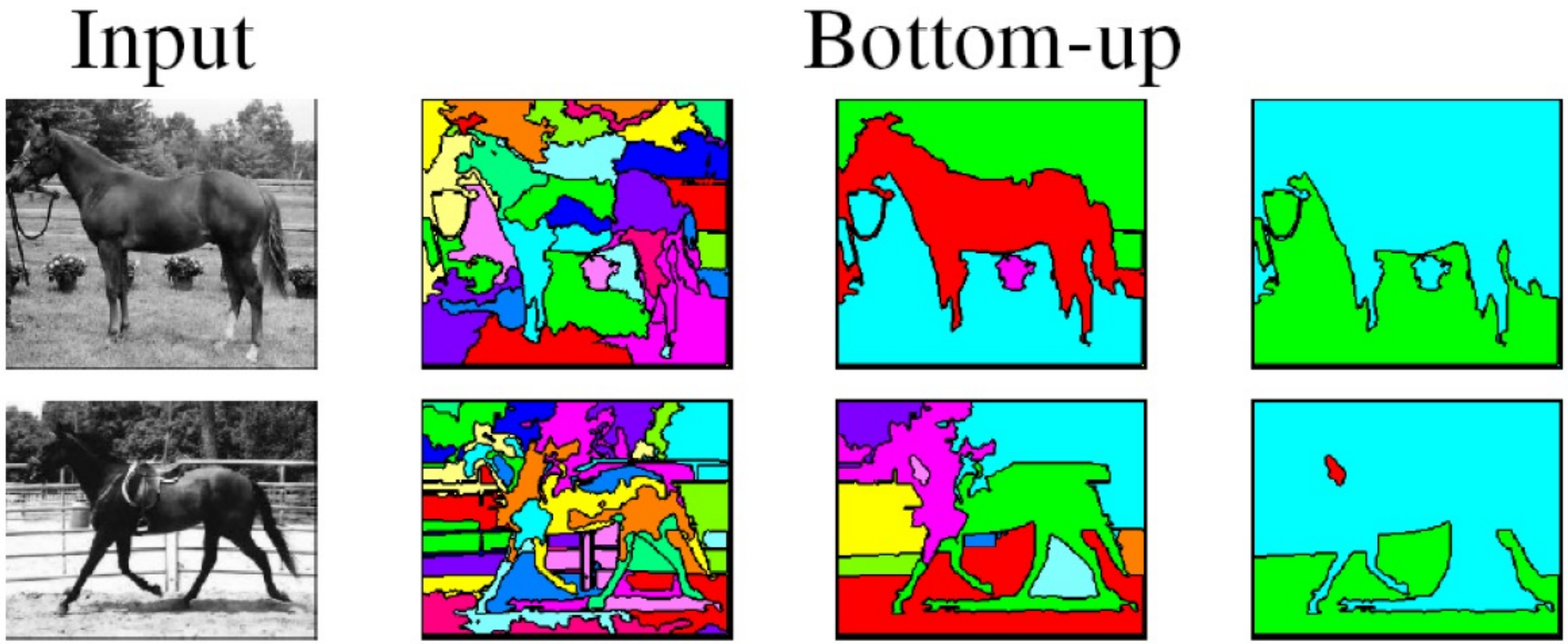
Segmented Cow

- Segmentation should (ideally) be
  - shaped like the object e.g. cow-like
  - obtained efficiently in an unsupervised manner
  - able to handle self-occlusion

M. P. Kumar

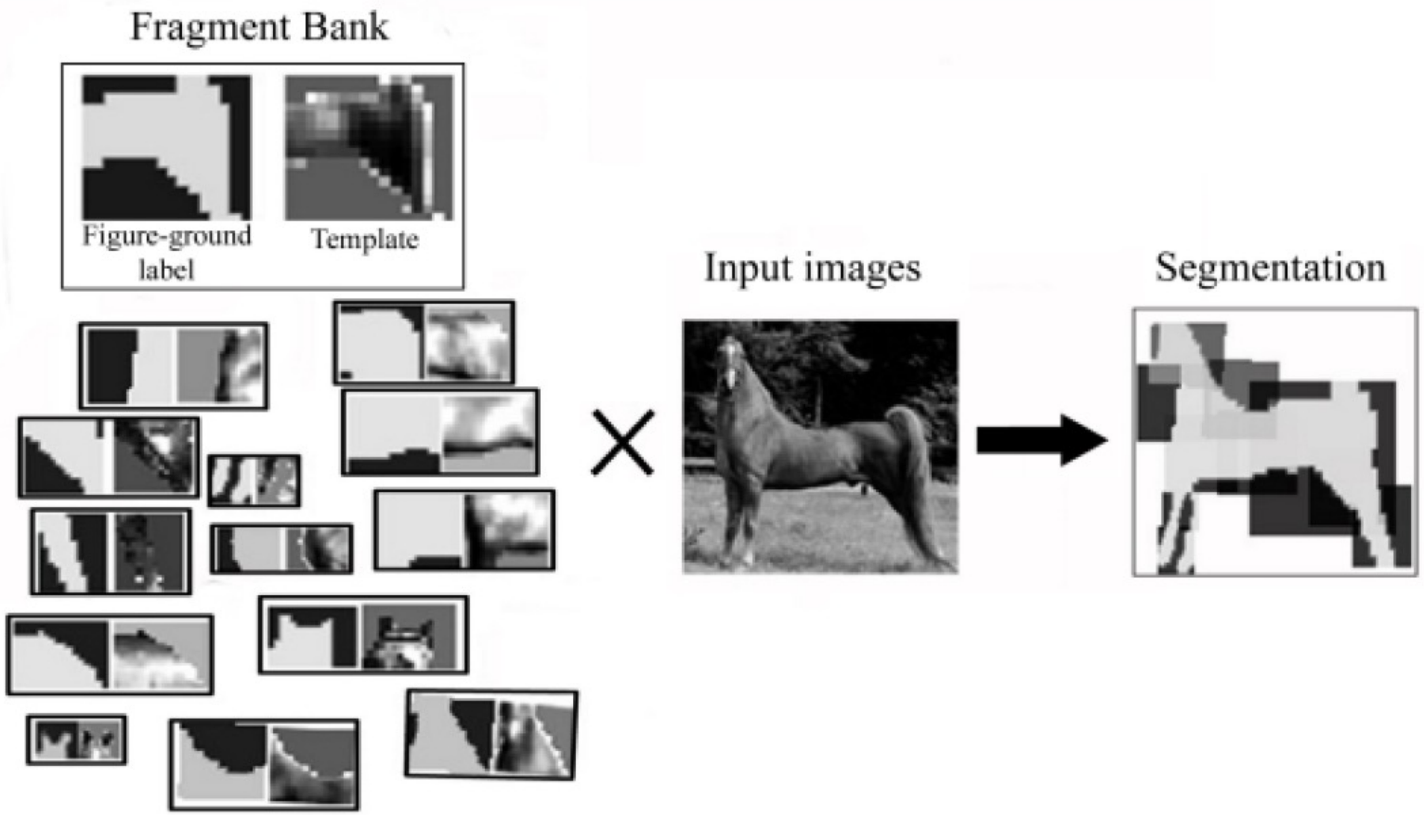
Early Studies of Semantic Segmentation

Using Normalized Cuts, Shi & Malik, 1997



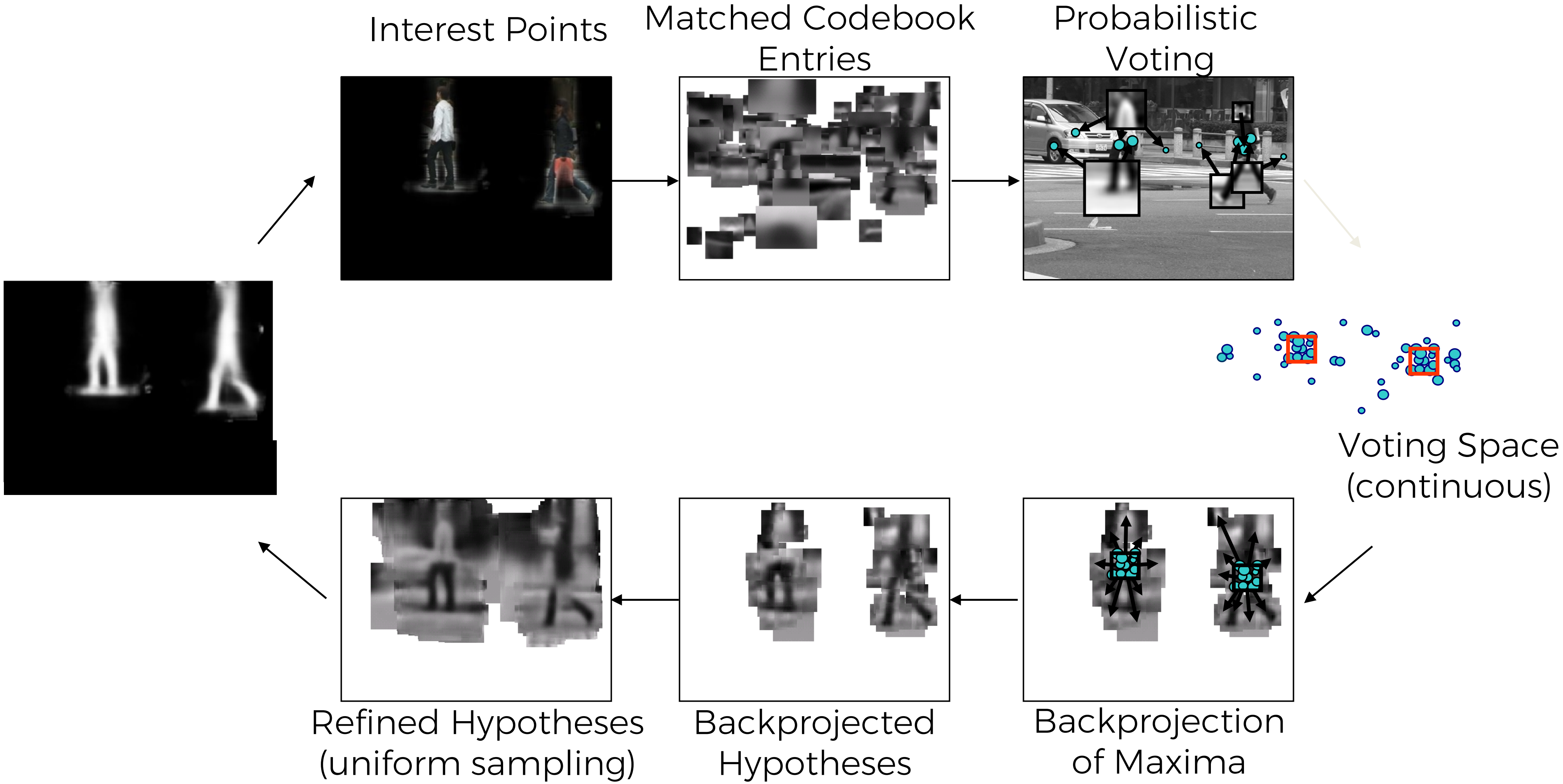
R. Fergus

Jigsaw approach: Borenstein and Ullman, 2002



R. Fergus

Implicit Shape Model - Liebe and Schiele, 2003



R. Fergus

Random Fields for segmentation

I = Image pixels (observed)
h = foreground/background labels (hidden) – one label per pixel
θ = Parameters

$$\underbrace{p(h \mid I, \theta)}_{\text{Posterior}}$$

Random Fields for segmentation

I = Image pixels (observed)
h = foreground/background labels (hidden) – one label per pixel
θ = Parameters

$$\underbrace{p(h \mid I,\theta)}_{\text{Posterior}} \propto \underbrace{p(I,h \mid \theta)}_{\text{Joint}} = \underbrace{p(I \mid h,\theta)}_{\text{Likelihood}}\underbrace{p(h \mid \theta)}_{\text{Prior}}$$

1. Generative approach models joint
   → Markov random field (MRF)

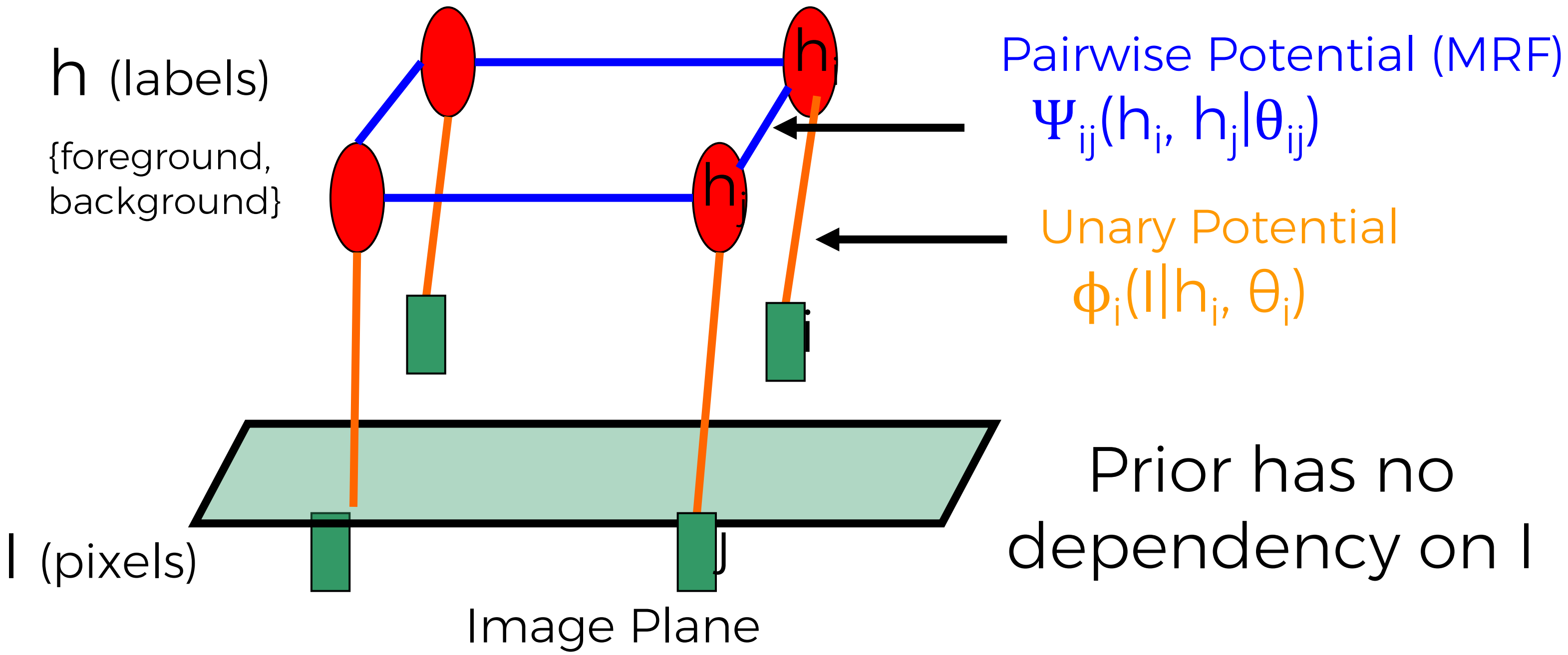
2. Discriminative approach models posterior directly
   → Conditional random field (CRF)

R. Fergus

Generative Markov Random Field

$$p(h, I \mid \theta) = \boxed{p(I \mid h, \theta)}\,\boxed{p(h \mid \theta)}$$

$$= \frac{1}{Z(\theta)} \left[\underbrace{\prod_i \phi_i(I \mid h_i, \theta_i)}_{\text{Likelihood}} \underbrace{\prod_{ij} \psi_{ij}(h_i, h_j \mid \theta_{ij})}_{\text{MRF Prior}} \right]$$



Prior has no dependency on I

R. Fergus

Conditional Random Field

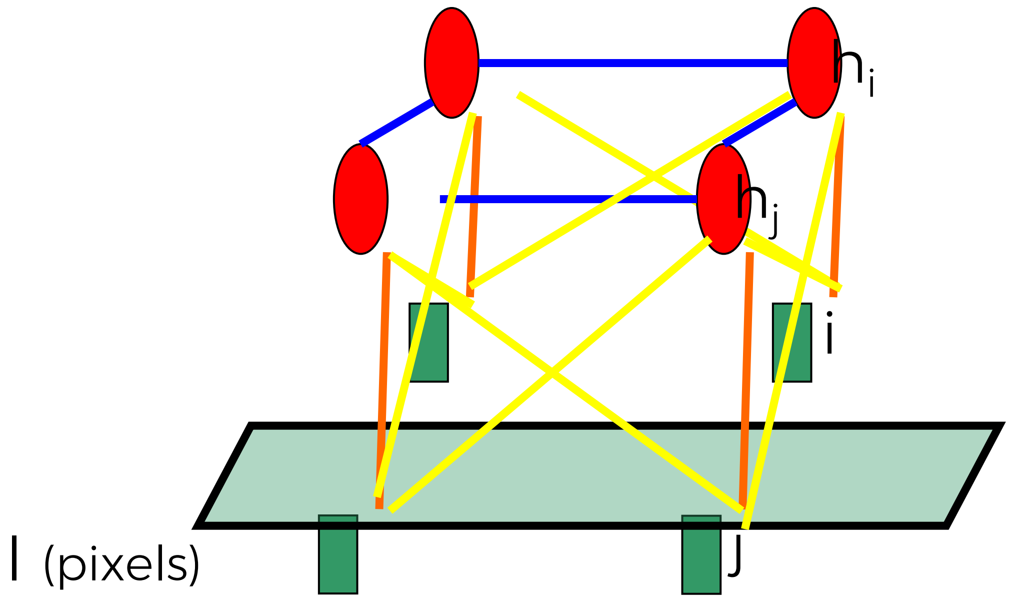
Discriminative approach

Lafferty, McCallum and Pereira 2001

$$p(h \mid I, \theta) = \frac{1}{Z(I,\theta)} \left[\prod_i \underbrace{\phi_i(h_i, I \mid \theta_i)}_{\text{Unary}} \prod_{ij} \underbrace{\psi_{ij}(h_i, h_j, I \mid \theta_{ij})}_{\text{Pairwise}} \right]$$

- Dependency on I allows introduction of pairwise terms that make use of image.
- For example, neighboring labels should be similar only if pixel colors are similar → Contrast term

e.g Kumar and Hebert 2003



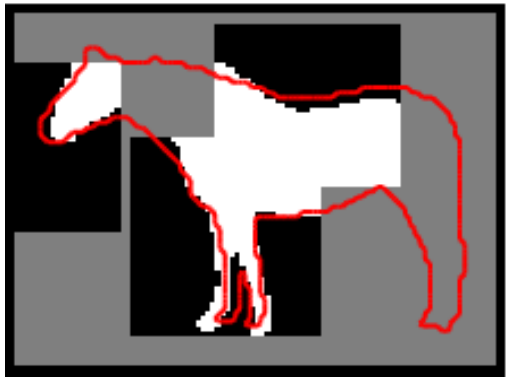
R. Fergus

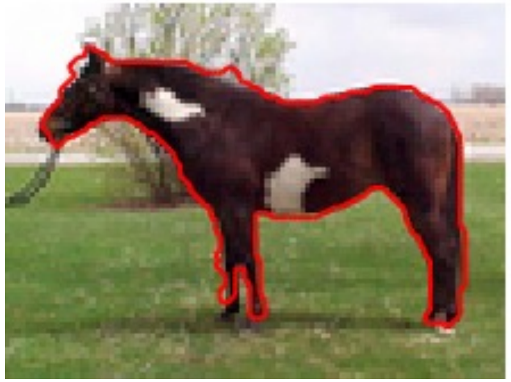
Levin & Weiss [ECCV 2006]

$$E(h;I) = \sum_i \lambda_i \left| h - h_{F_i,I} \right| + \sum_{ij} w(i,j) \left| h_i - h_j \right|$$

Consistency with fragments segmentation

Segmentation alignment with image edges



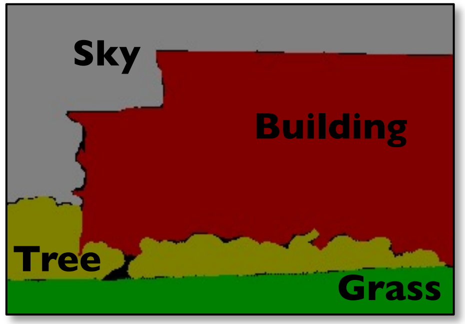


Resulting min-cut segmentation

R. Fergus

Semantic Segmentation

Joint Object recognition & segmentation

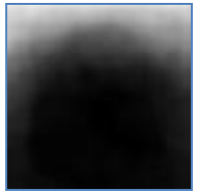


$$E(x,\omega) = \sum_i \theta_i(\omega, x_i) + \sum_i \theta_i(x_i) + \sum_i \theta_i(x_i) + \sum_{i,j} \theta_{ij}(x_i,x_j)$$

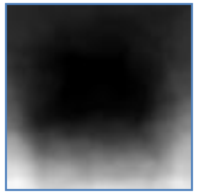
(color) (locatio
n) (class) (edge aware
Ising prior)

$x_i \in \{1,\ldots,K\}$ for K object classes

Location

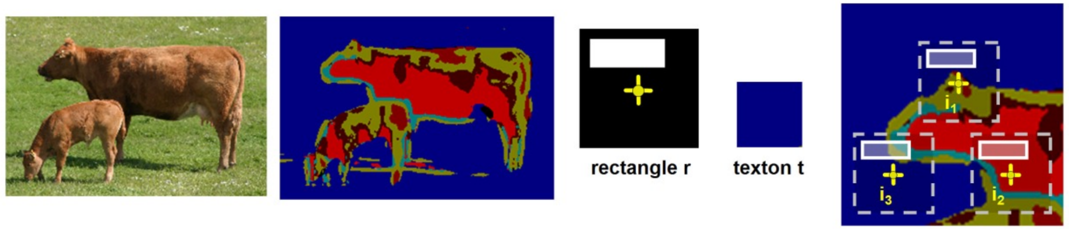


sky



grass

Class (boosted textons)

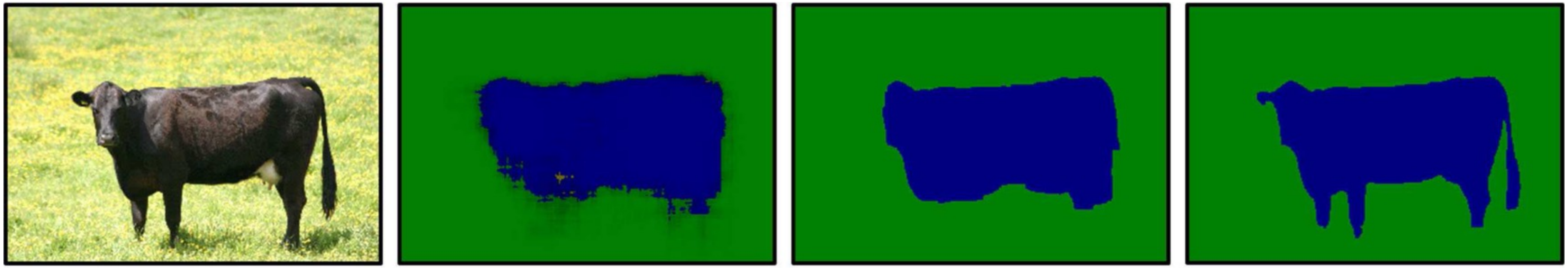


(a) Input image (b) Texton map (c) Feature pair = (r,t) (d) Superimposed rectangles

