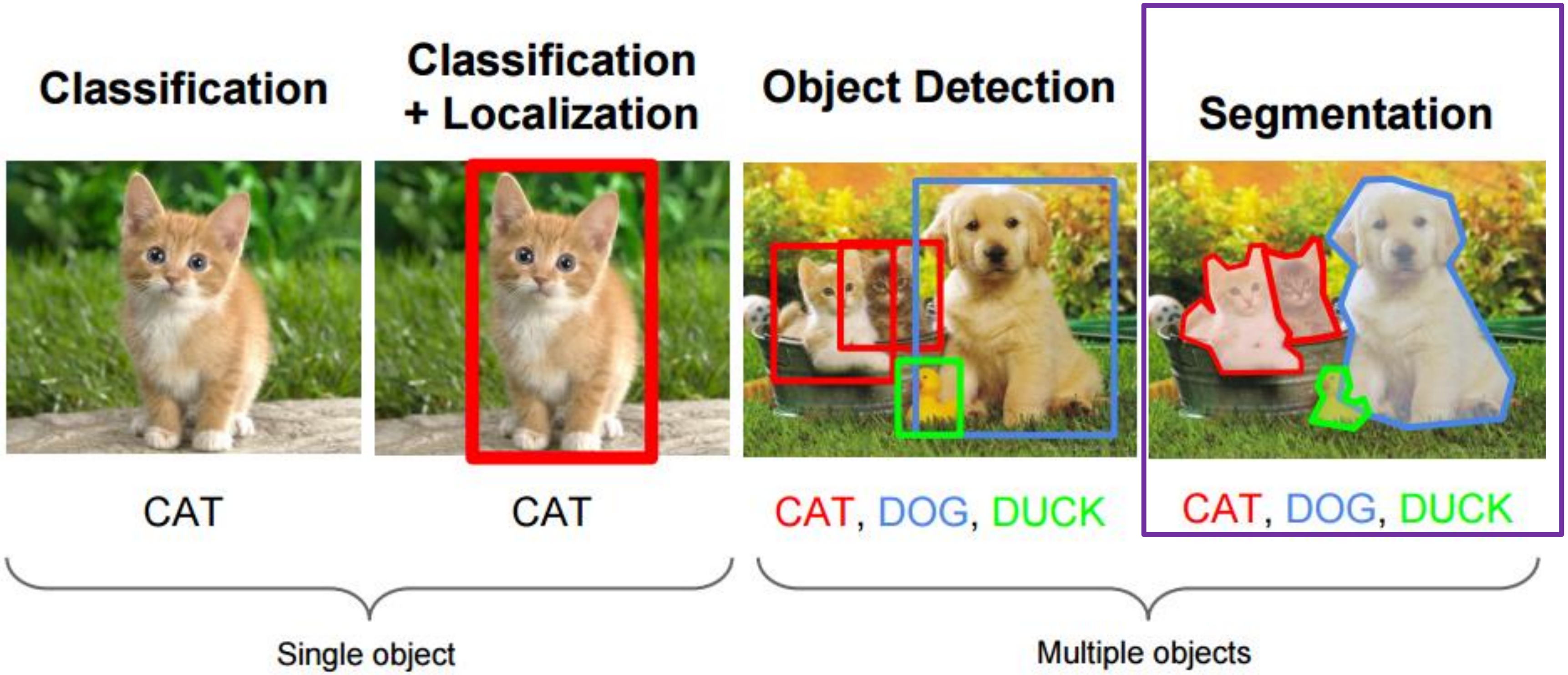
# SEGMENTATION

HYUNG IL KOO

# Computer Vision Tasks



Classification

Classification + Localization

Object Detection

Segmentation

CAT

CAT

CAT, DOG, DUCK

CAT, DOG, DUCK

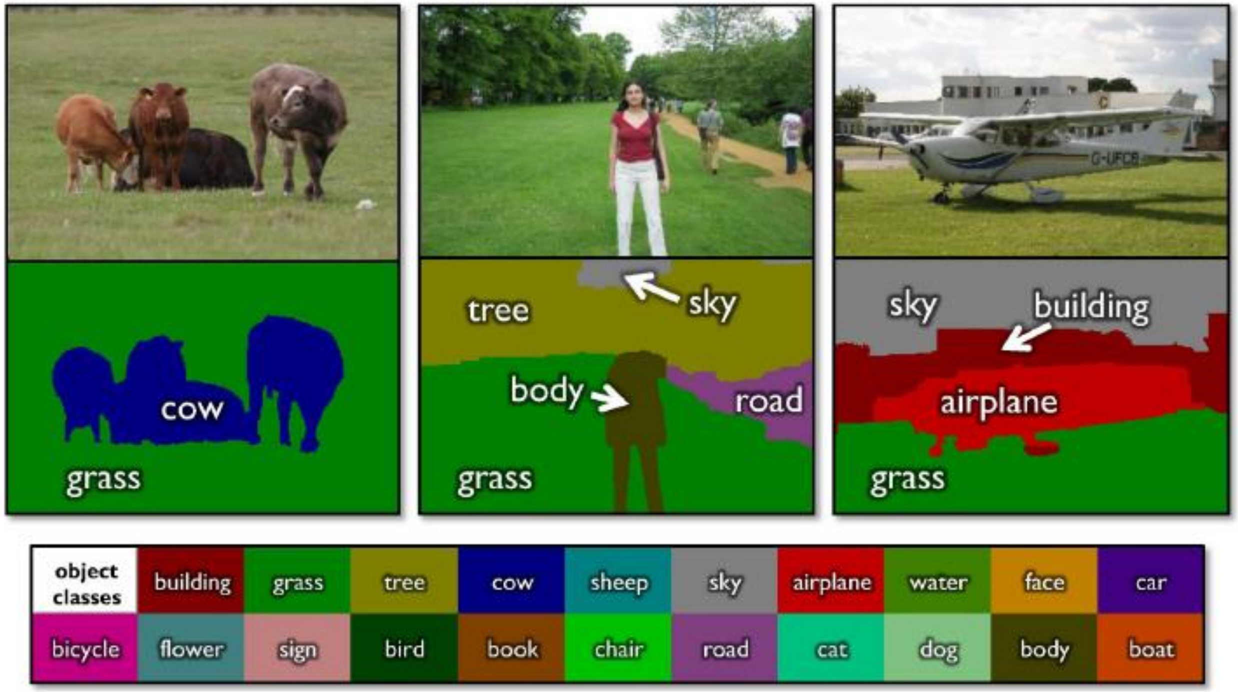
Single object

Multiple objects

Fei-Fei Li & Andrej Karpathy & Justin Johnson

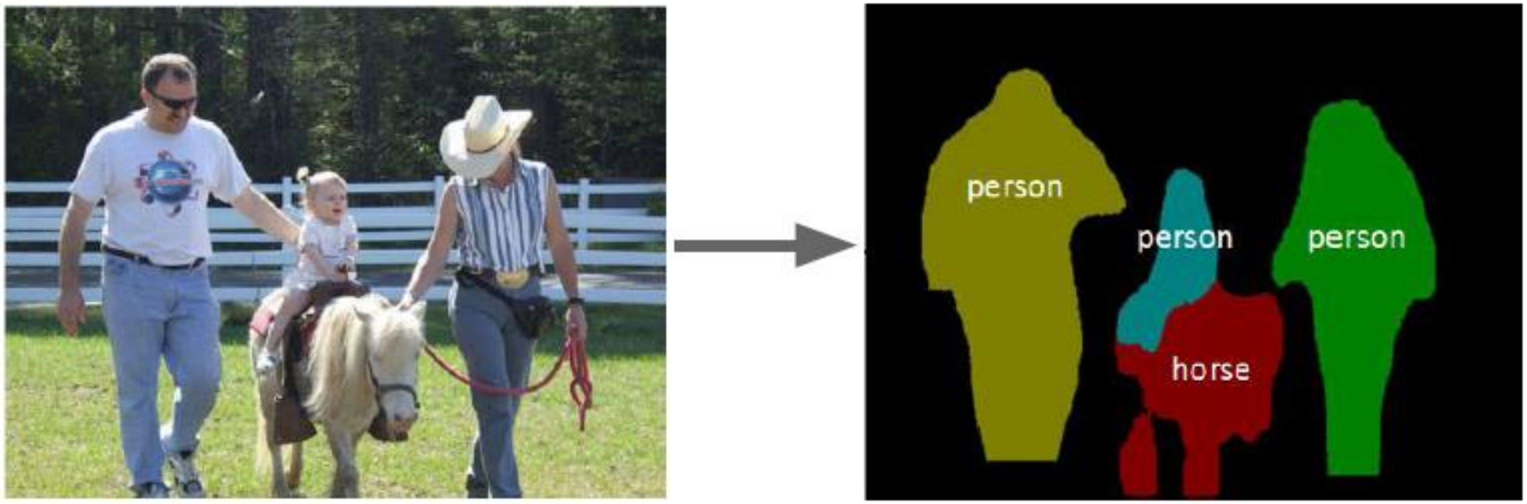