[TextonBoost; Shotton et al, '06]

C. Rother

Semantic Segmentation
Joint Object recognition & segmentation



(a) (b) 69.6% (c) 70.3% (d) 72.2%

Class + location | + edges | + color

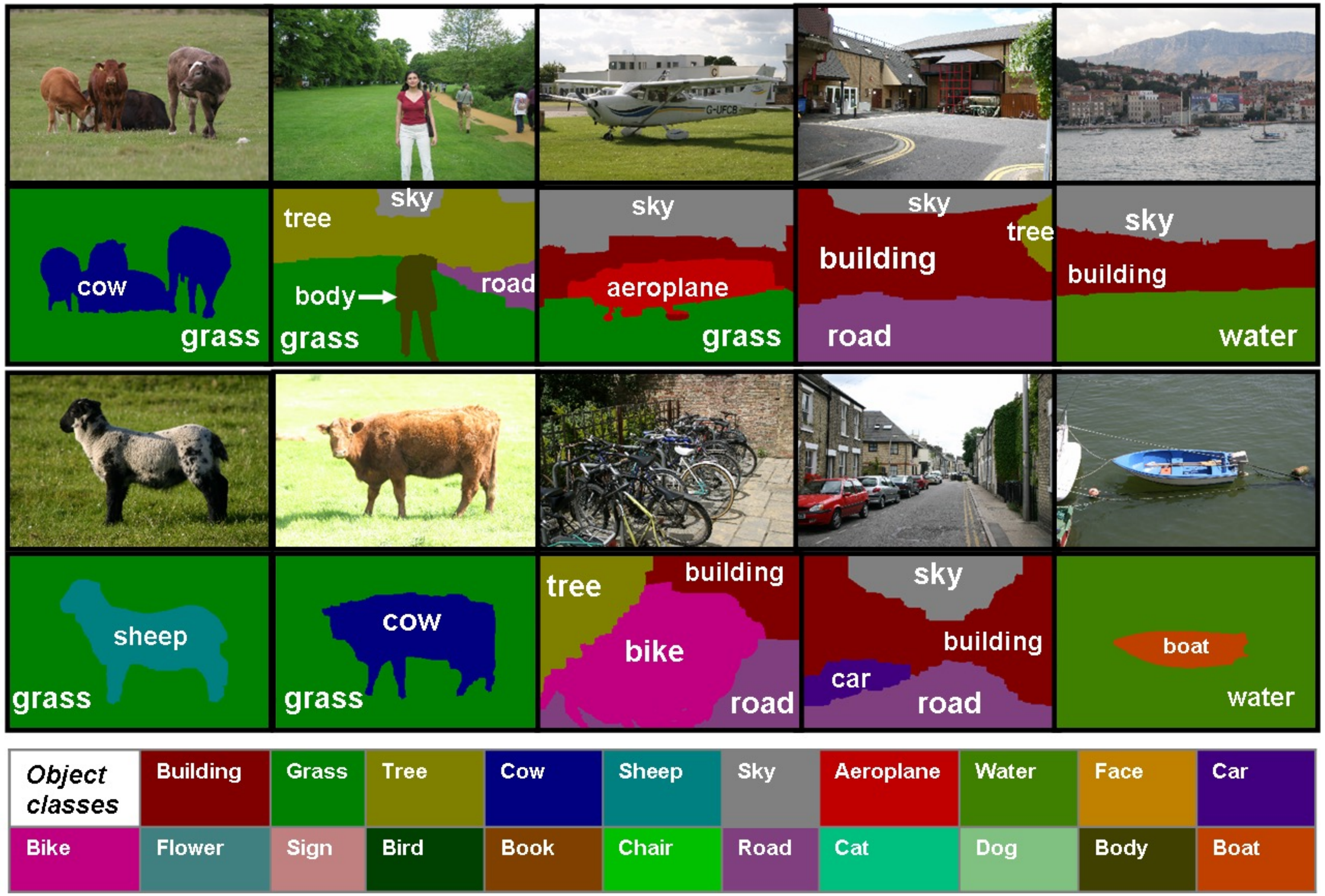
[TextonBoost; Shotton et al, '06]

C. Rother

Semantic Segmentation

Joint Object recognition & segmentation

Good results …



[TextonBoost; Shotton et al, '06]

C. Rother

Semantic Segmentation
Joint Object recognition & segmentation

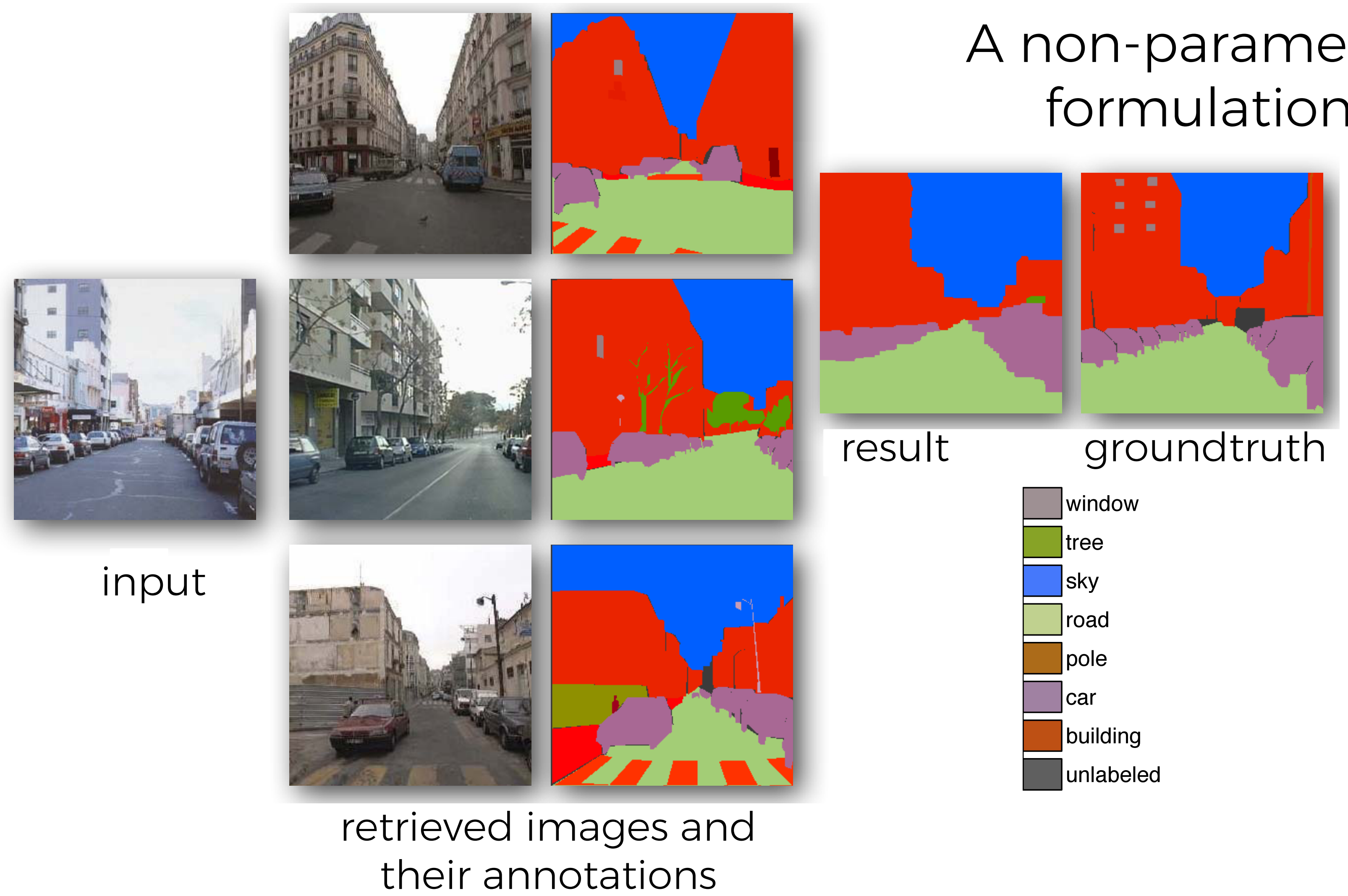
Failure cases…

[TextonBoost; Shotton et al, '06]

C. Rother

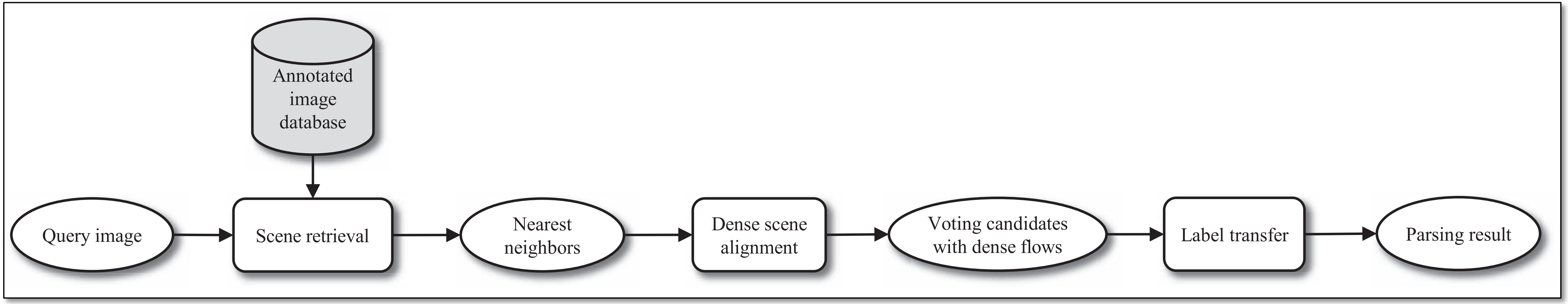
Nonparametric Scene Parsing via Label Transfer (Liu et al. TPAMI'12)

A non-parametric formulation



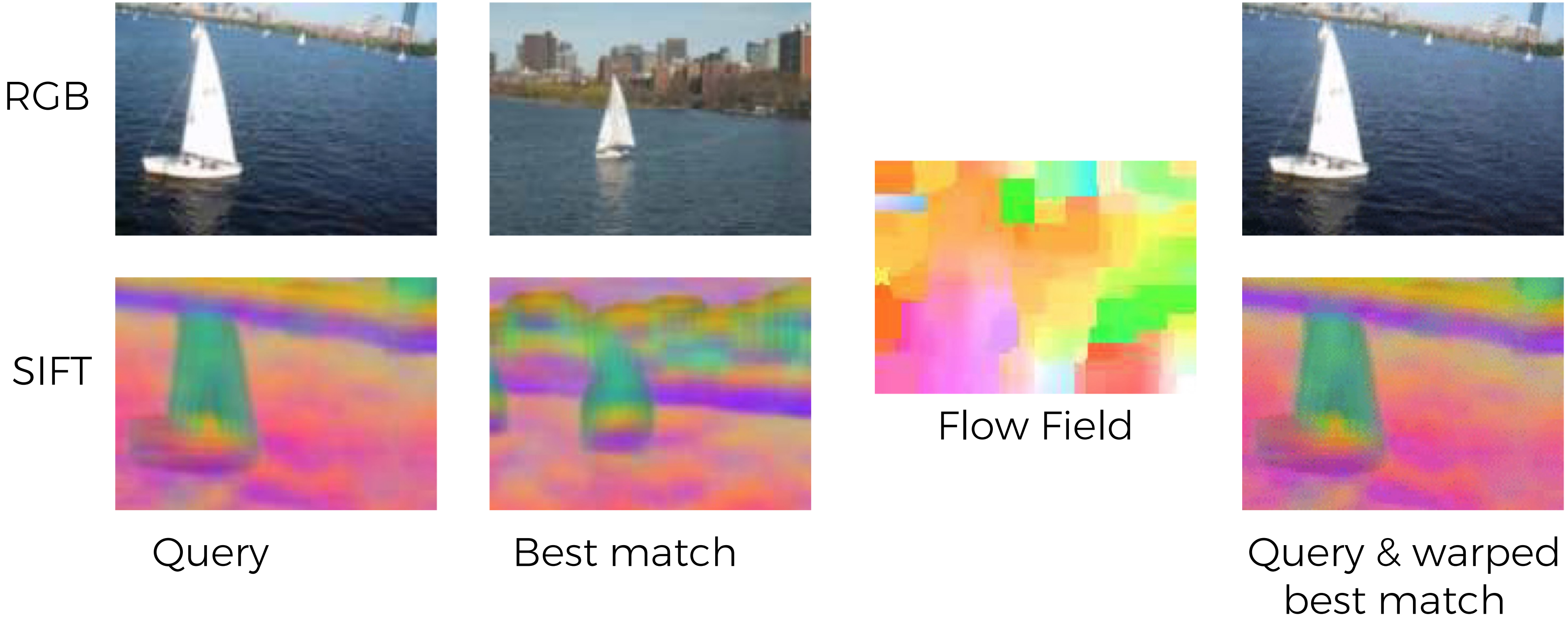
Nonparametric Scene Parsing via Label Transfer

- Framework consists of three main modules:
  1. Scene retrieval: finding nearest neighbors (k-NN approach)
  2. Dense scene alignment: dense scene matching (SIFT Flow)
  3. Label transfer: using a MRF model to label input image



Dense Scene Alignment via SIFT Flow

- SIFT Flow (Liu et al., ECCV 2008)
  - Finds semantically meaningful correspondences among two images by matching local SIFT descriptors



Dense Scene Alignment via SIFT Flow

- SIFT Flow (Liu et al., ECCV 2008)
  - Finds semantically meaningful correspondences among two images by matching local SIFT descriptors

$$E(\mathbf{w}) = \sum_{\mathbf{p}} \min(\|s_1(\mathbf{p}) - s_2(\mathbf{p}+\mathbf{w}(\mathbf{p}))\|_1, t) +$$

data term

$$\sum_{\mathbf{p}} \eta(|u(\mathbf{p})| + |v(\mathbf{p})|) +$$

small displacement term

$$\sum_{(\mathbf{p},\mathbf{q})\in\varepsilon} \min(\lambda|u(\mathbf{p}) - u(\mathbf{q})|, d) +$$
$$\min(\lambda|v(\mathbf{p}) - v(\mathbf{q})|, d),$$

smoothness term

w(**p**)=(u(**p**), v(**p**)) : flow vector at point p

Label Transfer

- A set of voting candidates $\{s_i, c_i, w_i\}_{i=1:M}$ is obtained from the retrieved images with s_i, c_i, and w_i denoting the SIFT image, annotation, and SIFT flow field of the *i*th voting candidate.
- A probabilistic MRF model is built to integrate
  - multiple category labels,
  - prior object (category) information
  - spatial smoothness of category labels

$$
\begin{aligned}
-\log P\big(c|I, s, \{s_i, c_i, \mathbf{w}_i\}\big) &= \sum_{\mathbf{p}} \psi\big(c(\mathbf{p}); s, \{s_i'\}\big) \\
&+ \alpha \sum_{\mathbf{p}} \lambda\big(c(\mathbf{p})\big) + \beta \sum_{\{\mathbf{p},\mathbf{q}\}\in\varepsilon} \phi\big(c(\mathbf{p}), c(\mathbf{q}); I\big) + \log Z
\end{aligned}
$$

Label Transfer

- Likelihood term:

$$\psi\big(c(\mathbf{p}) = l\big) = \begin{cases} \min_{i \in \Omega_{\mathbf{p},l}} \|s(\mathbf{p}) - s_i(\mathbf{p} + \mathbf{w}(\mathbf{p}))\|, & \Omega_{\mathbf{p},l} \neq \emptyset, \\ \tau, & \Omega_{\mathbf{p},l} = \emptyset, \end{cases}$$

- $\Omega_{\mathbf{p},l} = \{i; c_i(\mathbf{p} + \mathbf{w}(\mathbf{p})) = l\}$ where *l=1,...,L* indicates the index set of the voting candidates whose label is *l* after being warped to pixel p.
- τ is set to be the value of the maximum difference of SIFT feature:

$$\tau = \max_{s_1, s_2, \mathbf{p}} \|s_1(\mathbf{p}) - s_2(\mathbf{p})\|$$

Label Transfer

- Prior term :

$$\lambda\big(c(\mathbf{p}) = l\big) = -\log \mathrm{hist}_l(\mathbf{p})$$

- The prior probability that the object category *l* appears at pixel **p**.
  - obtained by counting the occurrence of each object category at each location in the training set
  - Location prior

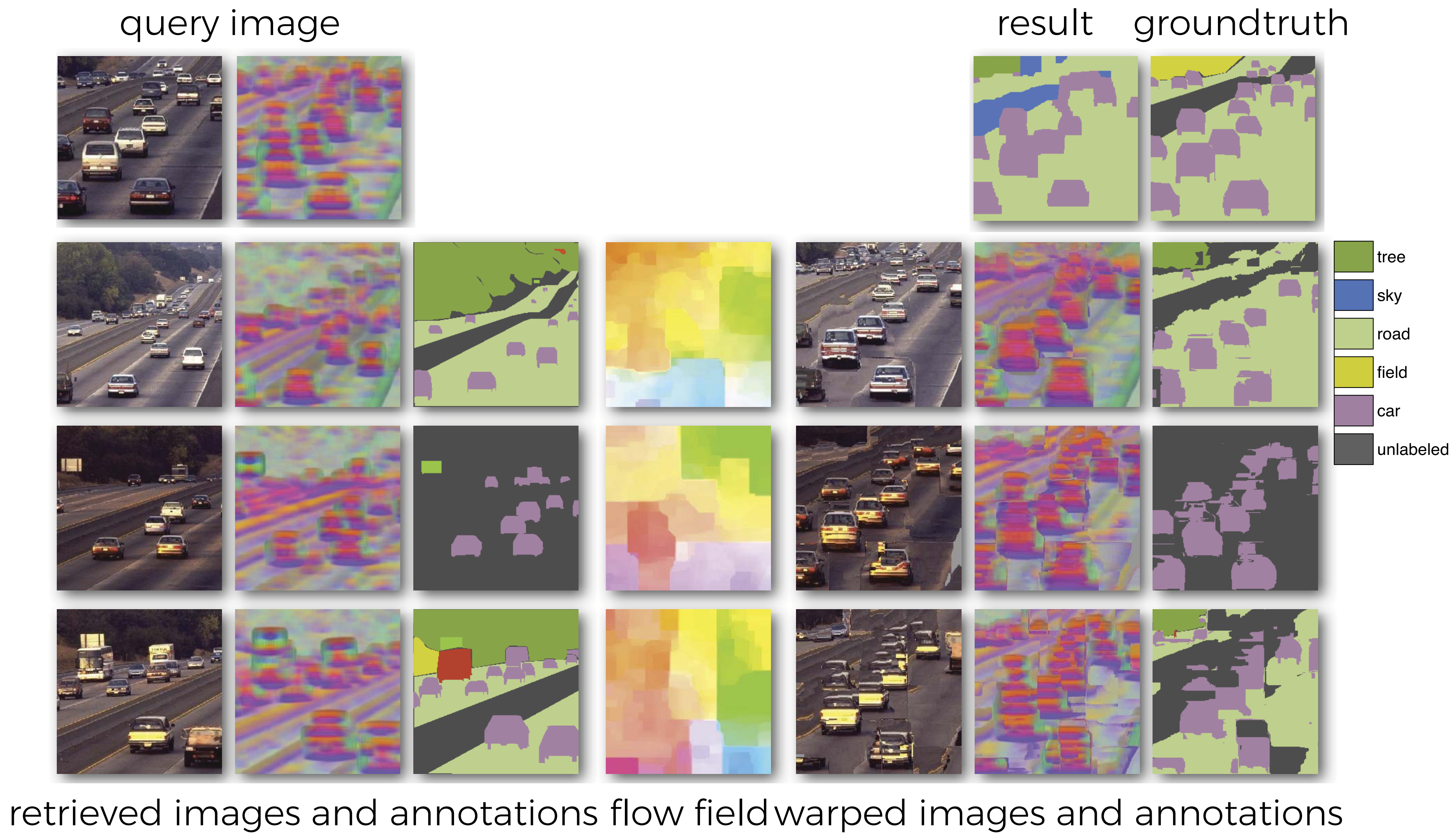
Label Transfer

- Spatial smoothness term:

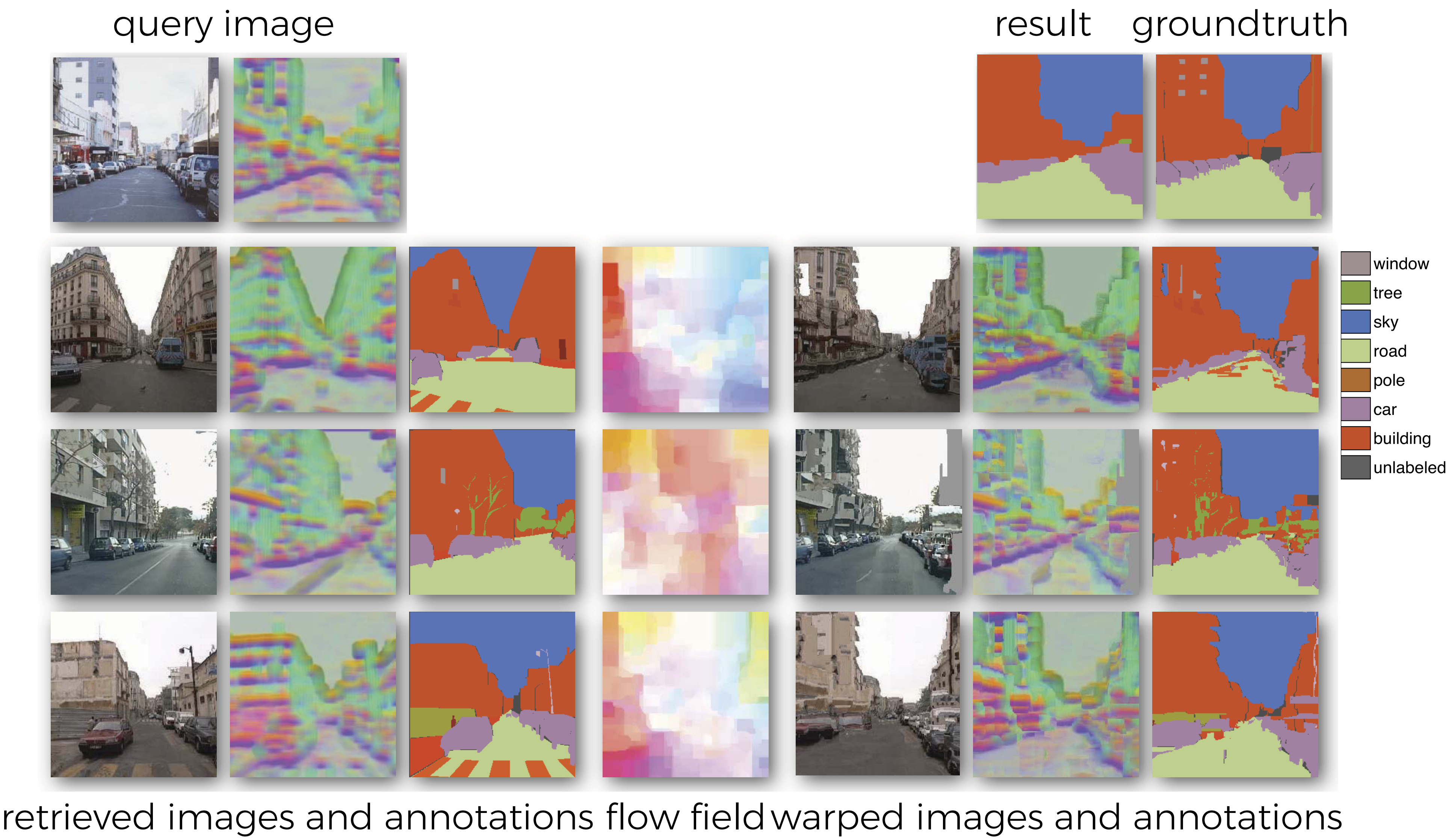
$$\phi\big(c(\mathbf{p}), c(\mathbf{q})\big) = \delta[c(\mathbf{p}) \neq c(\mathbf{q})]\left(\frac{\xi + e^{-\gamma\|I(\mathbf{p})-I(\mathbf{q})\|^2}}{\xi + 1}\right)$$

- The neighboring pixels into having the same label with the probability depending on the image edges:
  - Stronger the contrast, the more likely it is that the neighboring pixels may have different labels.

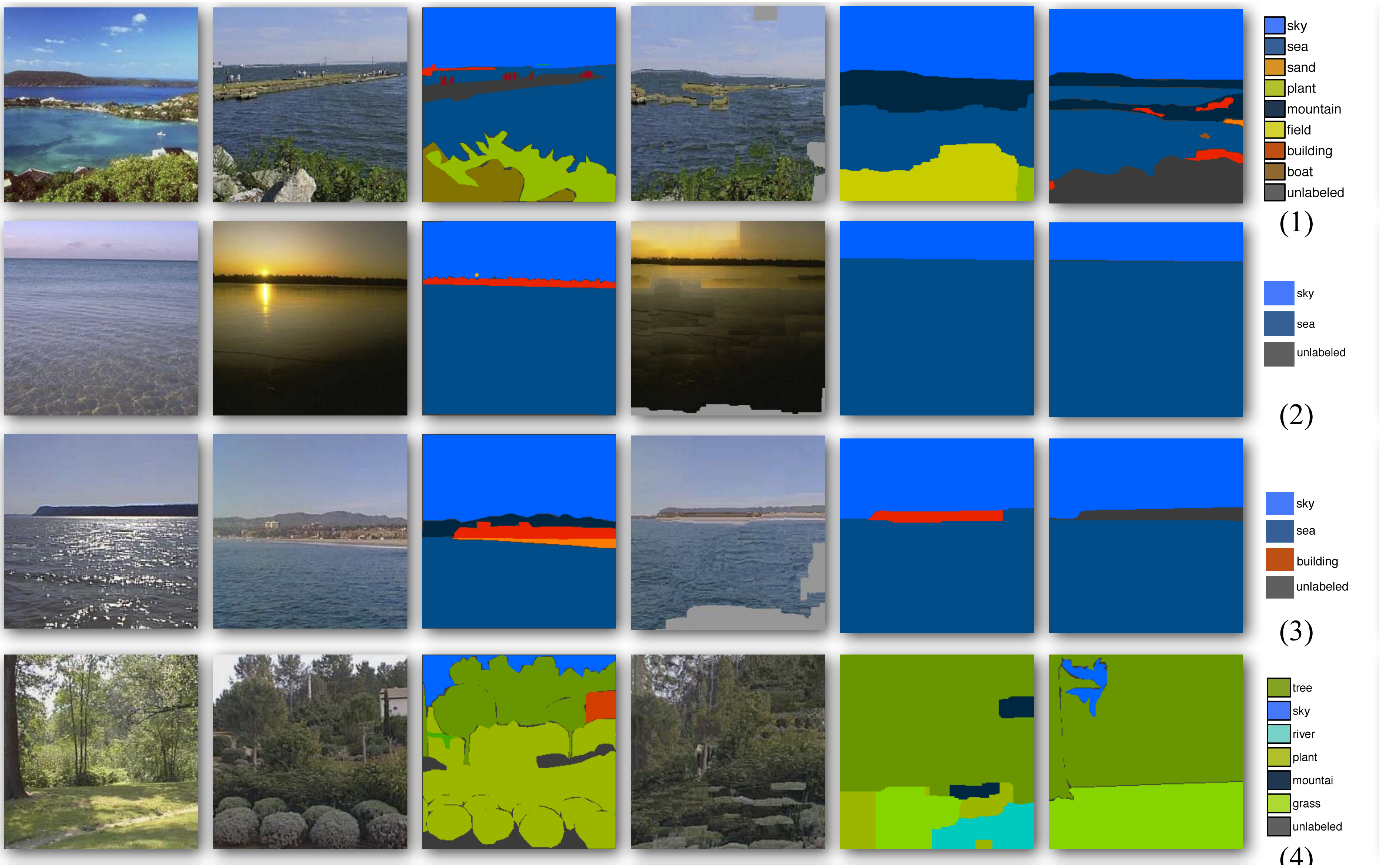
Parsing Results



Parsing Results



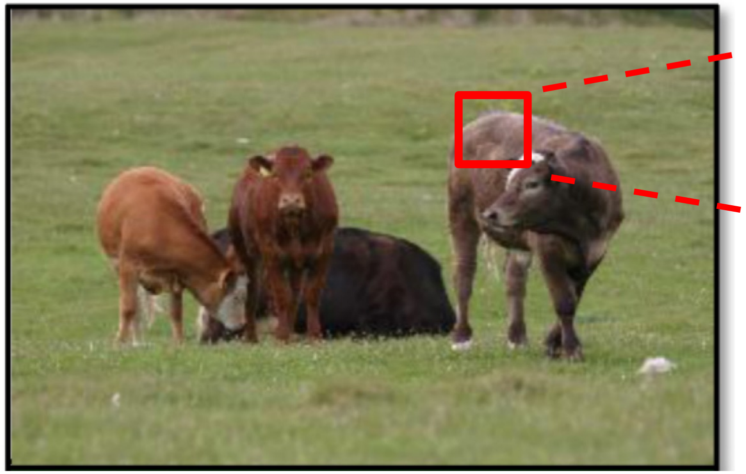
Parsing Results

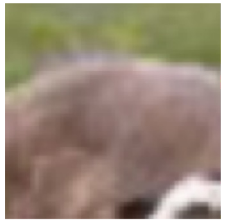


Deep Semantic Segmentation

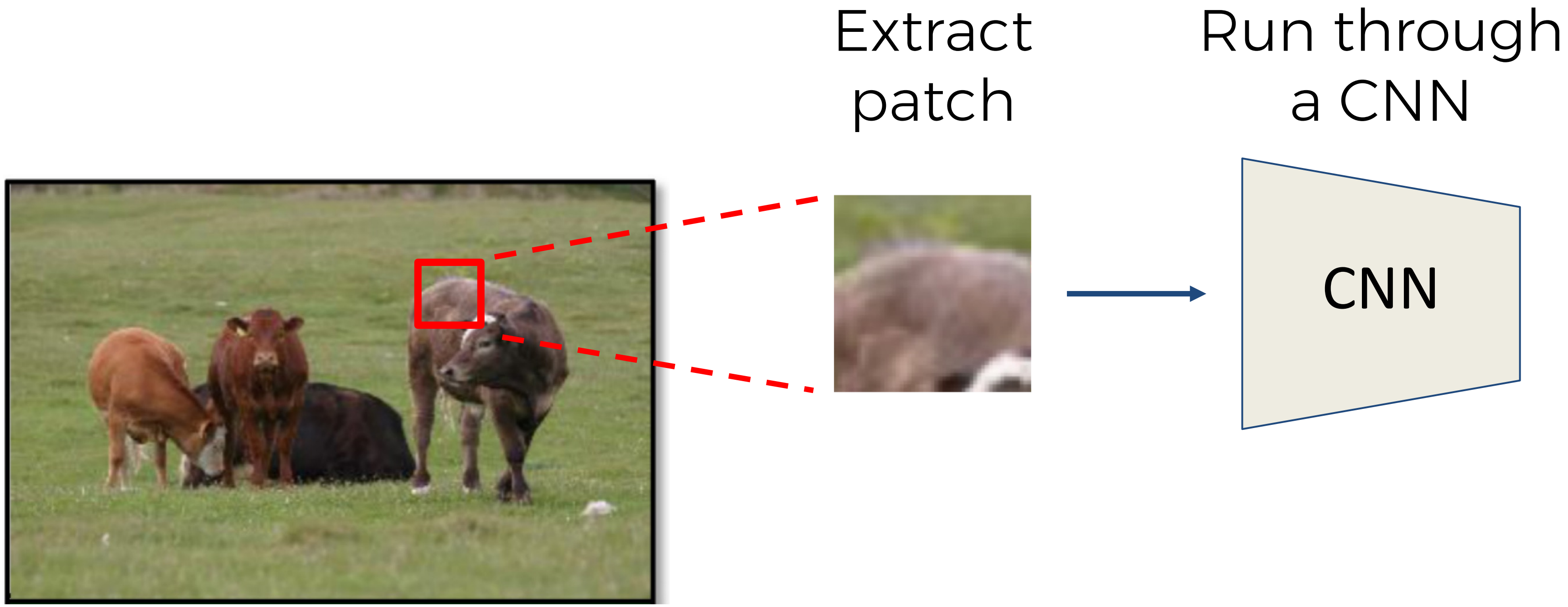
Deep Semantic Segmentation

Extract
patch

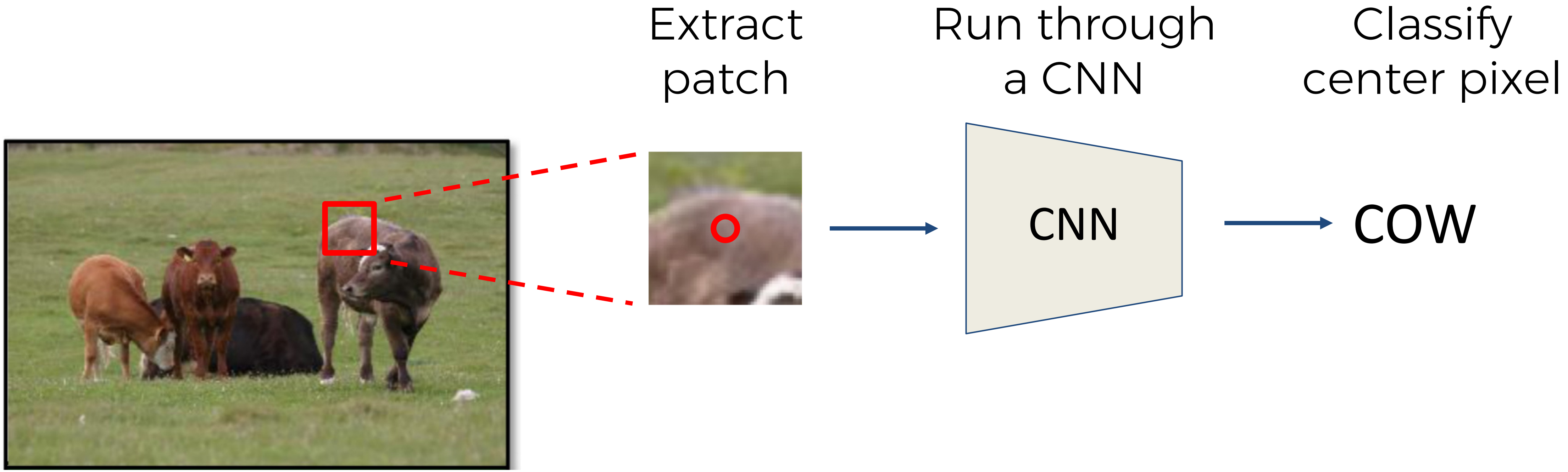




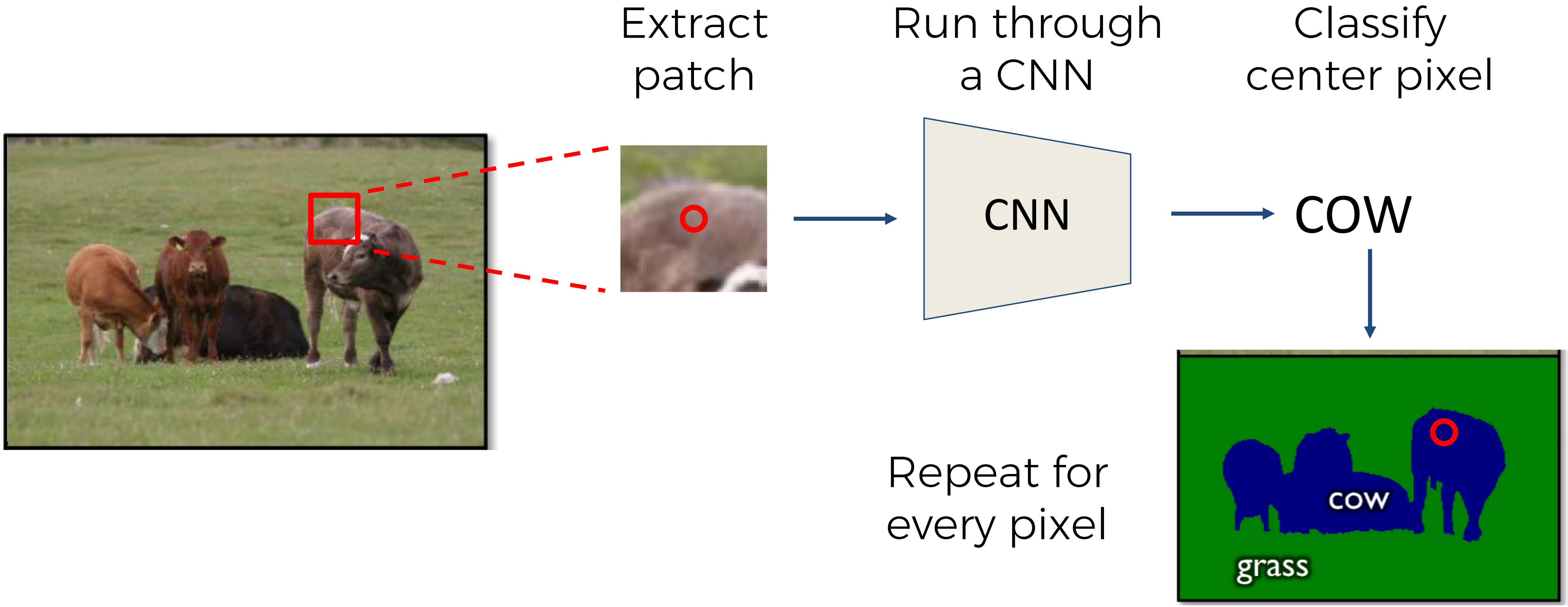
Deep Semantic Segmentation



Deep Semantic Segmentation



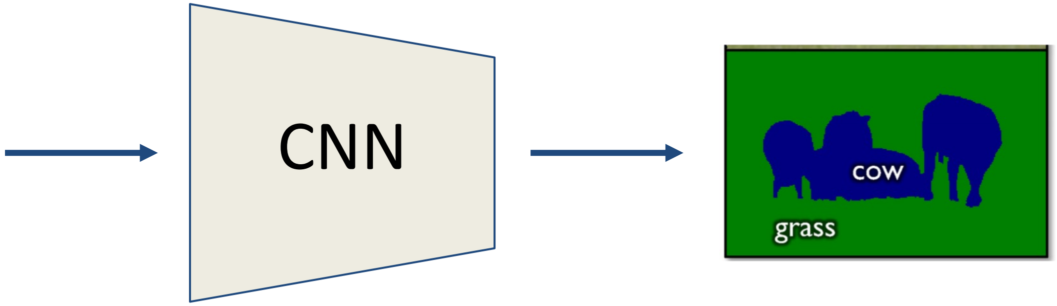
Deep Semantic Segmentation



F.-F. Li, A. Karpathy and J. Johnson

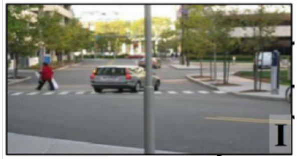
Deep Semantic Segmentation

Run "fully convolutional" network to get all pixels at once



Smaller output due to pooling

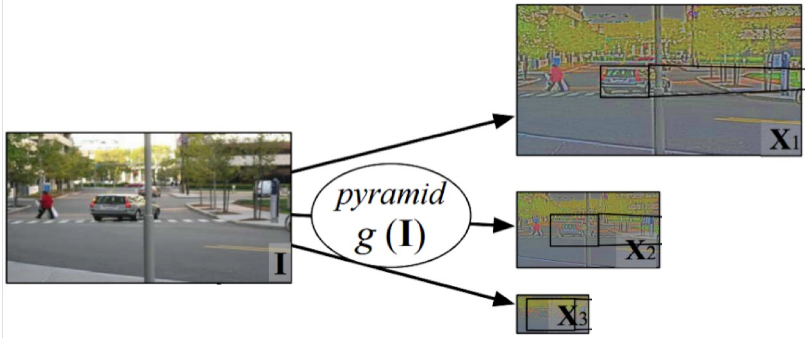
Semantic Segmentation: Multi-Scale



Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013

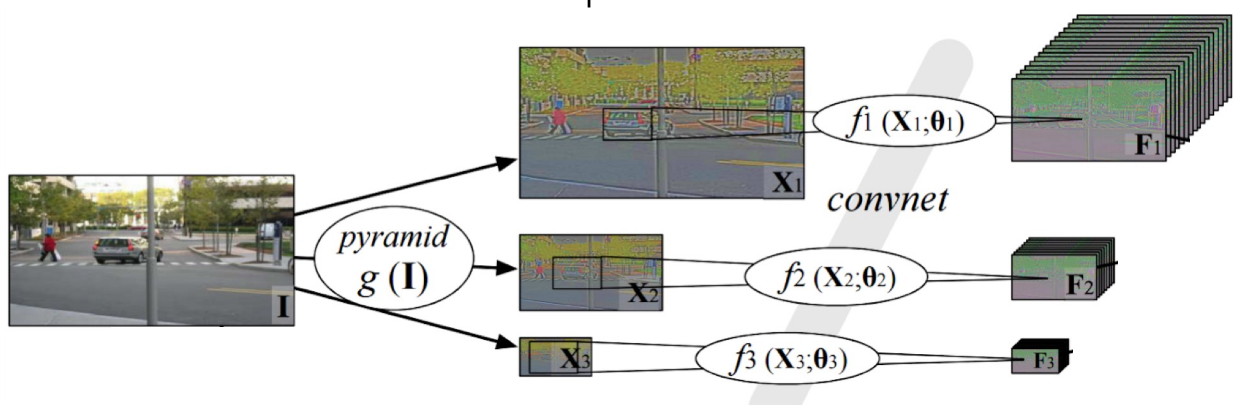
Semantic Segmentation: Multi-Scale

Resize image to
multiple scales



Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013

Semantic Segmentation: Multi-Scale



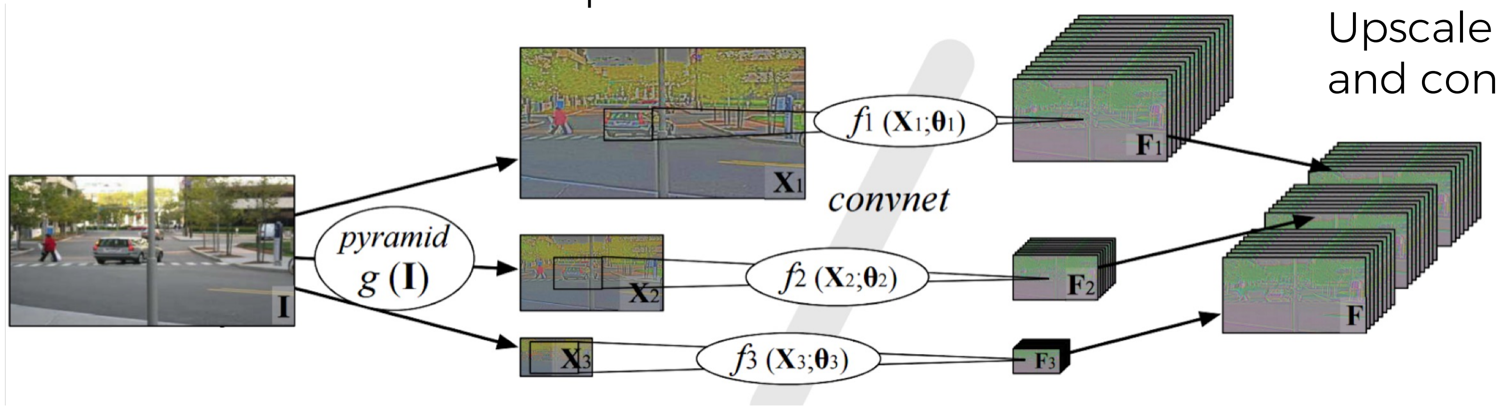
Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013