# Semantic segmentation

- Label every pixel
- Don’t differentiate instances



# Instance Segmentation

- Detect instances, give category, label pixels



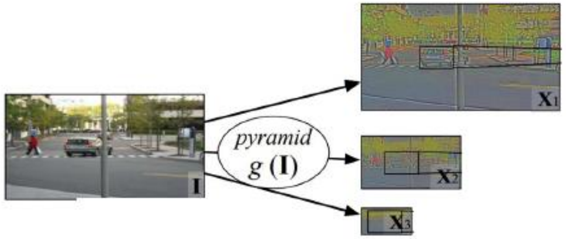
# LEARNING HIERARCHICAL FEATURES FOR SCENE LABELING

# Semantic Segmentation

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

# Semantic Segmentation

Resize image to multiple scales

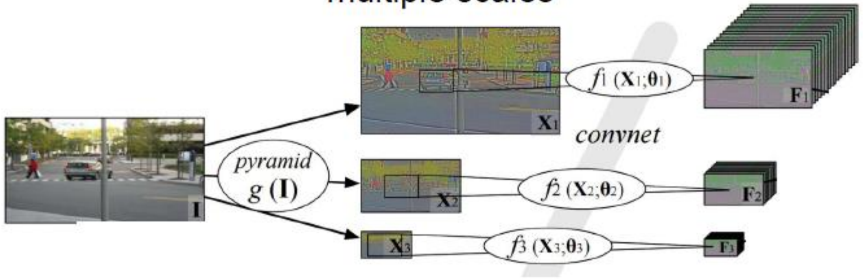


Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013

# Semantic Segmentation

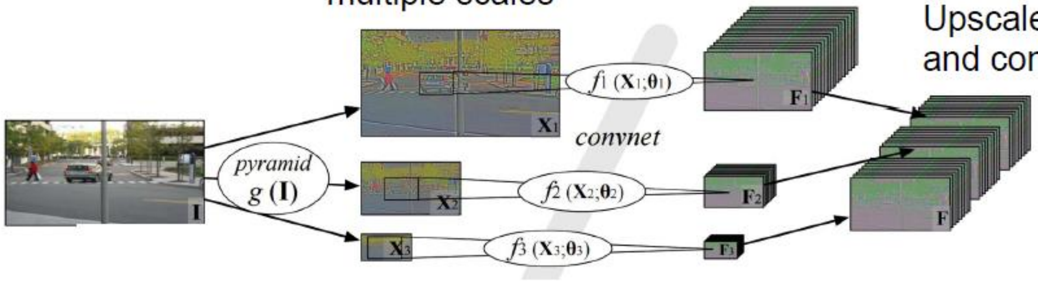
Resize image to multiple scales

Run one CNN per scale



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

# Semantic Segmentation



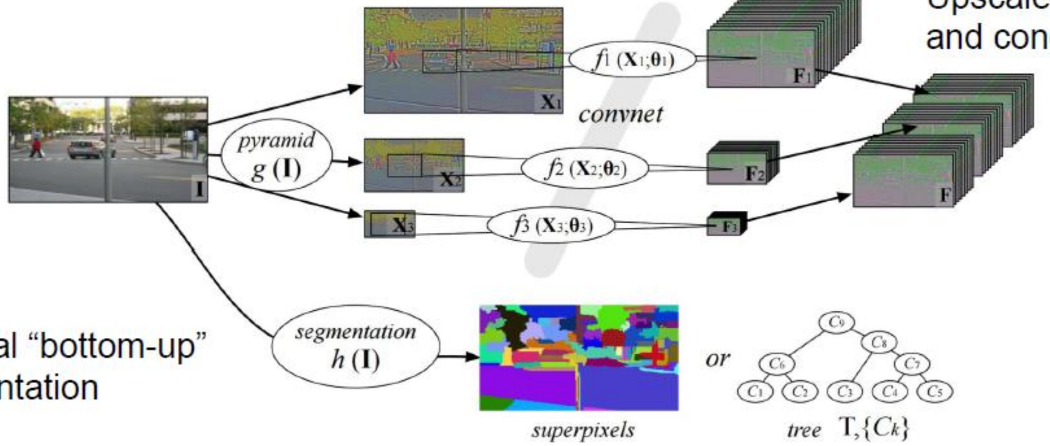
Resize image to multiple scales

Run one CNN per scale

Upscale outputs and concatenate

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

# Semantic Segmentation



Resize image to multiple scales

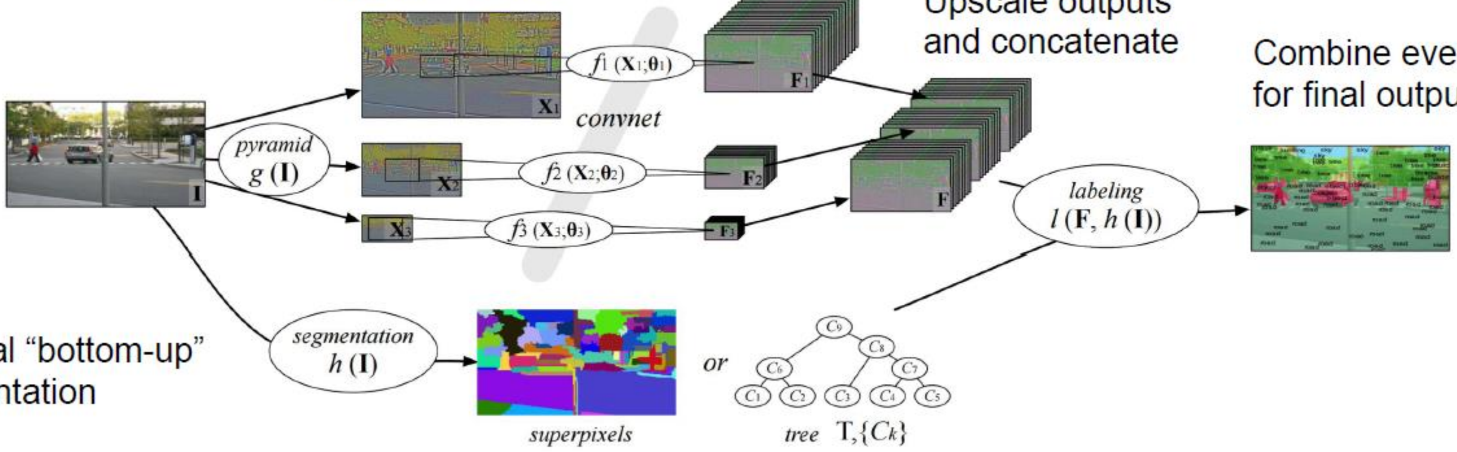
Run one CNN per scale

Upscale outputs and concatenate

External "bottom-up" segmentation