Semantic Segmentation: Multi-Scale

Resize image to multiple scales

Run one CNN per scale

Upscale outputs and concatenate



Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013

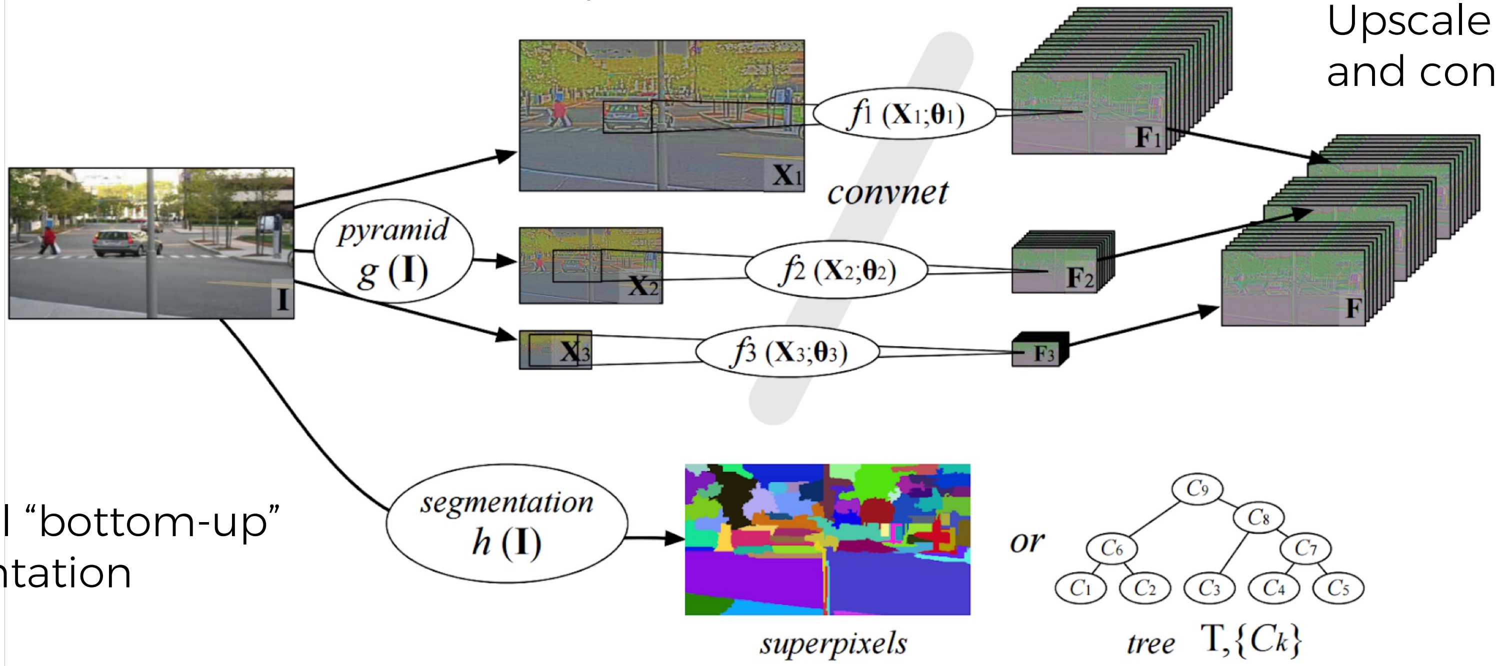
Semantic Segmentation: Multi-Scale

Resize image to multiple scales

Run one CNN per scale

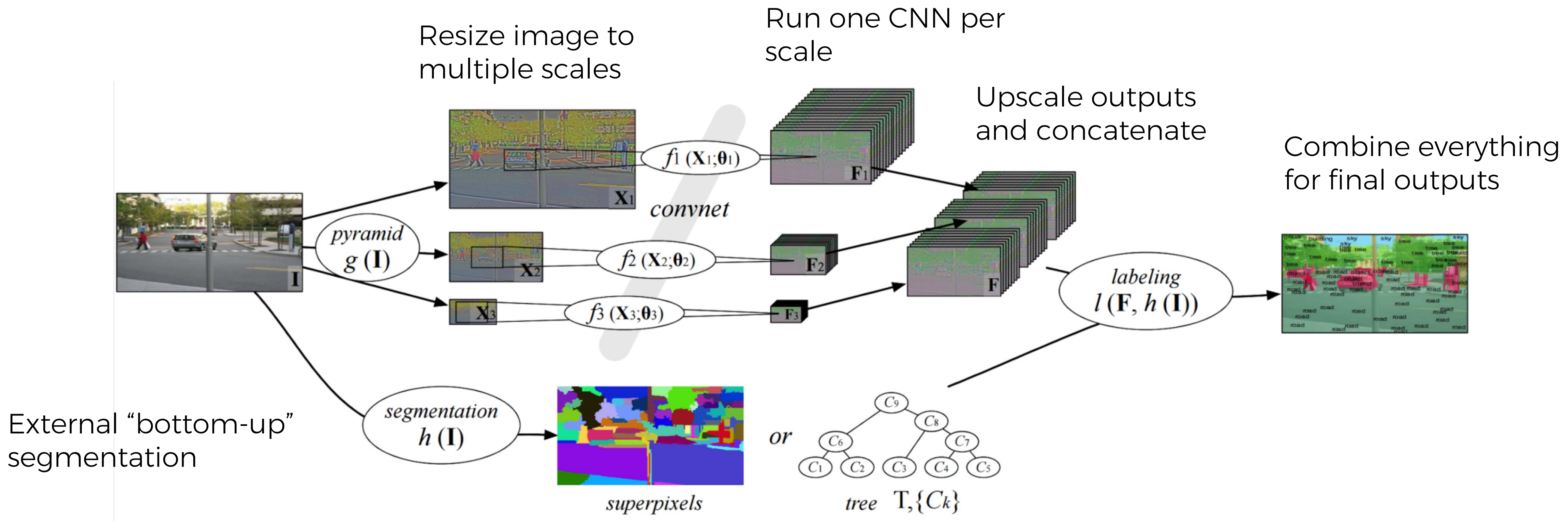
Upscale outputs and concatenate

External "bottom-up" segmentation



Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013

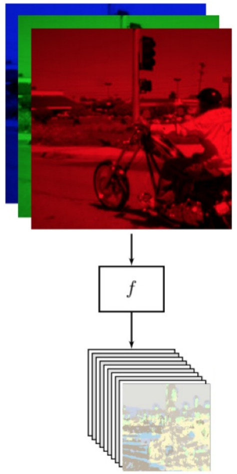
Semantic Segmentation: Multi-Scale



Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013

Semantic Segmentation: Refinement

Apply CNN once to get labels

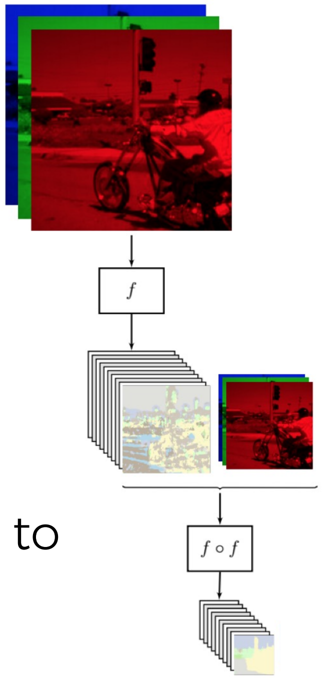


Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

Semantic Segmentation: Refinement

Apply CNN once to get labels

Apply AGAIN to refine labels



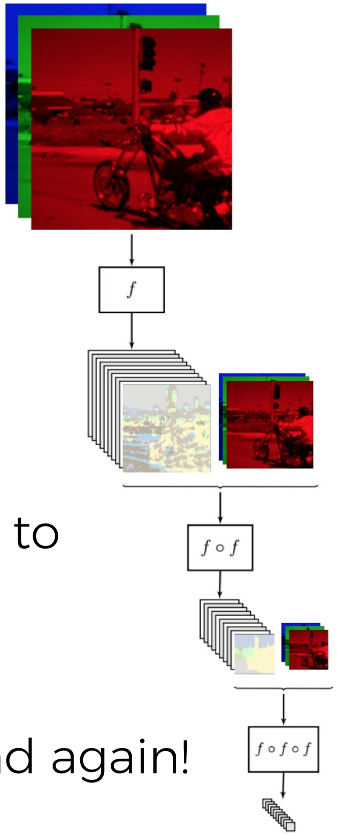
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

Semantic Segmentation: Refinement

Apply CNN once to get labels

Apply AGAIN to refine labels

And again!



Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

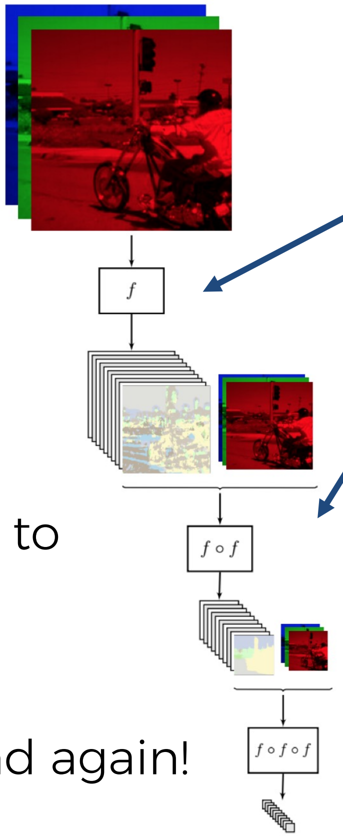
Semantic Segmentation: Refinement

Same CNN weights:
recurrent convolutional network

Apply CNN once to get labels

Apply AGAIN to refine labels

And again!



Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

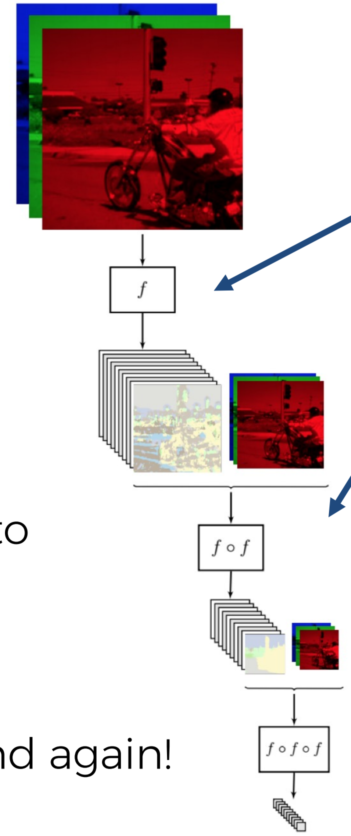
Semantic Segmentation: Refinement

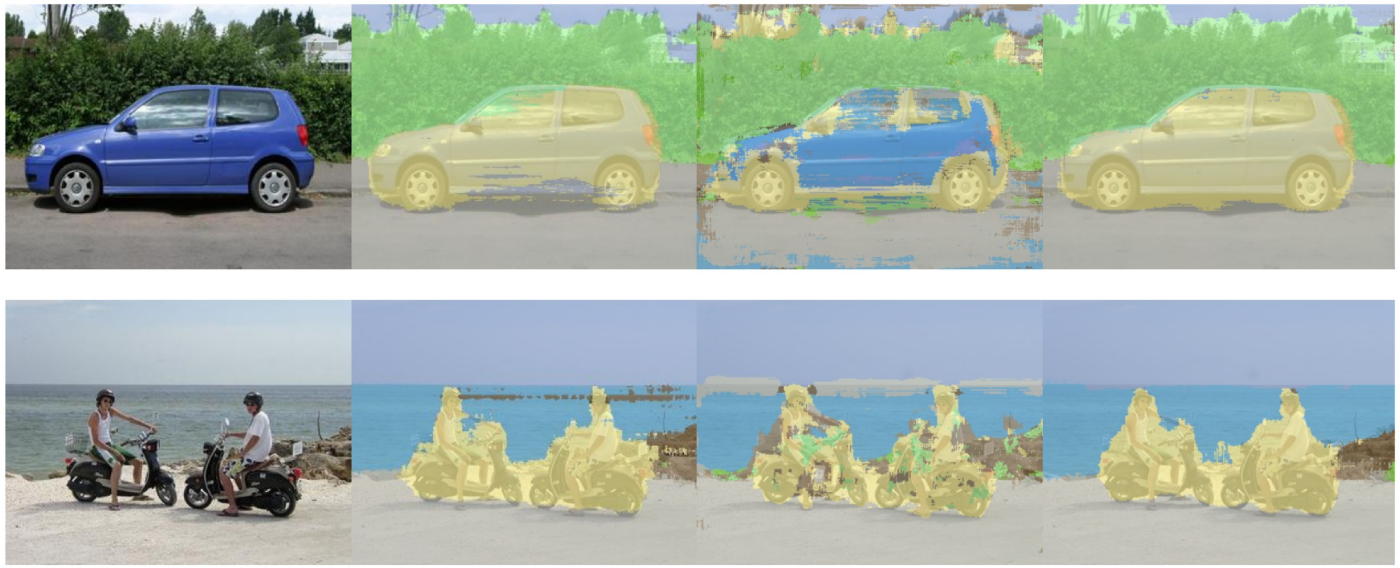
Same CNN weights:
recurrent convolutional network

Apply CNN once
to get labels

Apply AGAIN to
refine labels

And again!

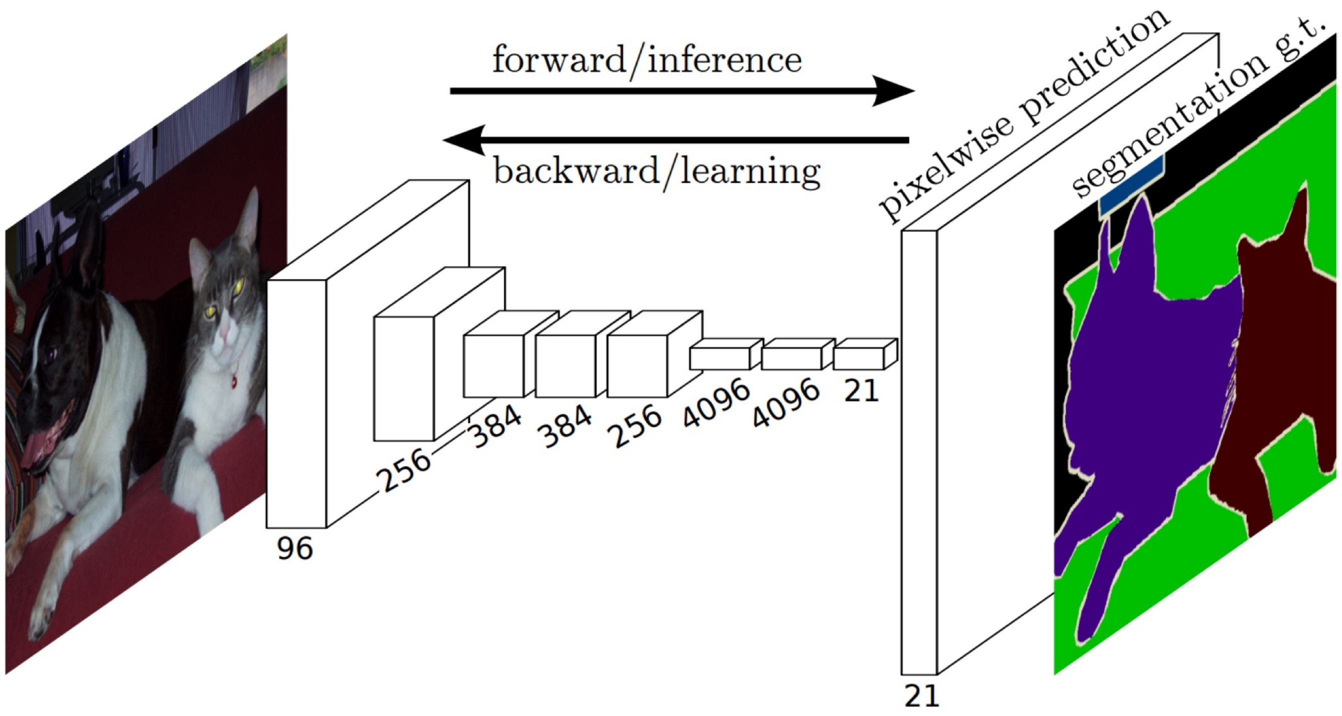




More iterations improve results

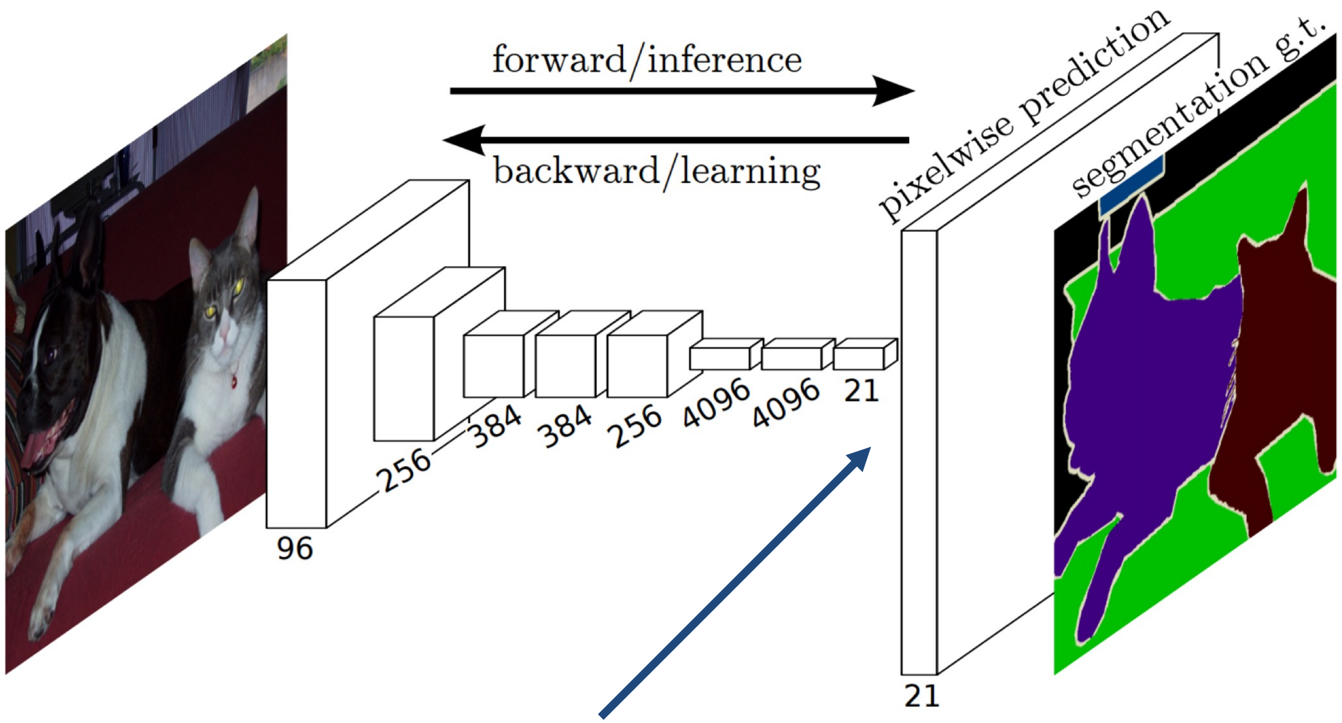
Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014

Semantic Segmentation: Upsampling



Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

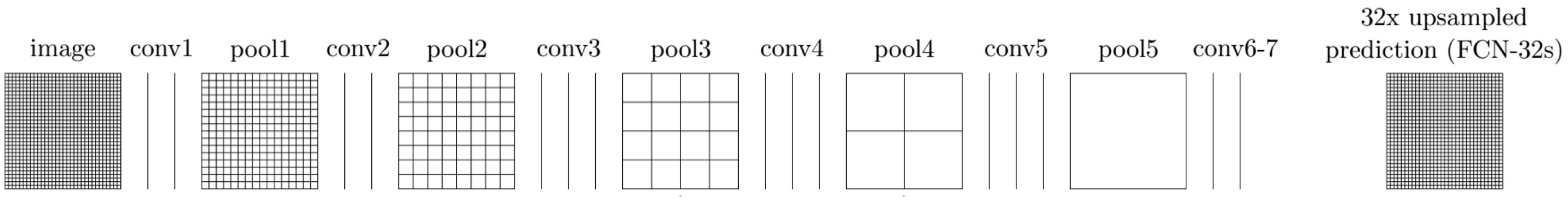
Semantic Segmentation: Upsampling



Learnable upsampling!

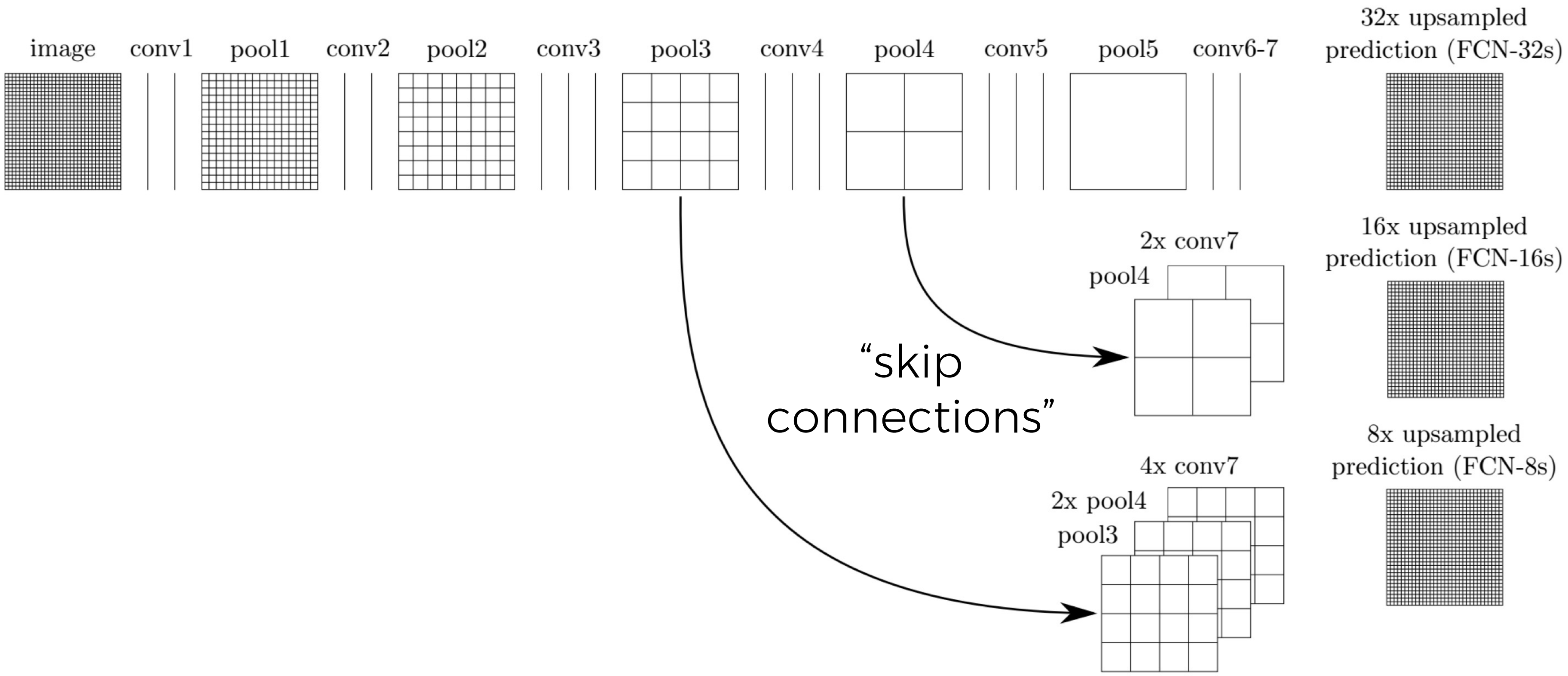
Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

Semantic Segmentation: Upsampling



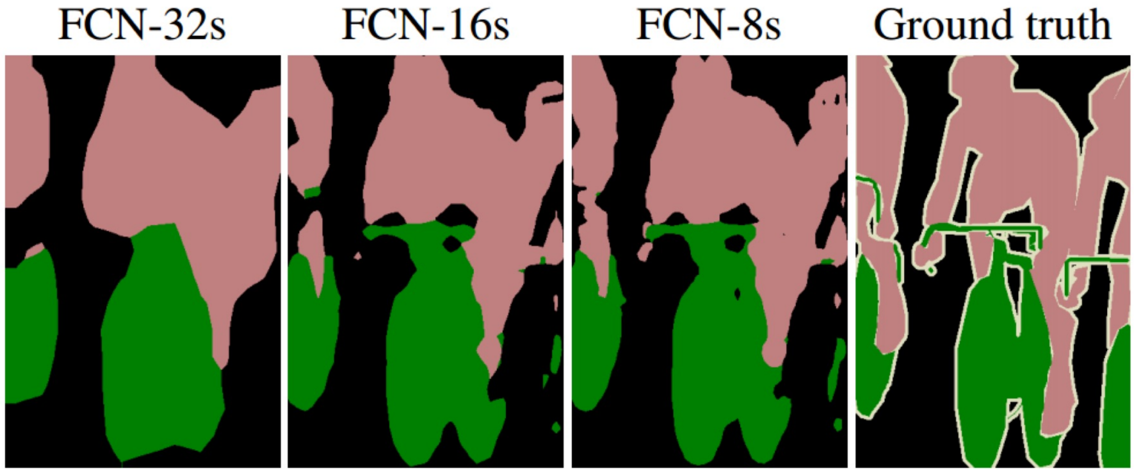
Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