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

# Semantic Segmentation



Resize image to multiple scales

Run one CNN per scale

Upscale outputs and concatenate

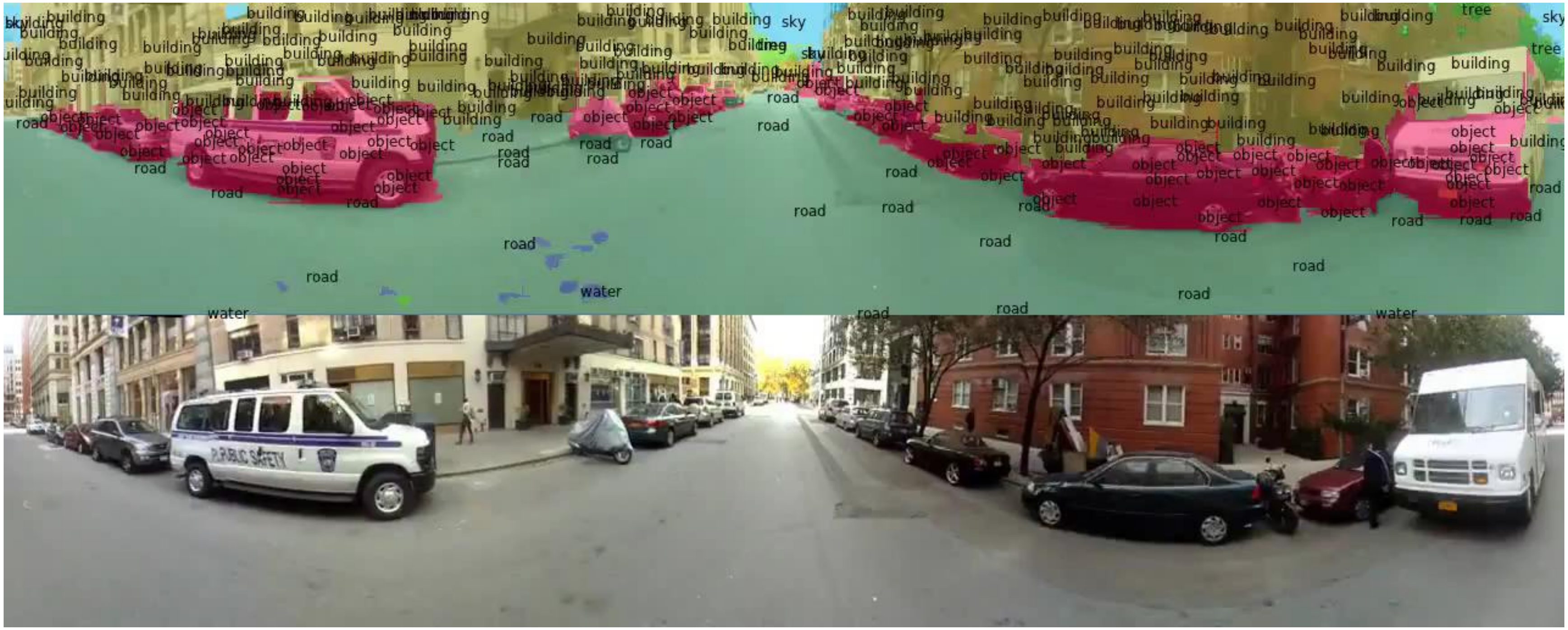
Combine everything for final outputs

External "bottom-up" segmentation

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

# Semantic Segmentation

- Building/road/sky/object/grass/water/tree

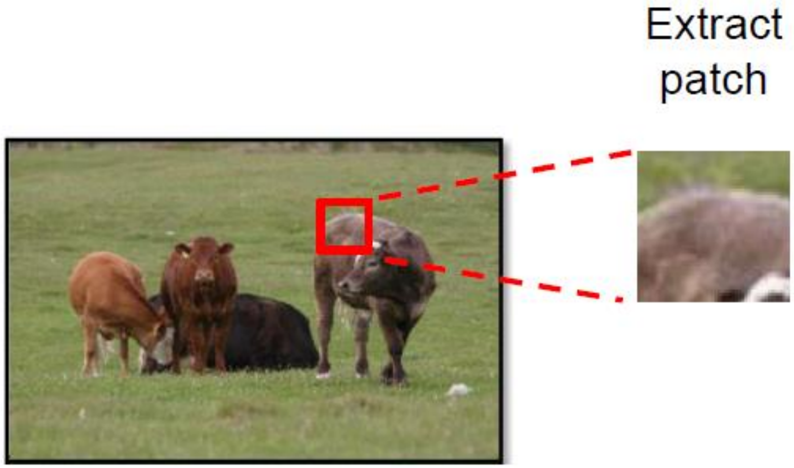


Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013

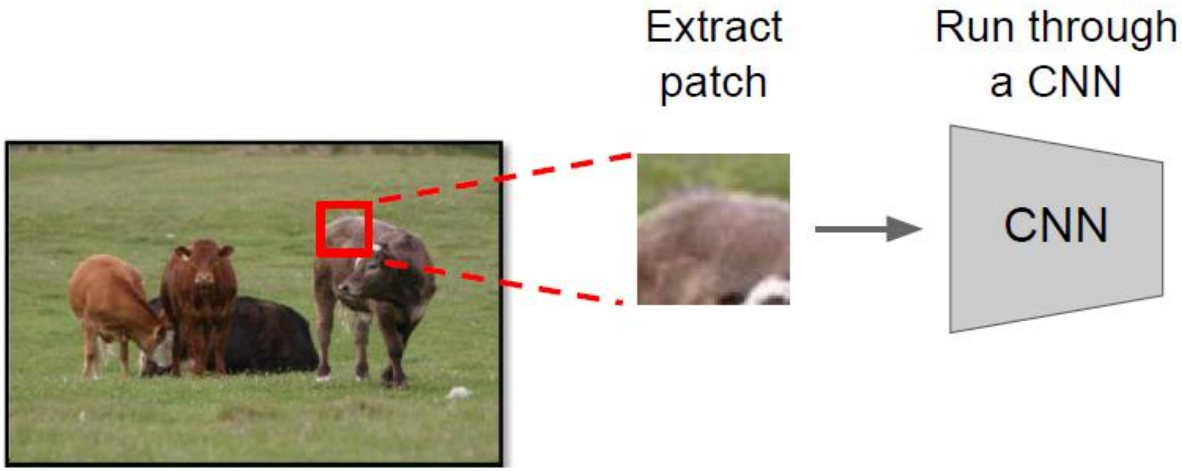
# PIXEL-WISE CLASSIFICATION

# Semantic Segmentation

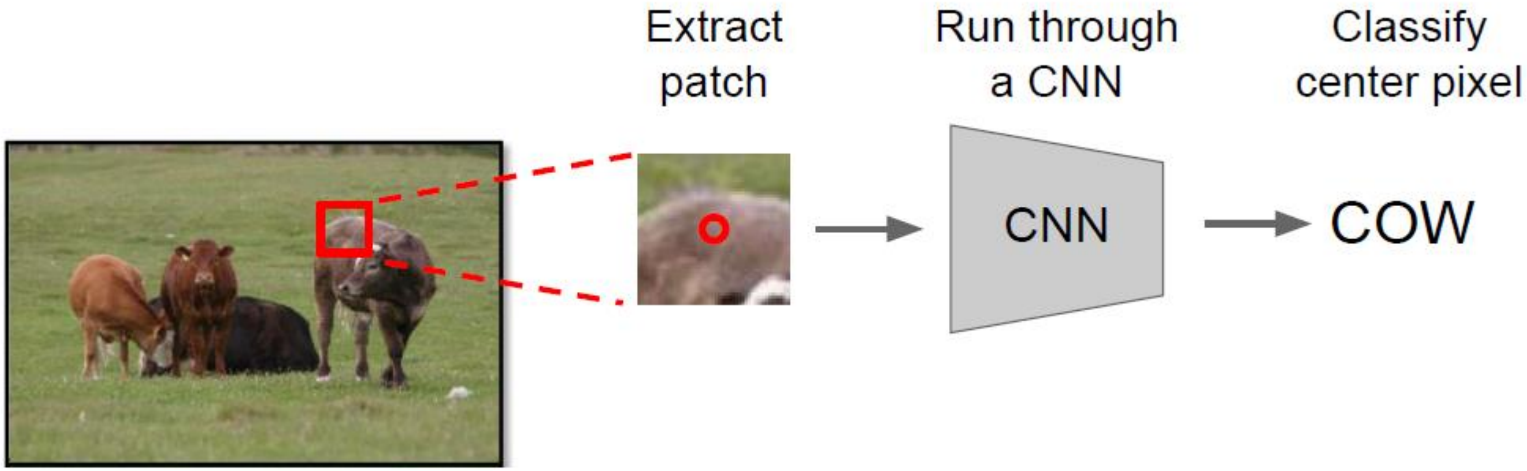
# Semantic Segmentation



# Semantic Segmentation



# Semantic Segmentation



Extract patch