Semantic Segmentation: Upsampling

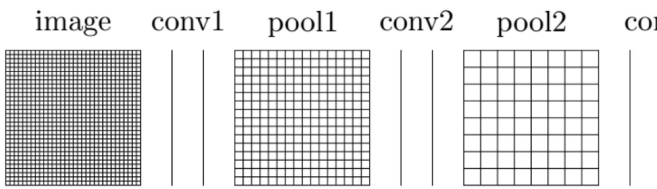
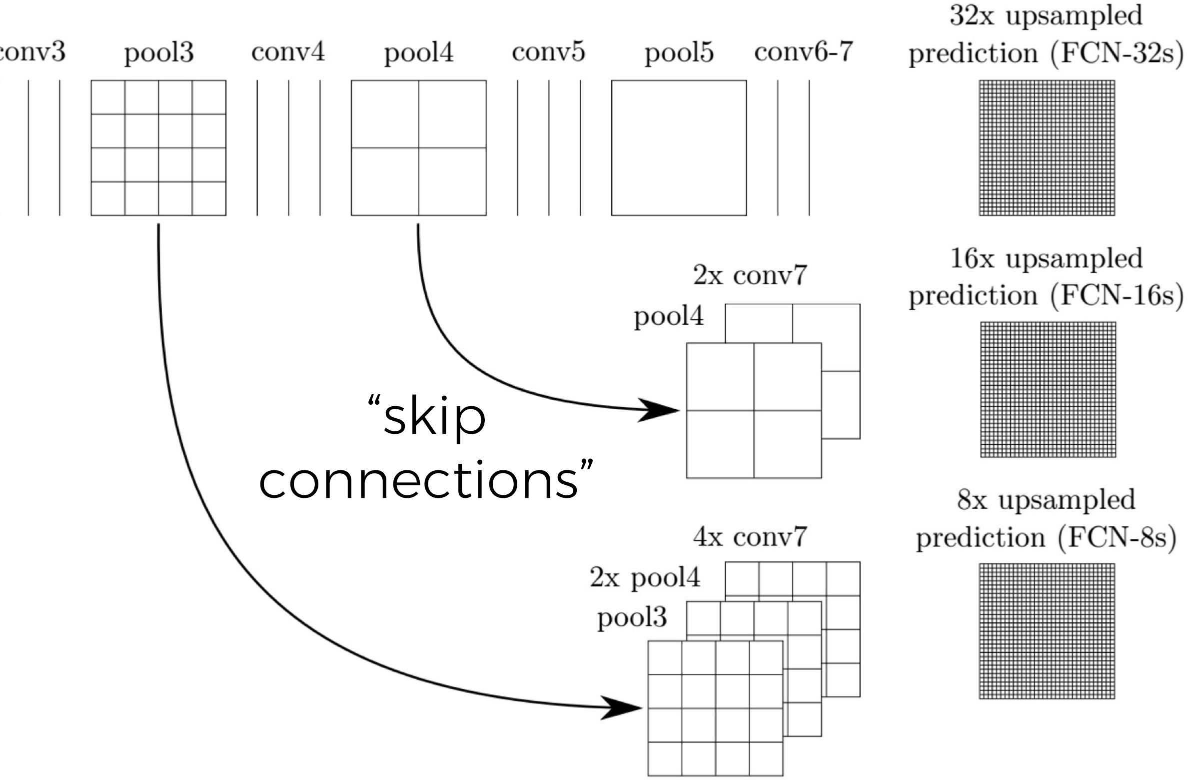


Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015

Semantic Segmentation: Upsampling



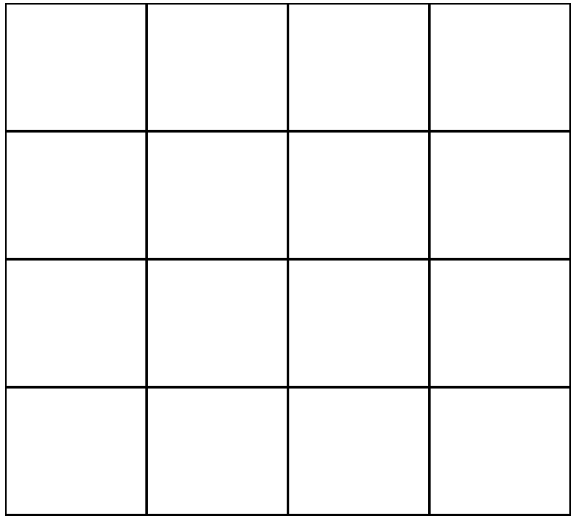
Skip connections = Better results

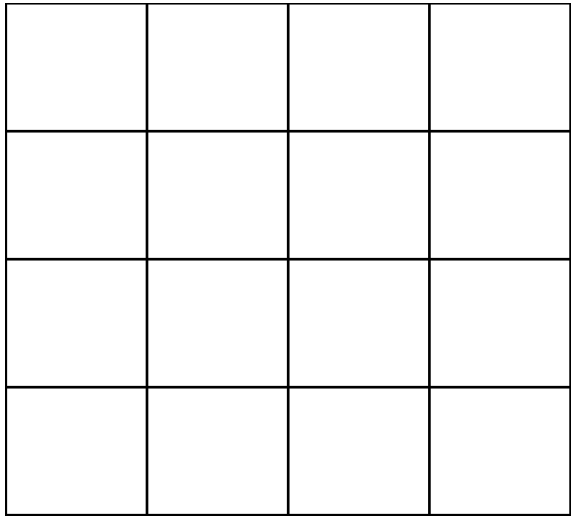
Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

Learnable Upsampling: “Deconvolution”

Typical 3 x 3 convolution, stride 1 pad 1



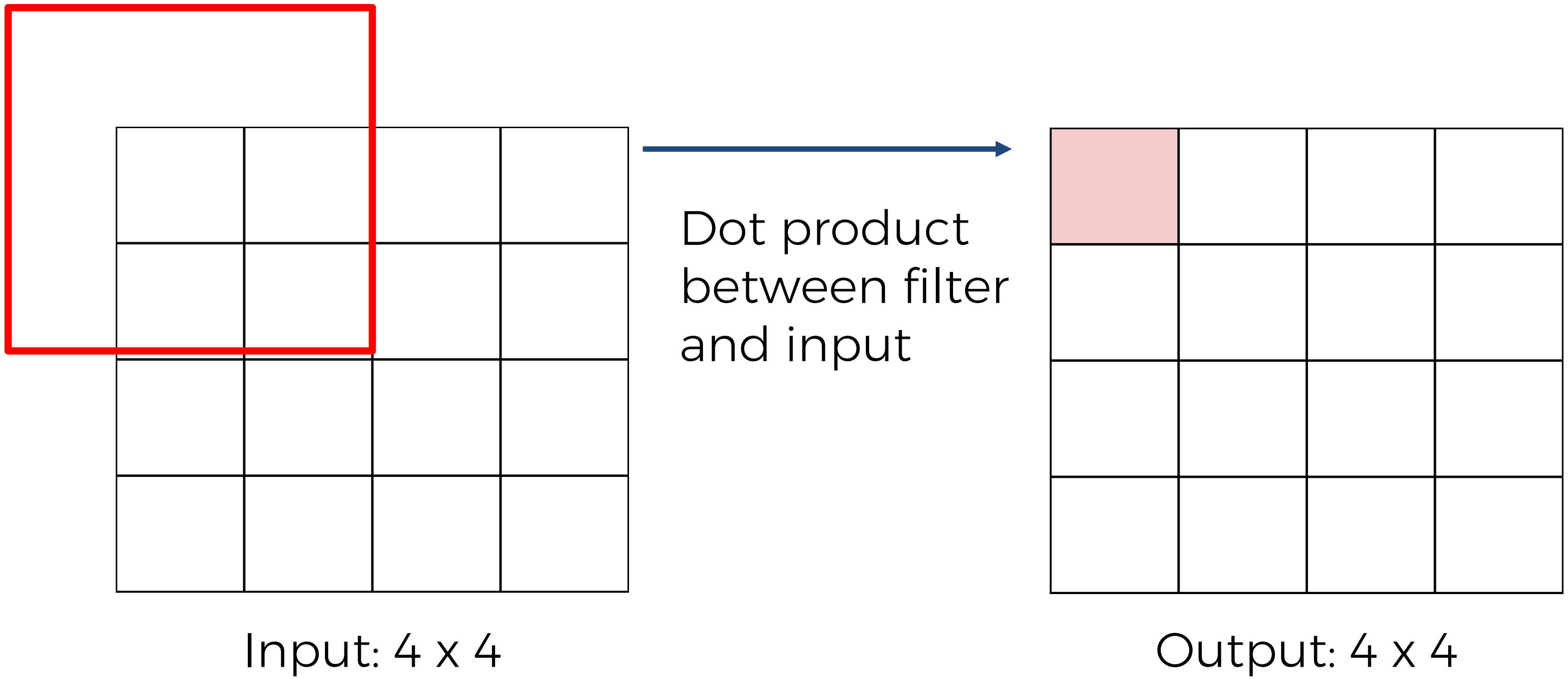
Input: 4 x 4



Output: 4 x 4

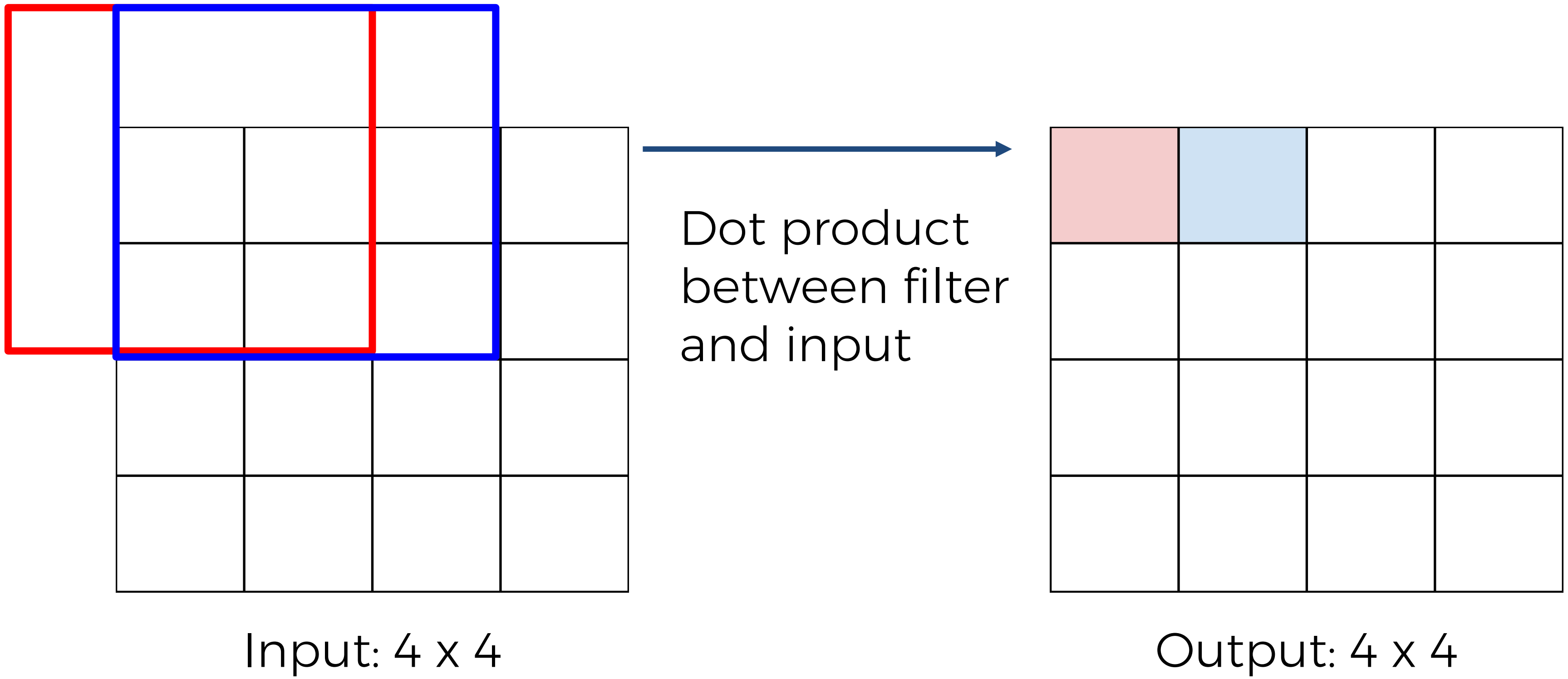
Learnable Upsampling: "Deconvolution"

Typical 3 x 3 convolution, stride 1 pad 1



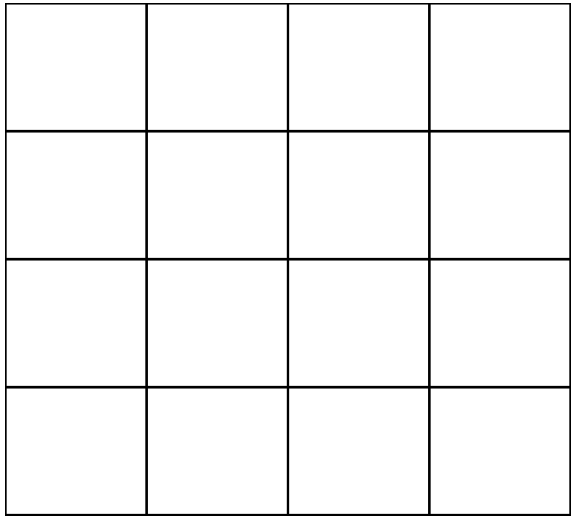
Learnable Upsampling: “Deconvolution”

Typical 3 x 3 convolution, stride 1 pad 1

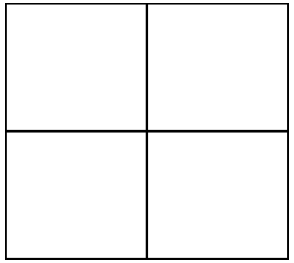


Learnable Upsampling: "Deconvolution"

Typical 3 x 3 convolution, **stride 2** pad 1



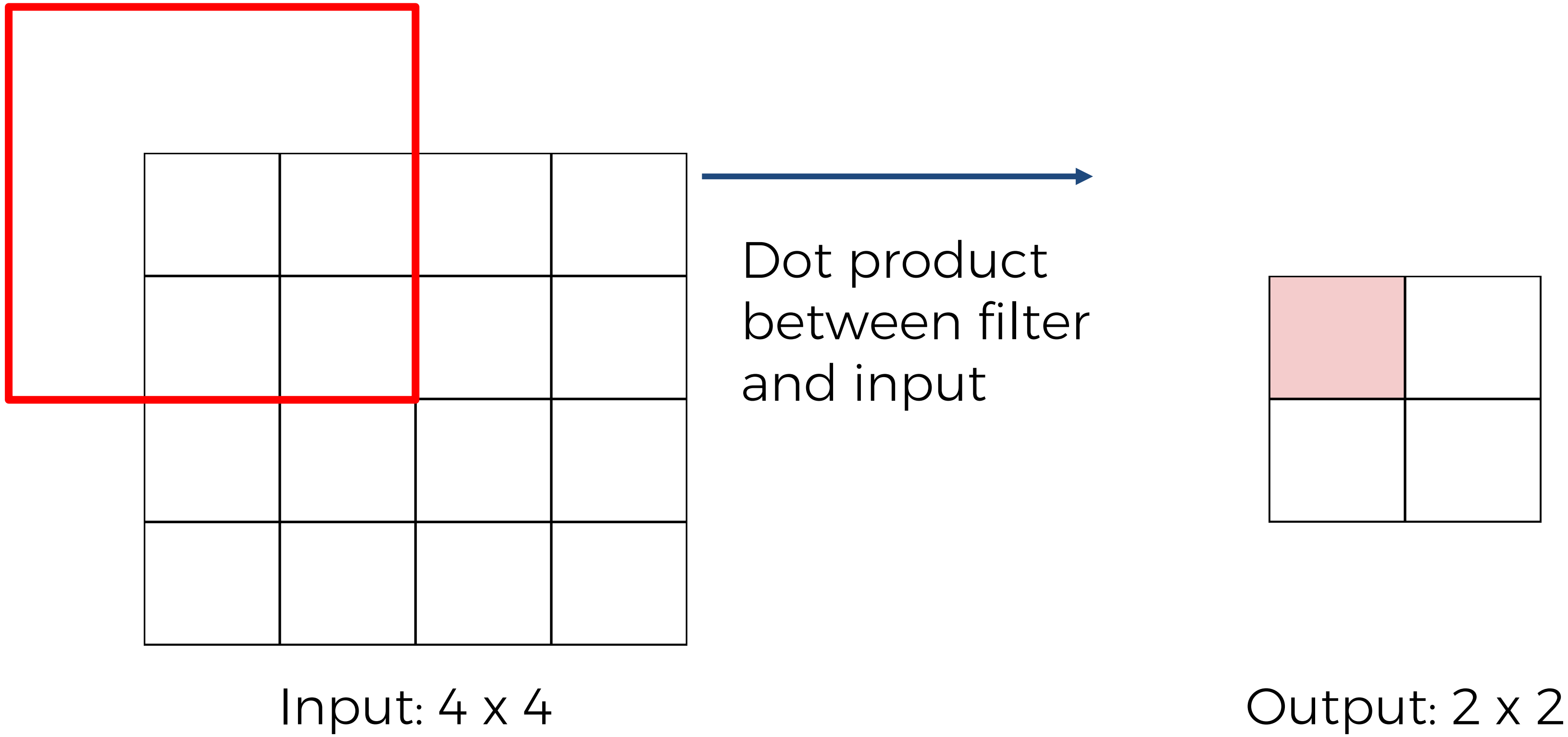
Input: 4 x 4



Output: 2 x 2

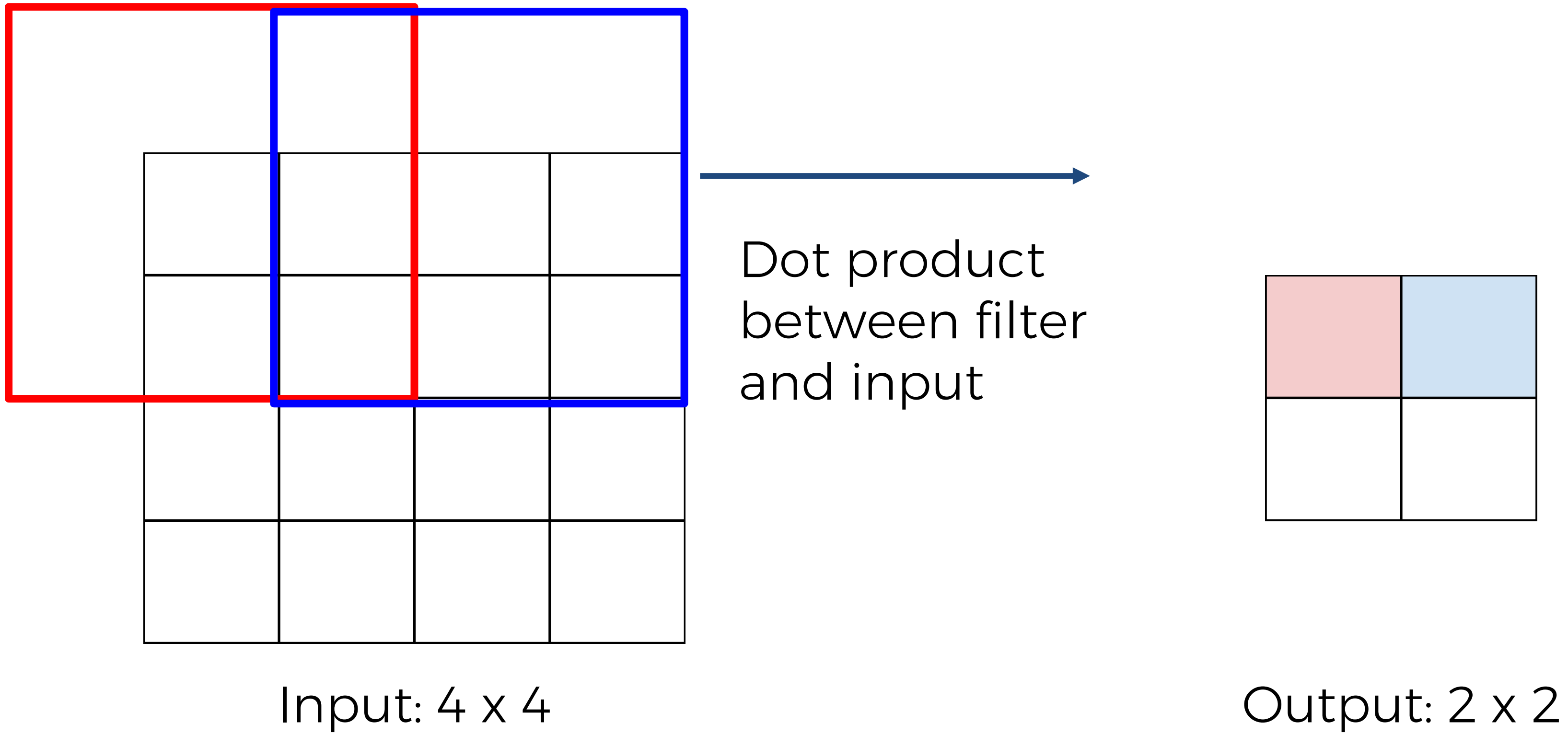
Learnable Upsampling: "Deconvolution"

Typical 3 x 3 convolution, **stride 2** pad 1



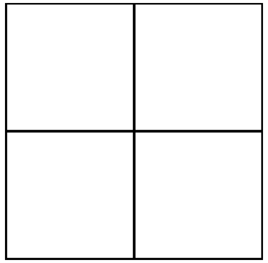
Learnable Upsampling: "Deconvolution"

Typical 3 x 3 convolution, **stride 2** pad 1

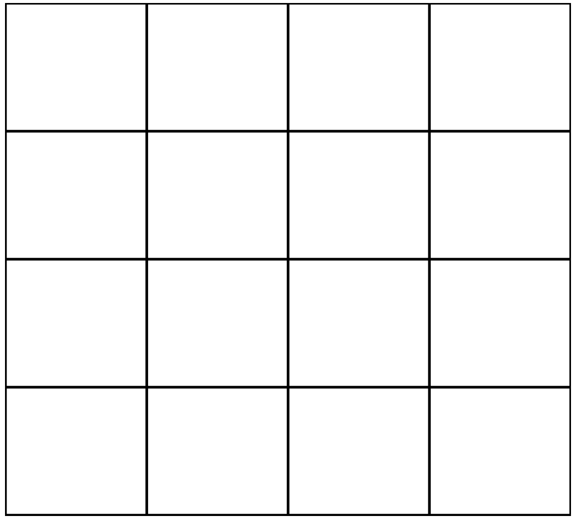


Learnable Upsampling: "Deconvolution"

3 x 3 deconvolution, stride 2 pad 1



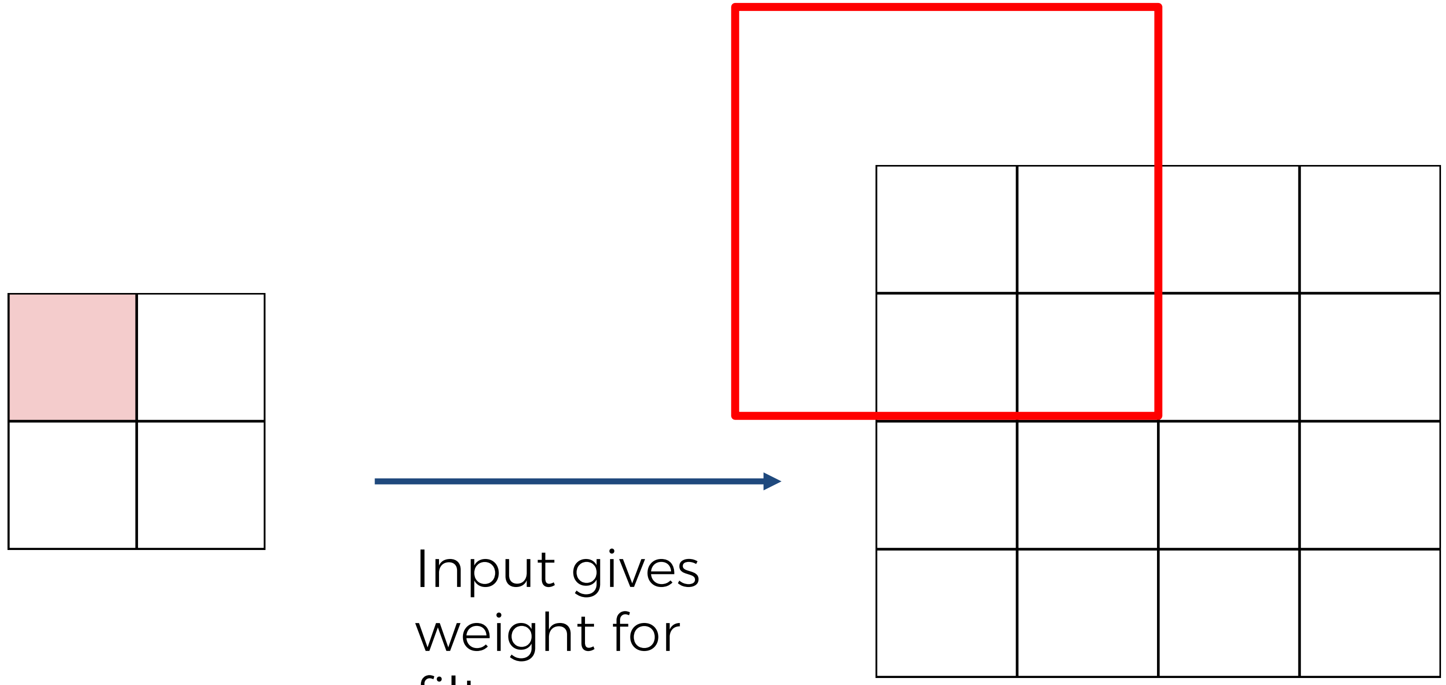
Input: 2 x 2



Output: 4 x 4

Learnable Upsampling: “Deconvolution”

3 x 3 deconvolution, stride 2 pad 1

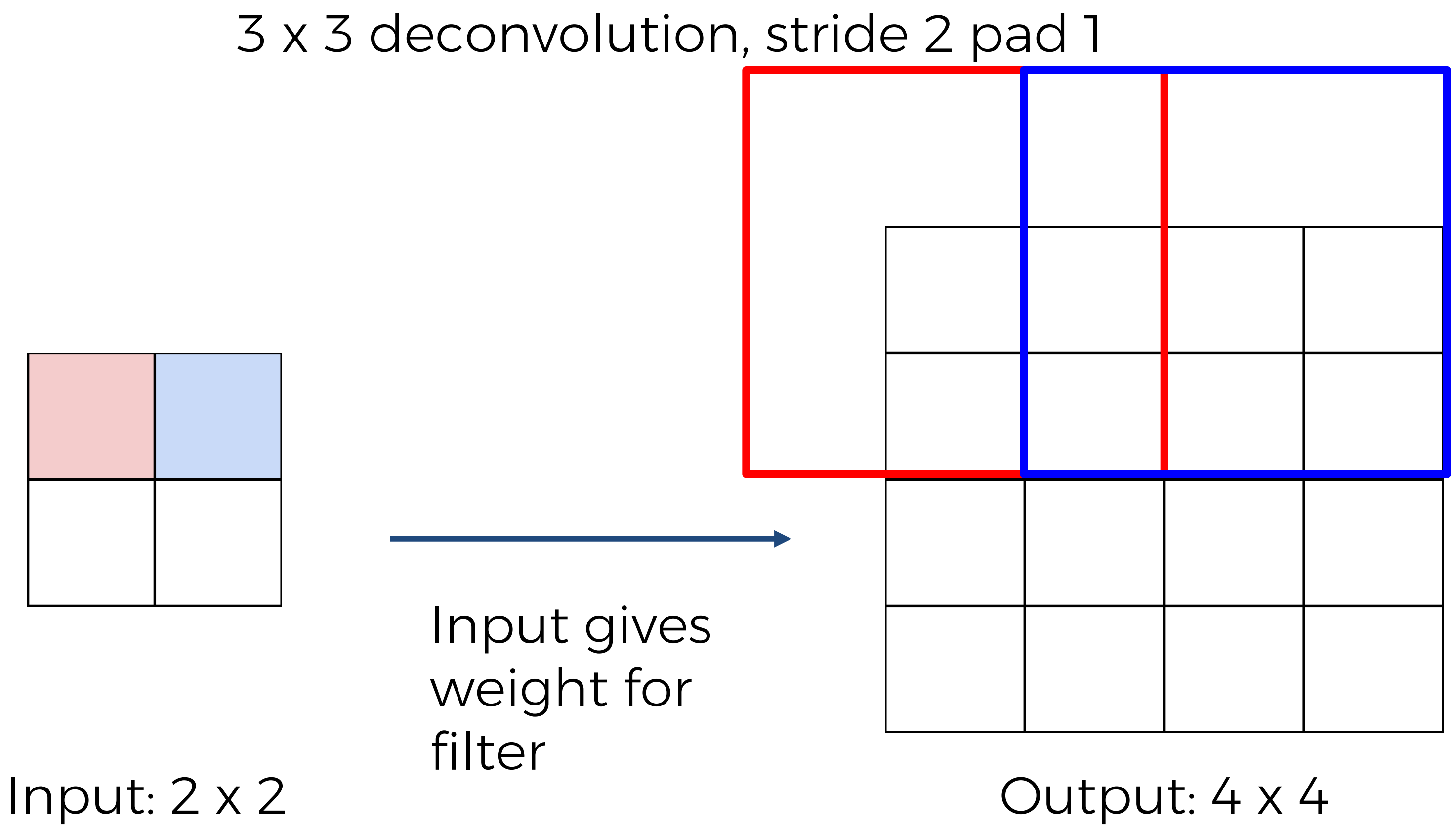


Input: 2 x 2

Input gives weight for filter

Output: 4 x 4

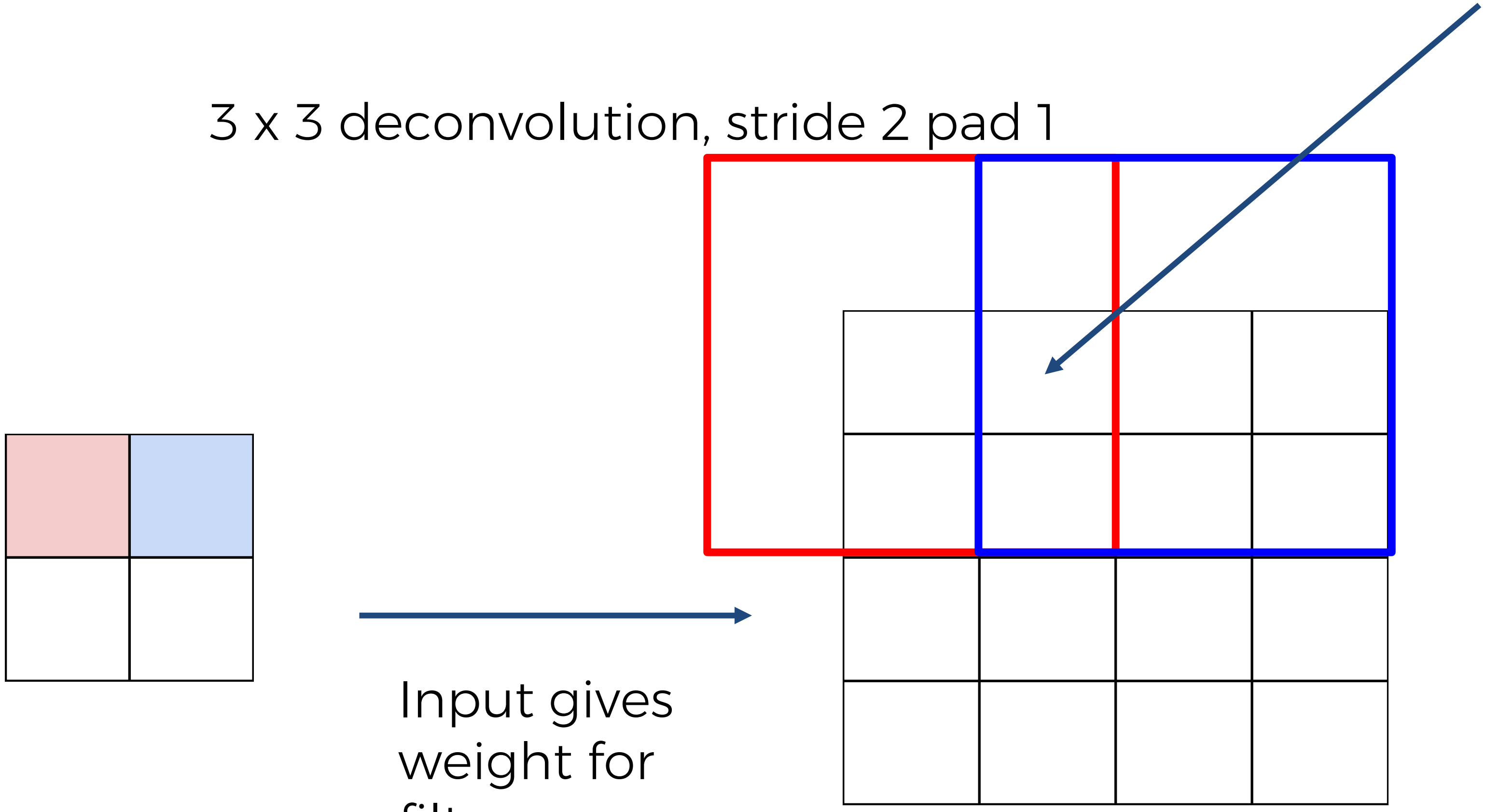
Learnable Upsampling: “Deconvolution”



Learnable Upsampling: “Deconvolution”

Sum where output overlaps

3 x 3 deconvolution, stride 2 pad 1



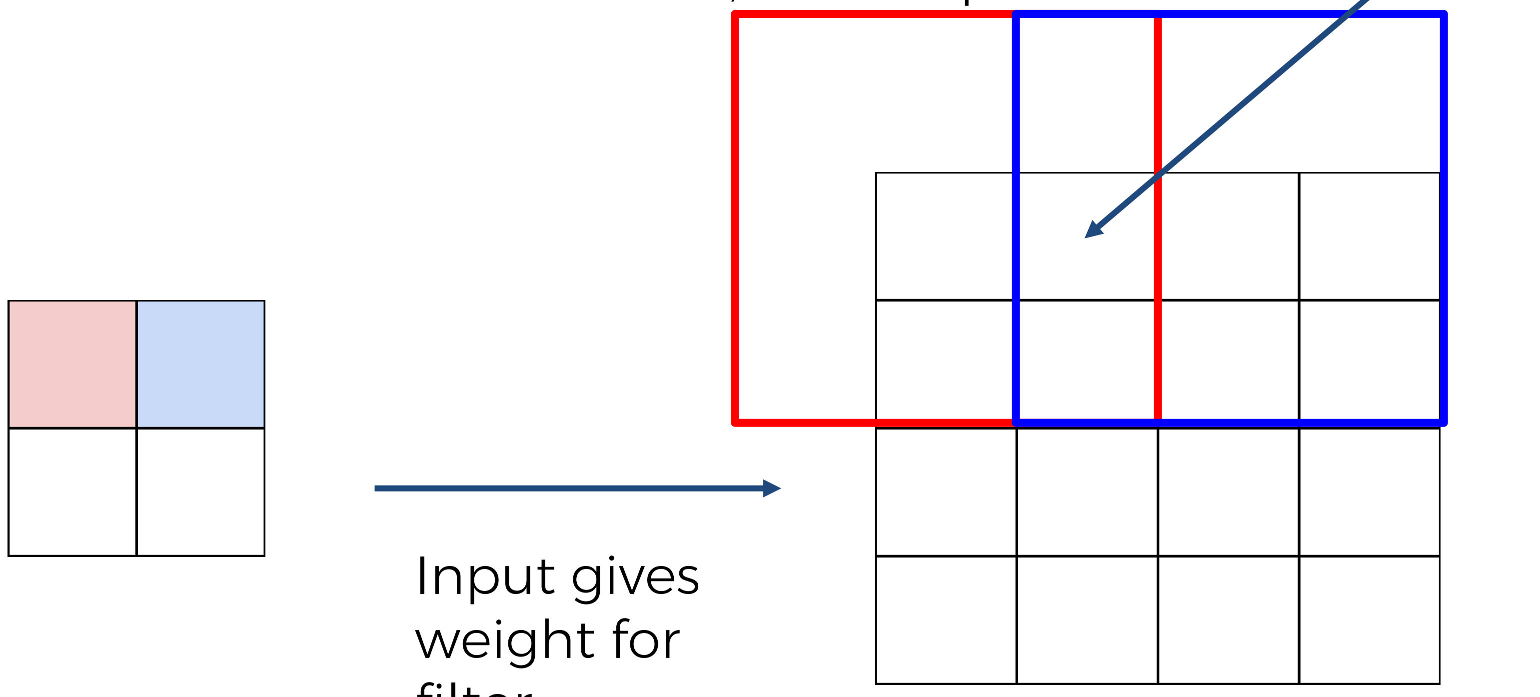
Input: 2 x 2

Input gives weight for filter

Output: 4 x 4

Learnable Upsampling: "Deconvolution"

3 x 3 deconvolution, stride 2 pad 1



Input: 2 x 2

Input gives weight for filter

Output: 4 x 4

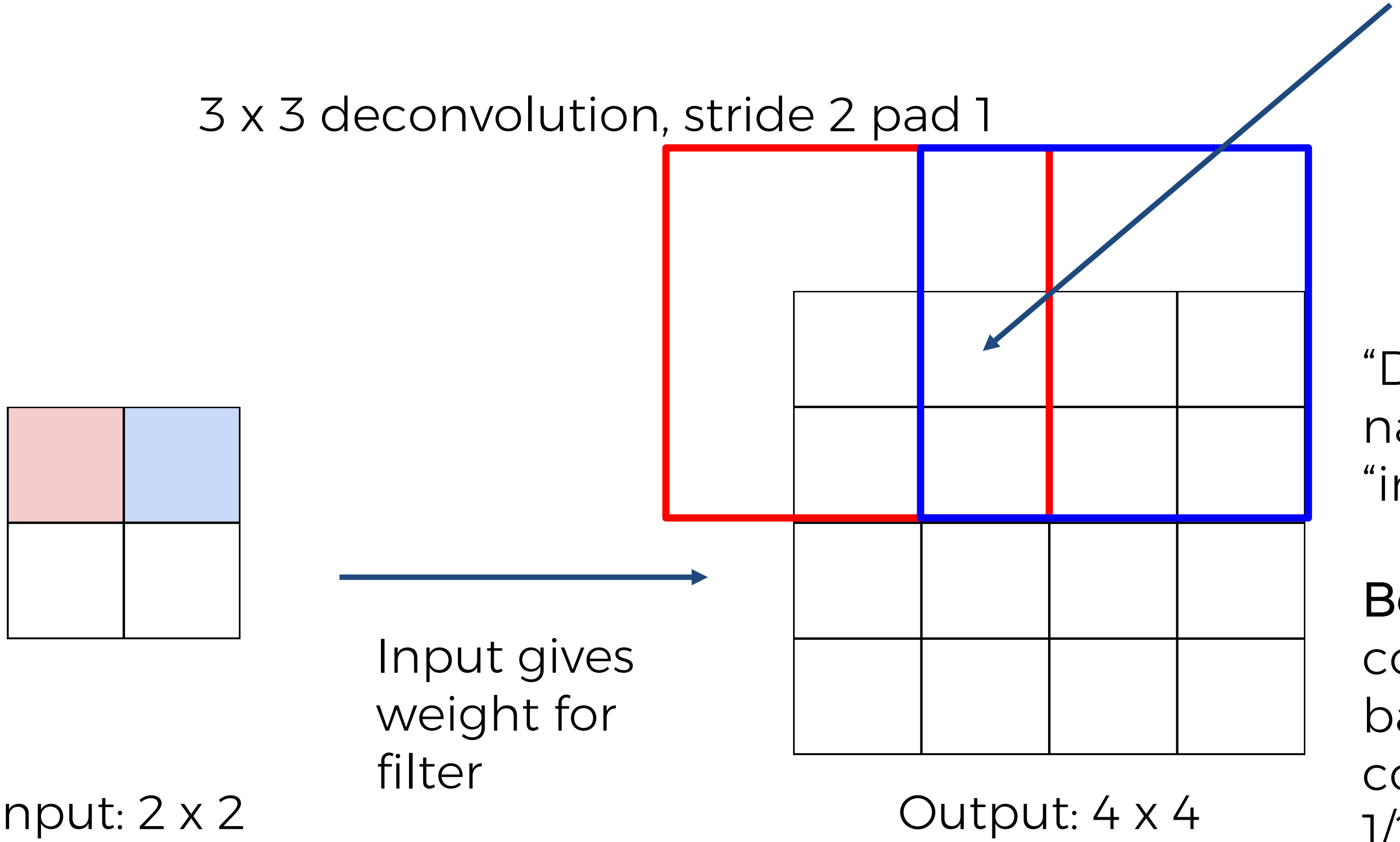
Sum where output overlaps

Same as backward pass for normal convolution!

Learnable Upsampling: “Deconvolution”

Sum where
output overlaps

Same as backward
pass for normal
convolution!



“Deconvolution” is a bad name, already defined as “inverse of convolution”

Better names:
convolution transpose,
backward strided
convolution,
1/2 strided convolution,
upconvolution

Learnable Upsampling: “Deconvolution”

[1]It is more proper to say “convolutional transpose operation” rather than “deconvolutional” operation. Hence, we will be using the term “convolutional transpose” from now.

Im et al, “Generating images with recurrent adversarial networks”, arXiv 2016

A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions

Radford et al, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, ICLR 2016

“Deconvolution” is a bad name, already defined as “inverse of convolution”

Better names:
convolution transpose,
backward strided convolution,
1/2 strided convolution,
upconvolution

Learnable Upsampling: "Deconvolution"

Great explanation in appendix

[1]It is more proper to say "convolutional transpose operation" rather than "deconvolutional" operation. Hence, we will be using the term "convolutional transpose" from now.