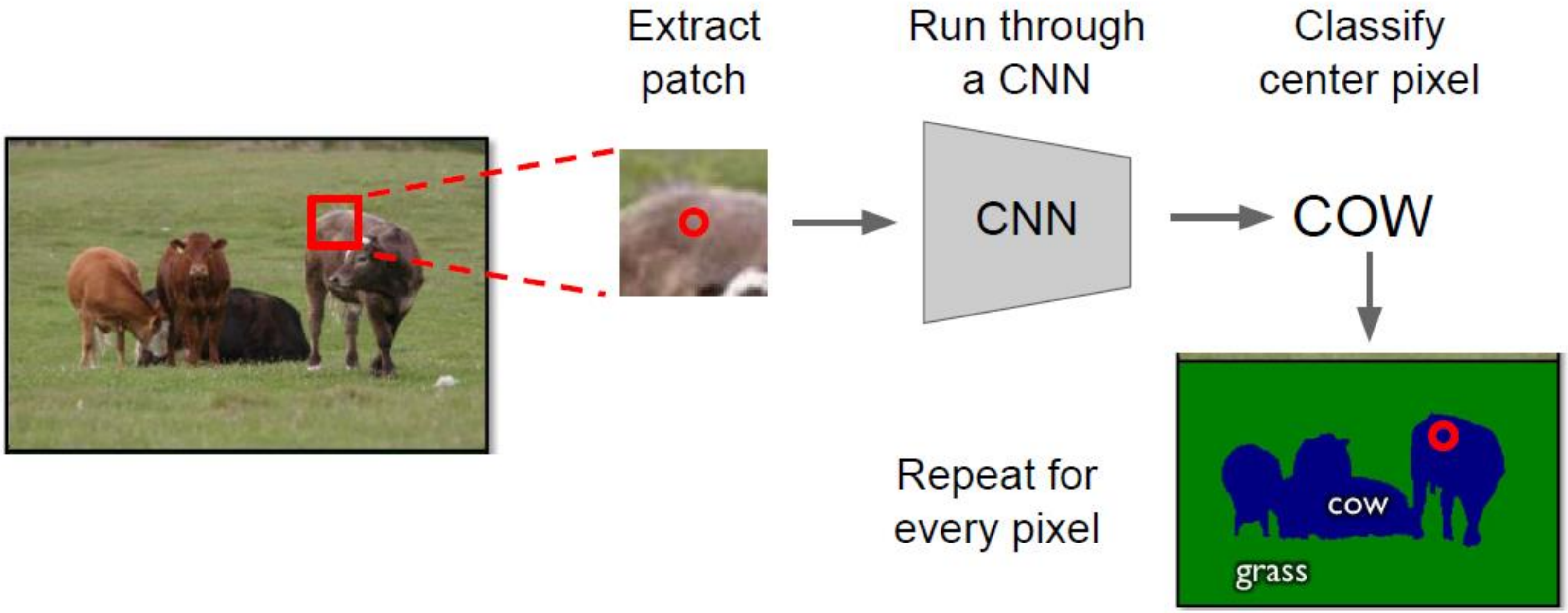
Run through a CNN

Classify center pixel

CNN

COW

# Semantic Segmentation



Extract patch

Run through a CNN

Classify center pixel

CNN

COW

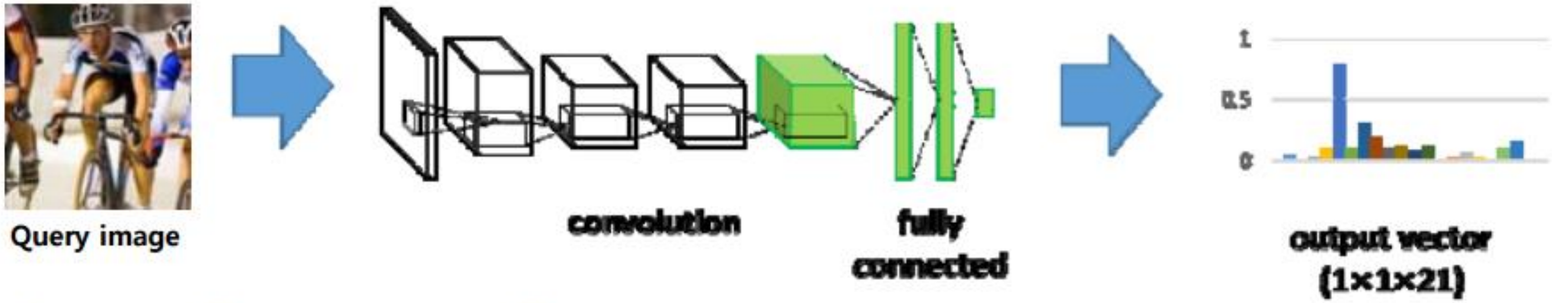
Repeat for every pixel

# ENCODER-DECODER ARCHITECTURE

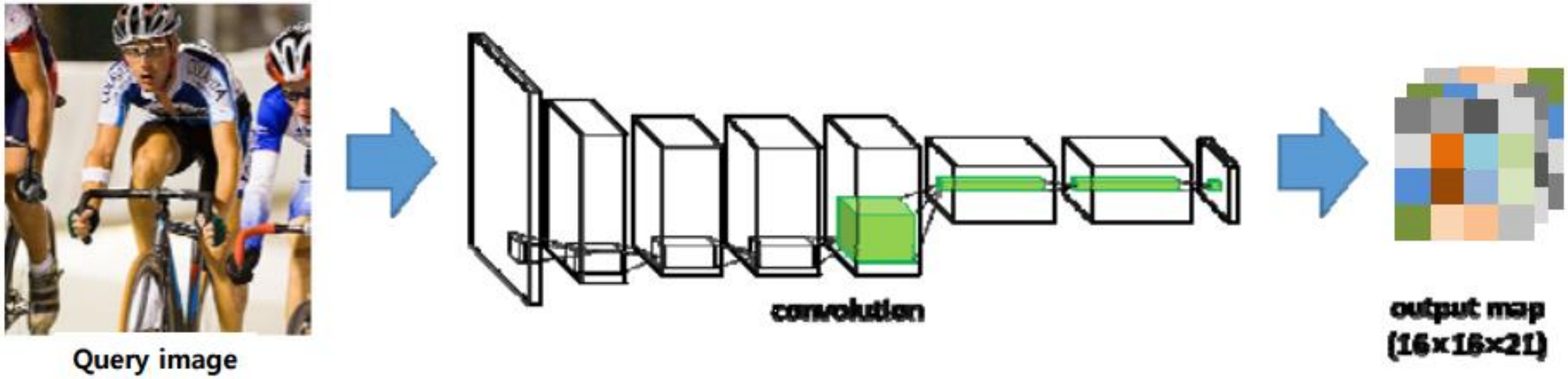
# FULLY CONNECTED LAYERS AS CONVOLUTION LAYERS