Im et al, "Generating images with recurrent adversarial networks", arXiv 2016

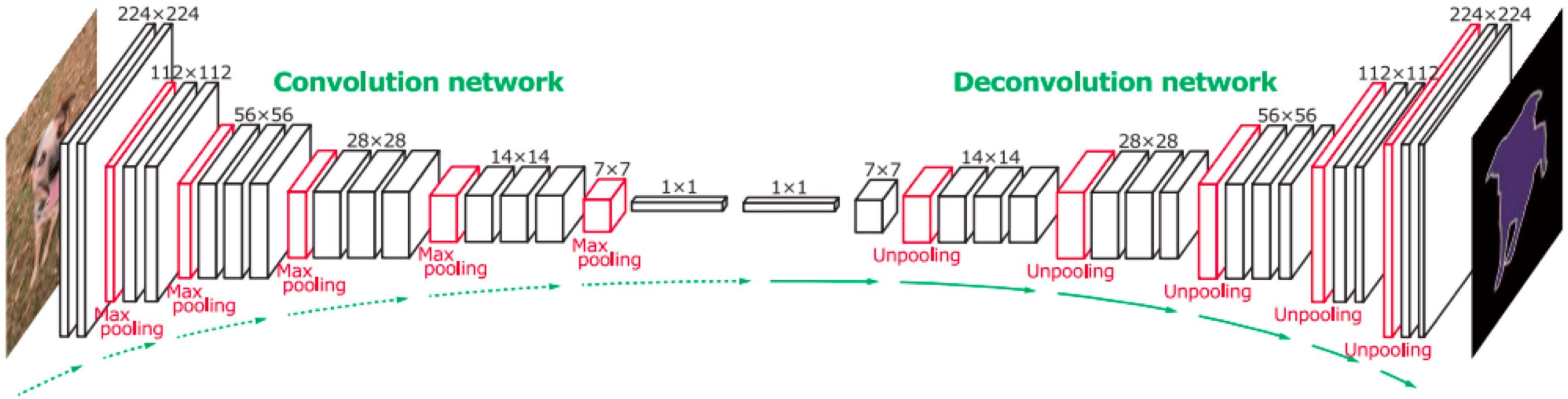
A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions

Radford et al, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", ICLR 2016

"Deconvolution" is a bad name, already defined as "inverse of convolution"

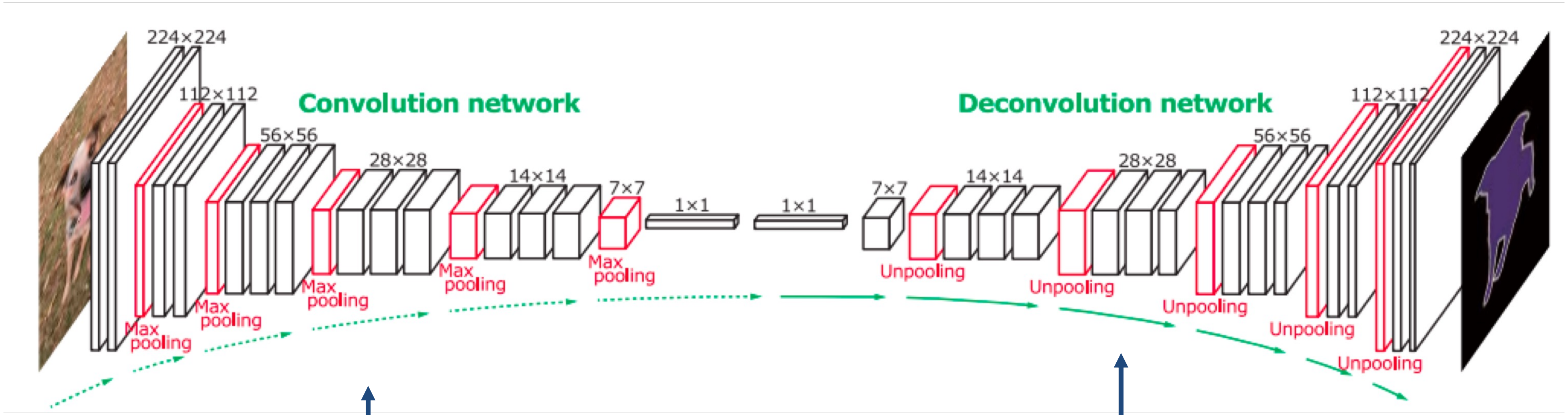
Better names:
convolution transpose,
backward strided convolution,
1/2 strided convolution,
upconvolution

Semantic Segmentation: Upsampling



Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Semantic Segmentation: Upsampling



Normal VGG

"Upside down" VGG

6 days of training on Titan X…

Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Instance Segmentation

Detect instances, give category, label pixels

“simultaneous detection and segmentation” (SDS)

Lots of recent work (MS-COCO)

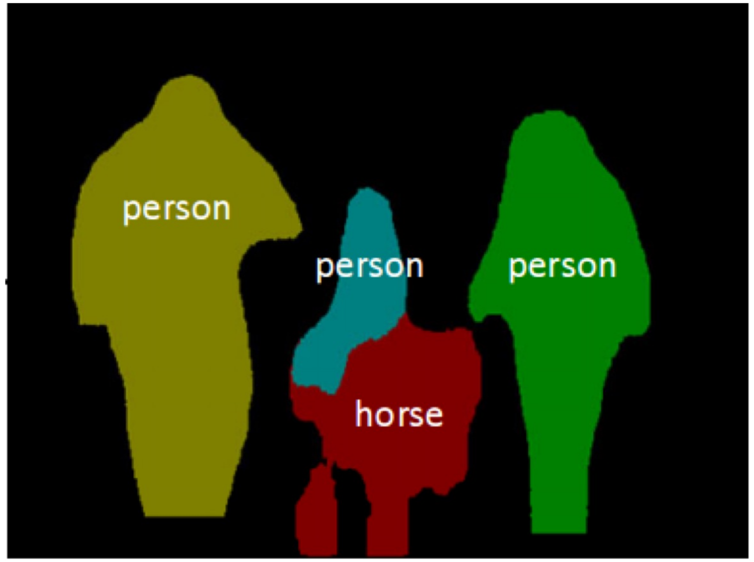
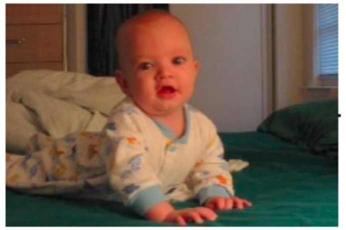


Figure credit: Dai et al, “Instance-aware Semantic Segmentation via Multi-task Network Cascades”, arXiv 2015

Instance Segmentation

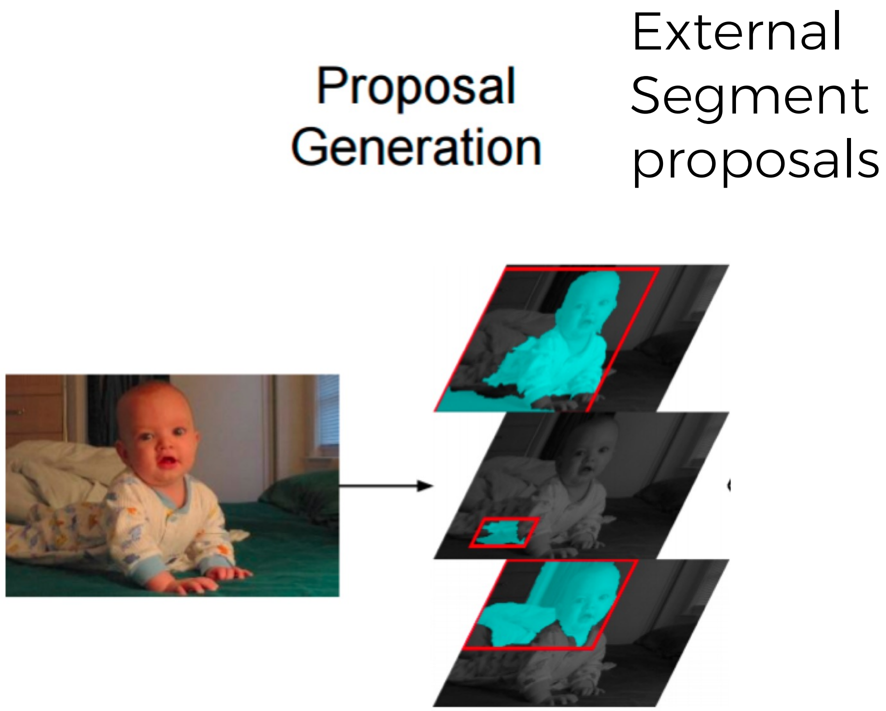
Similar to R-CNN, but with segments



Hariharan et al, "Simultaneous Detection and Segmentation", ECCV 2014

Instance Segmentation

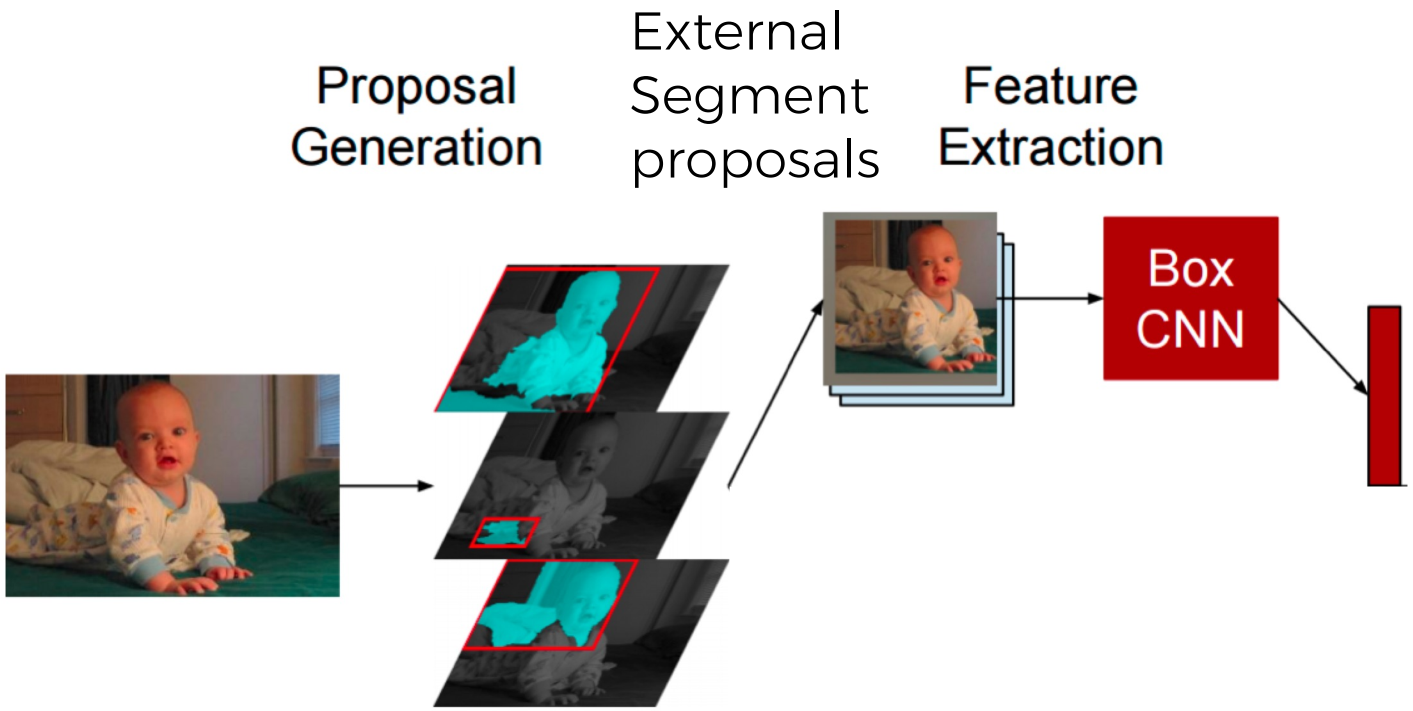
Similar to R-CNN, but with segments



Hariharan et al, “Simultaneous Detection and Segmentation”, ECCV 2014

Instance Segmentation

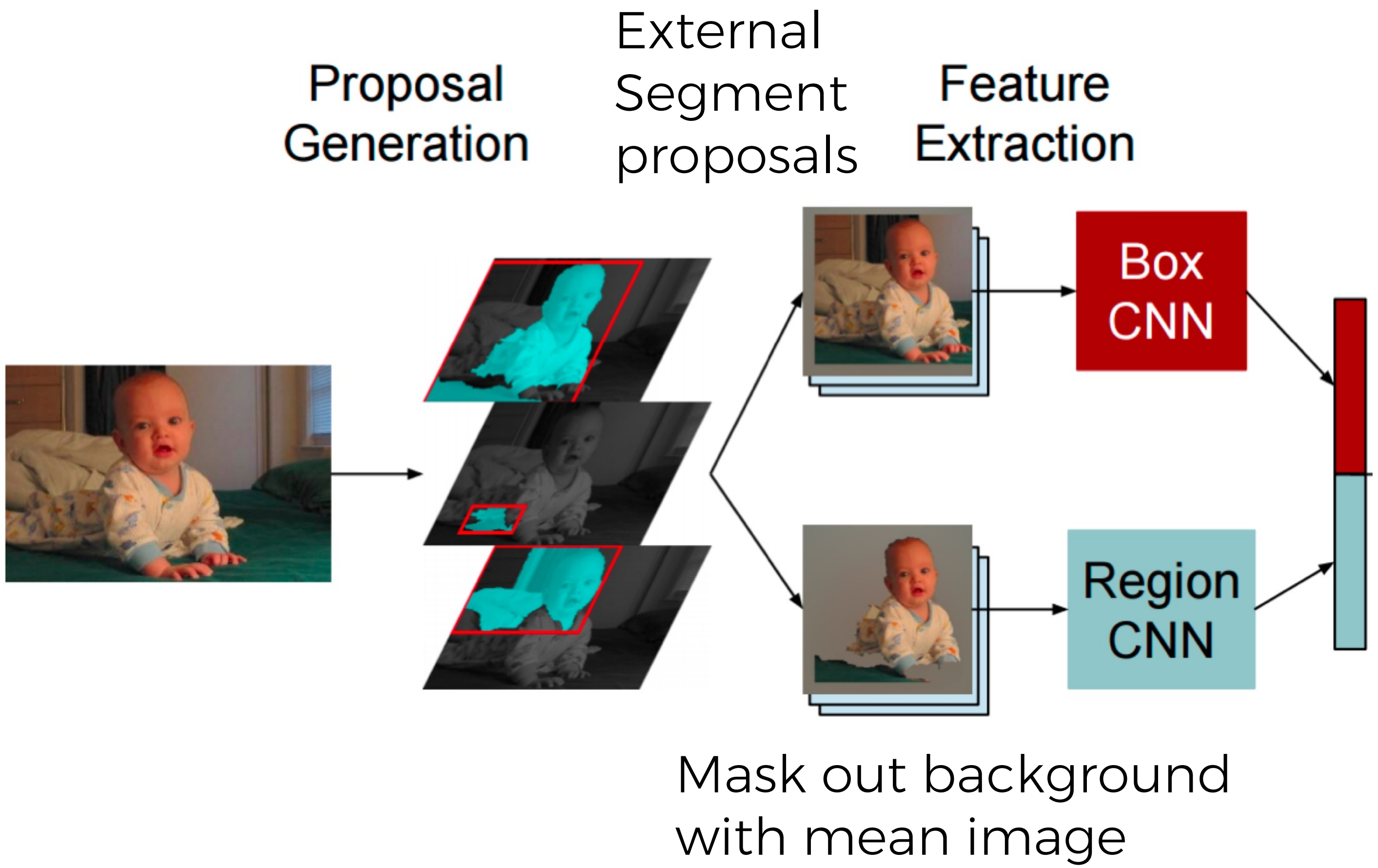
Similar to R-CNN, but with segments



Hariharan et al, "Simultaneous Detection and Segmentation", ECCV 2014

Instance Segmentation

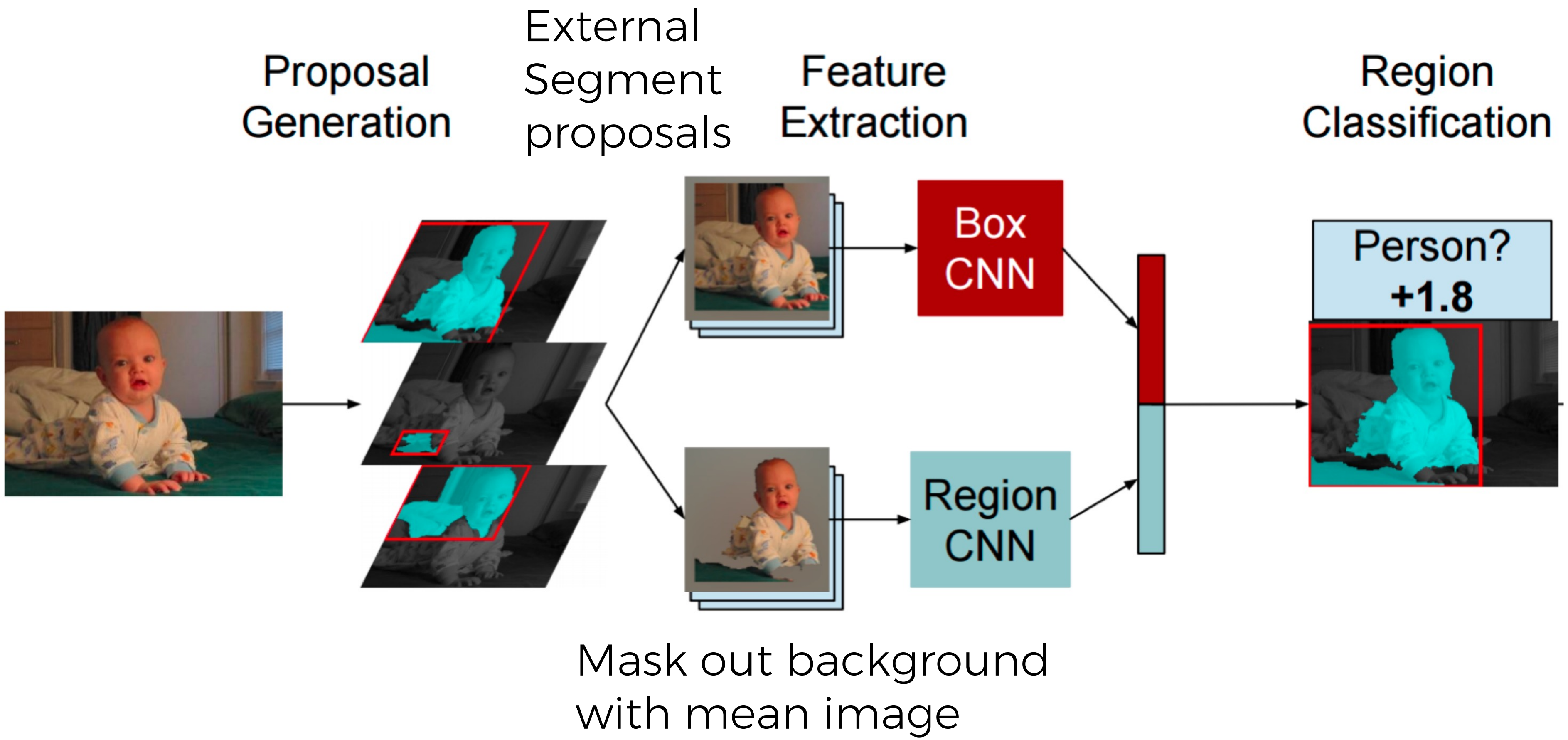
Similar to R-CNN, but with segments



Hariharan et al, “Simultaneous Detection and Segmentation”, ECCV 2014

Instance Segmentation

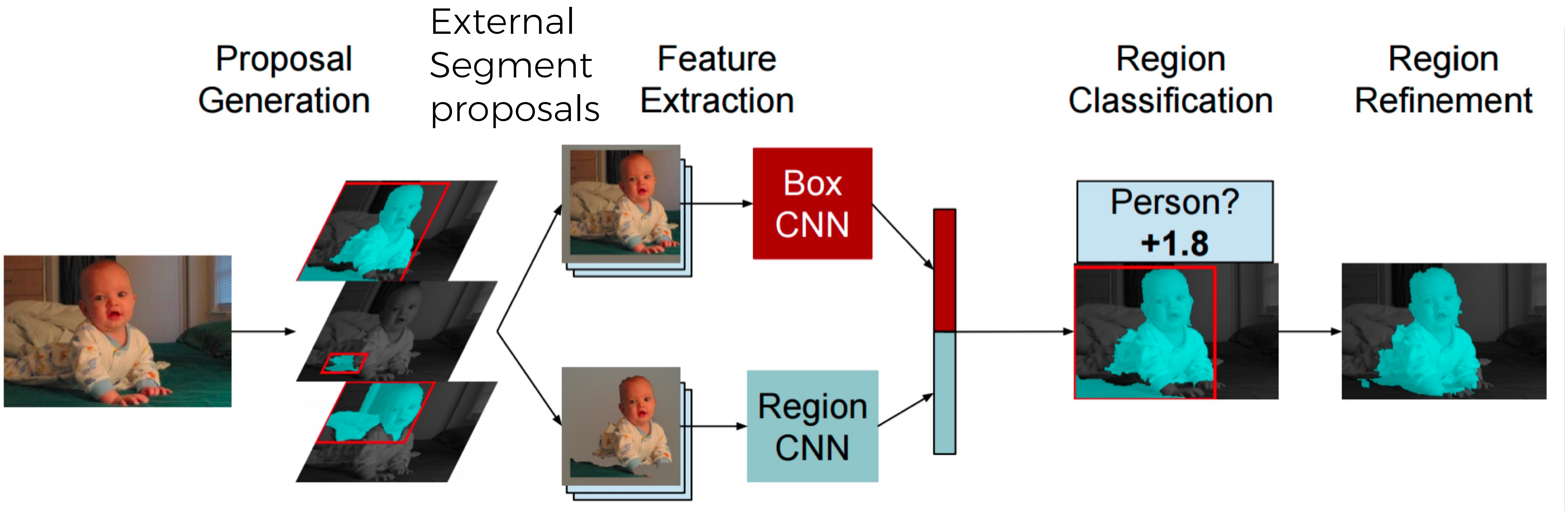
Similar to R-CNN, but with segments



Hariharan et al, “Simultaneous Detection and Segmentation”, ECCV 2014

Instance Segmentation

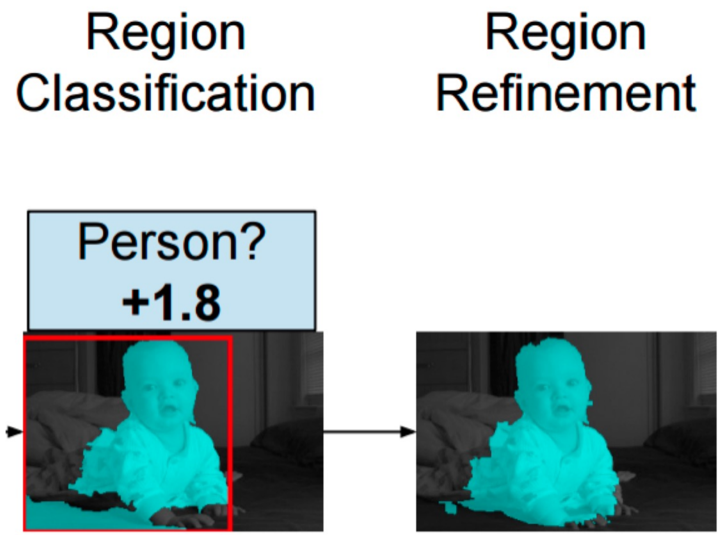
Similar to R-CNN, but with segments



Mask out background with mean image

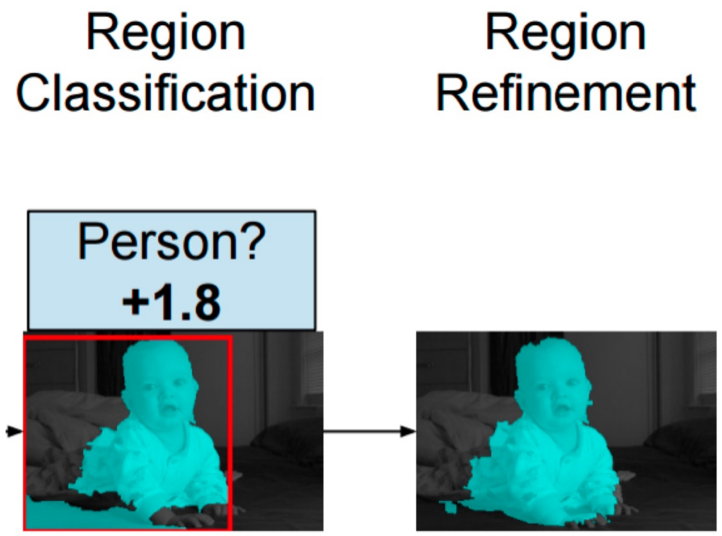
Hariharan et al, "Simultaneous Detection and Segmentation", ECCV 2014

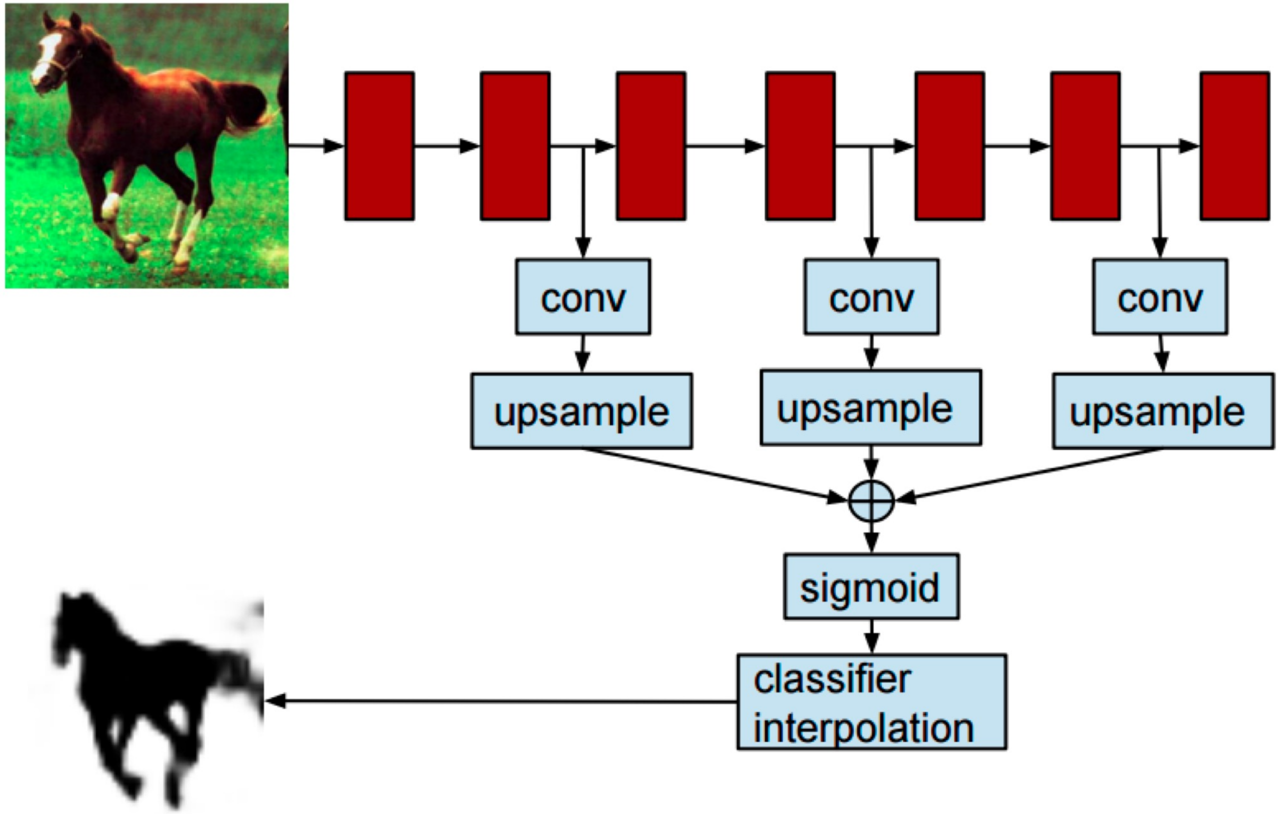
Instance Segmentation: Hypercolumns



Hariharan et al, "Hypercolumns for Object Segmentation and Fine-grained Localization", CVPR 2015

Instance Segmentation: Hypercolumns





Hariharan et al, "Hypercolumns for Object Segmentation and Fine-grained Localization", CVPR 2015

Instance Segmentation: Cascades

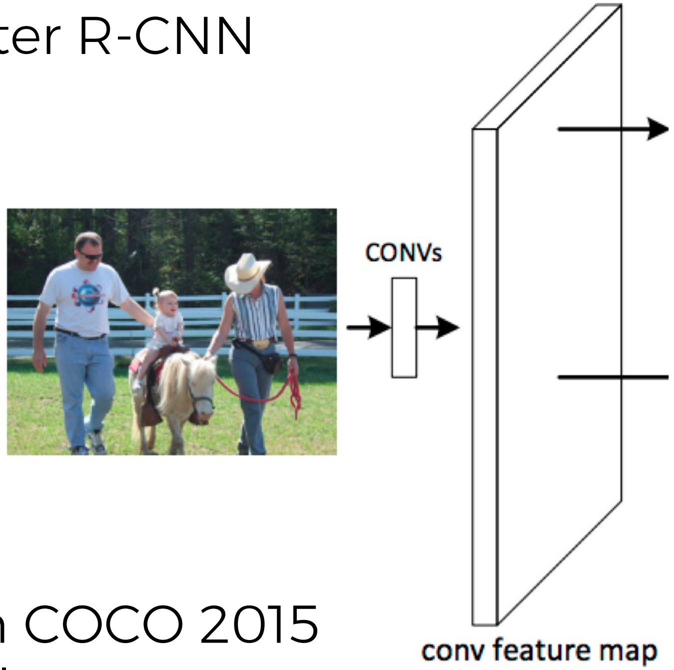
Similar to
Faster R-CNN

Won COCO 2015
challenge
(with ResNet)

Dai et al, "Instance-aware Semantic Segmentation via Multi-task Network Cascades", arXiv 2015

Instance Segmentation: Cascades

Similar to
Faster R-CNN

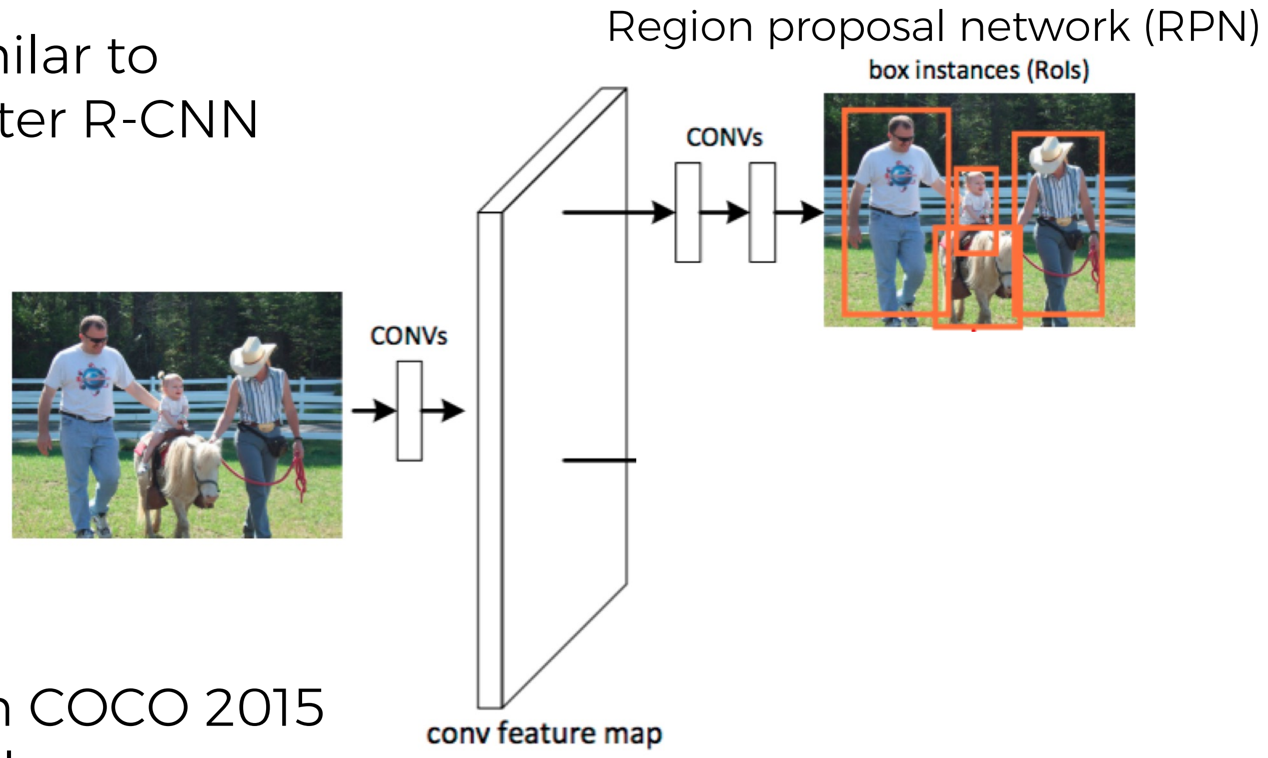


Won COCO 2015
challenge
(with ResNet)

Dai et al, "Instance-aware Semantic Segmentation via Multi-task Network Cascades", arXiv 2015

Instance Segmentation: Cascades

Similar to
Faster R-CNN



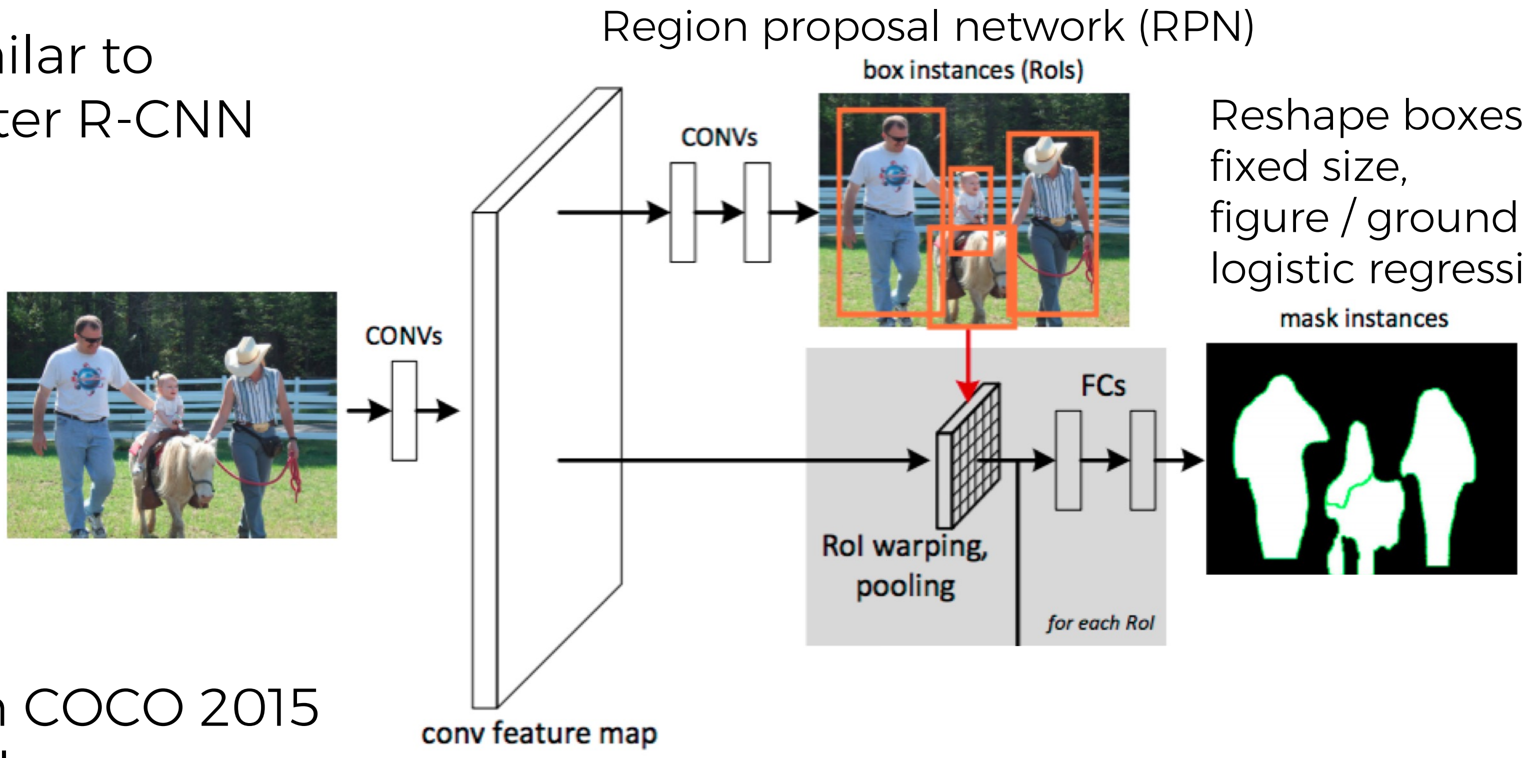
Won COCO 2015
challenge
(with ResNet)

Dai et al, "Instance-aware Semantic Segmentation via Multi-task Network Cascades", arXiv 2015

Instance Segmentation: Cascades

Similar to Faster R-CNN

Won COCO 2015 challenge (with ResNet)



Reshape boxes to fixed size, figure / ground logistic regression

Dai et al, "Instance-aware Semantic Segmentation via Multi-task Network Cascades", arXiv 2015

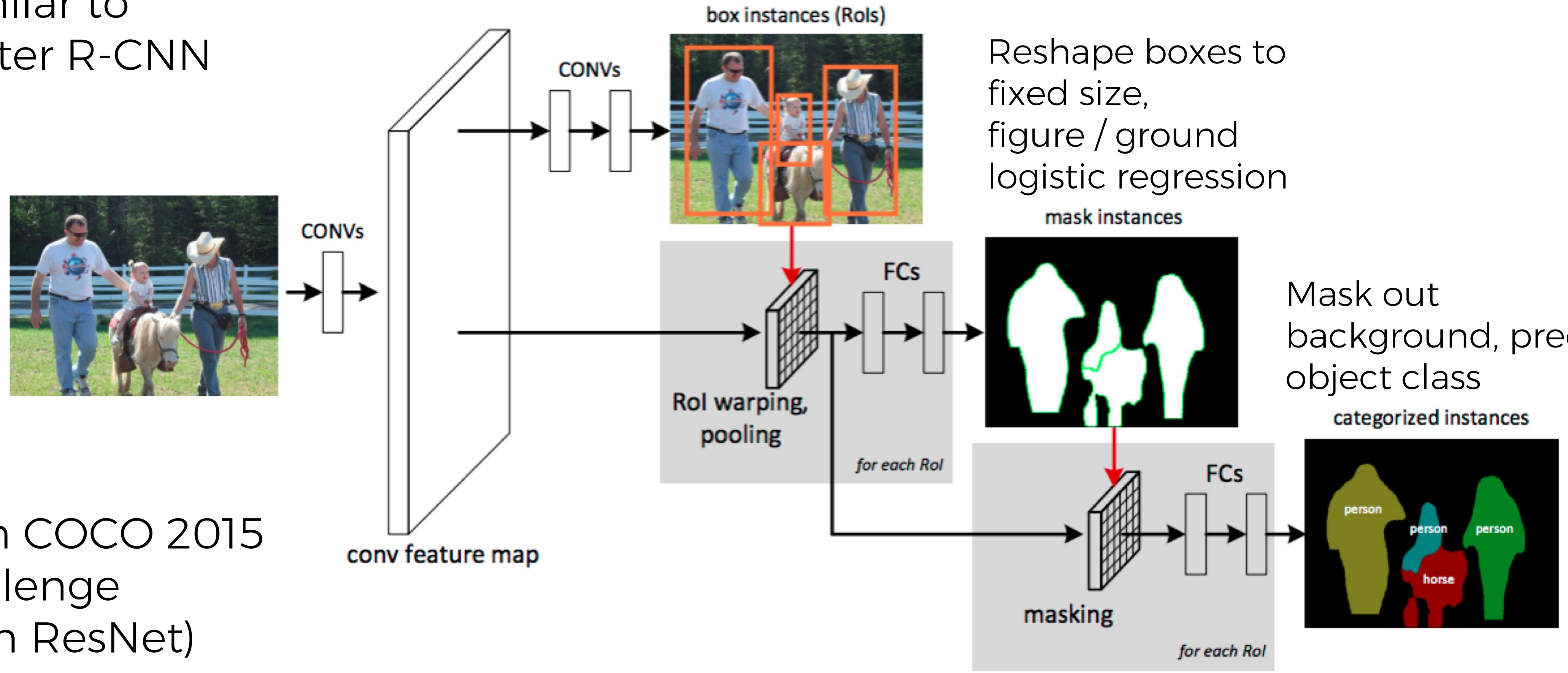
Instance Segmentation: Cascades

Similar to Faster R-CNN

Region proposal network (RPN)

Reshape boxes to fixed size, figure / ground logistic regression

Mask out background, predict object class



Won COCO 2015 challenge (with ResNet)

Dai et al, "Instance-aware Semantic Segmentation via Multi-task Network Cascades", arXiv 2015

Instance Segmentation: Cascades

Similar to Faster R-CNN

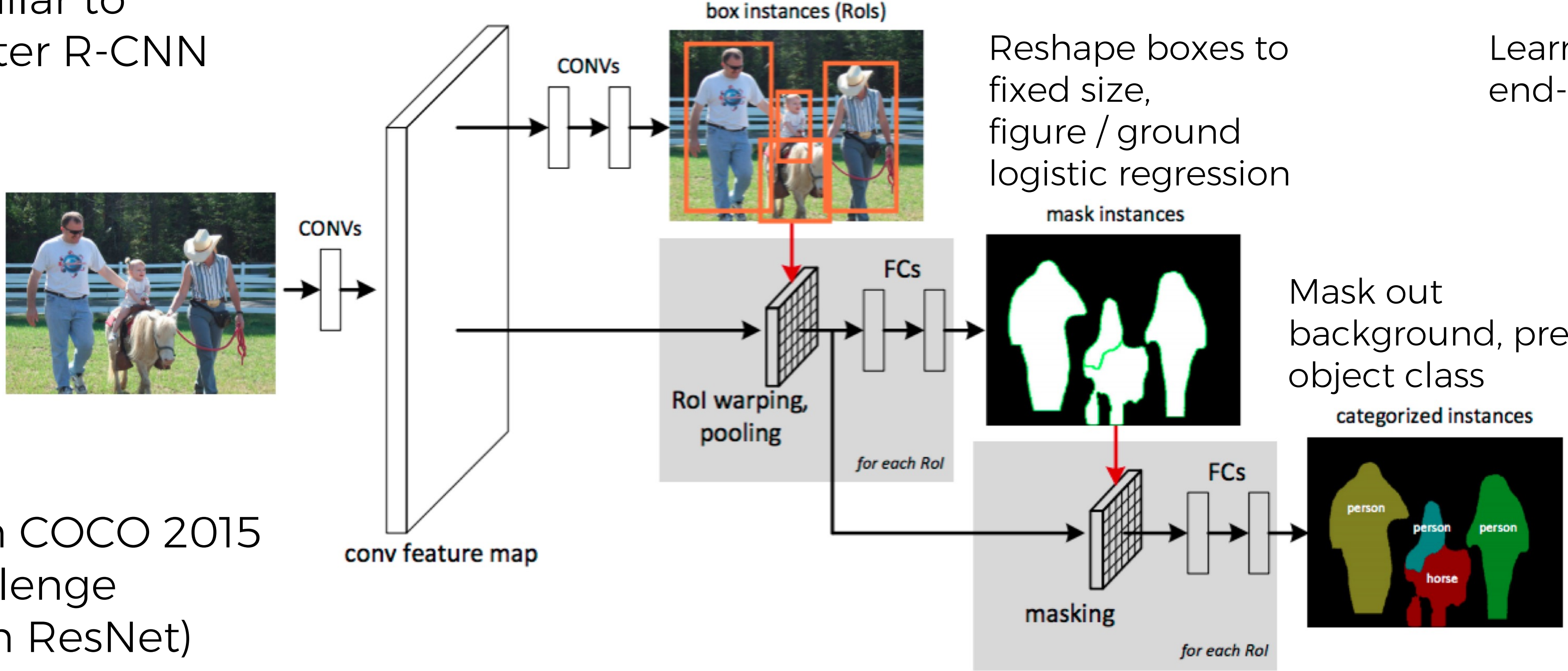
Won COCO 2015 challenge (with ResNet)

Region proposal network (RPN)

Reshape boxes to fixed size, figure / ground logistic regression

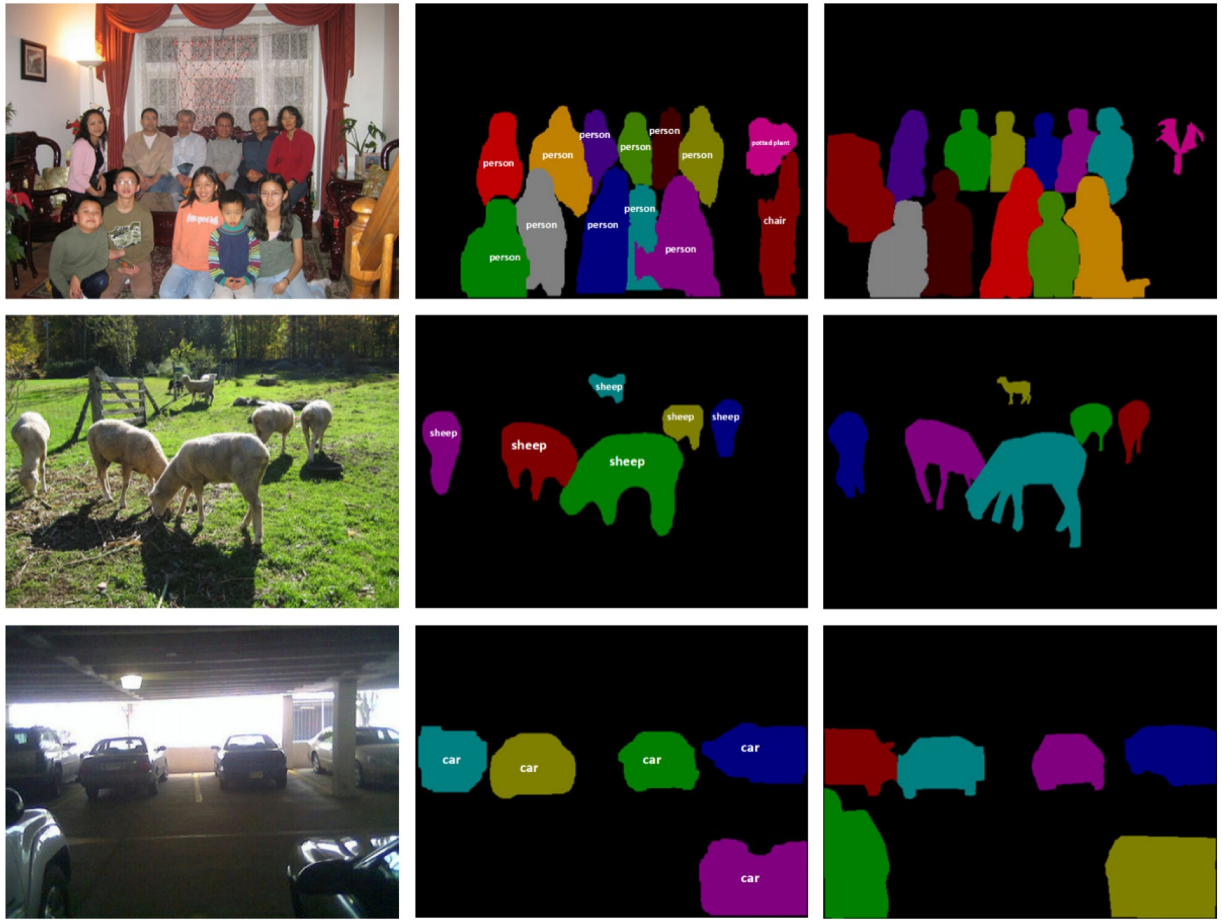
Mask out background, predict object class

Learn entire model end-to-end!



Dai et al, “Instance-aware Semantic Segmentation via Multi-task Network Cascades”, arXiv 2015

Instance Segmentation: Cascades



Predictions Ground truth

Dai et al, "Instance-aware Semantic Segmentation via Multi-task Network Cascades", arXiv 2015

Segmentation Overview

- Semantic segmentation
  - Classify all pixels
  - Fully convolutional models, downsample then upsample
  - Learnable upsampling: fractionally strided convolution
  - Skip connections can help
- Instance Segmentation
  - Detect instance, generate mask
  - Similar pipelines to object detection