# FC layer as Conv layer

- Image classification



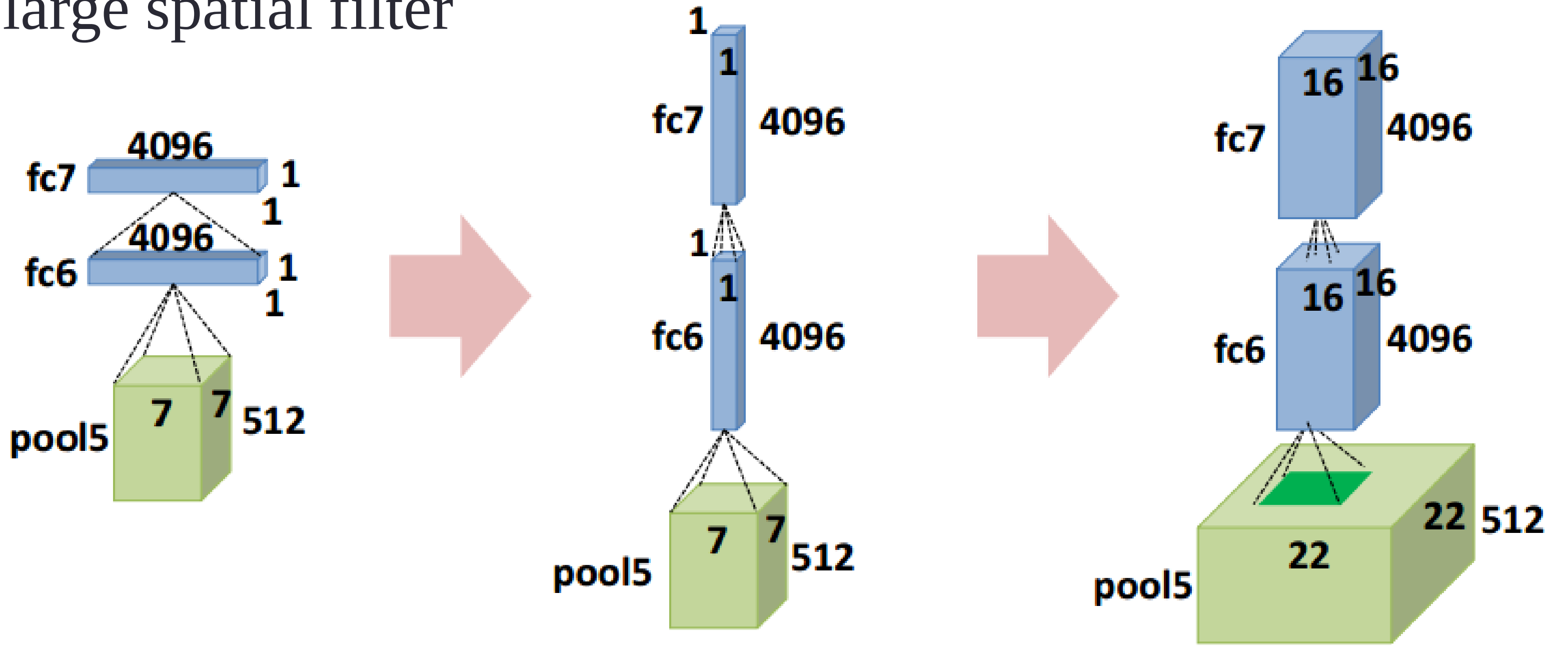
- Semantic segmentation
  - Given an input image, obtain pixel-wise segmentation mask using a deep Convolutional Neural Network (CNN)



Deconvolutions in Convolutional Neural Networks
By Prof. Bohyung Han

# FC layer as Conv layer

- Each fully connected layer is interpreted as a convolution with a large spatial filter



Fully connected layers

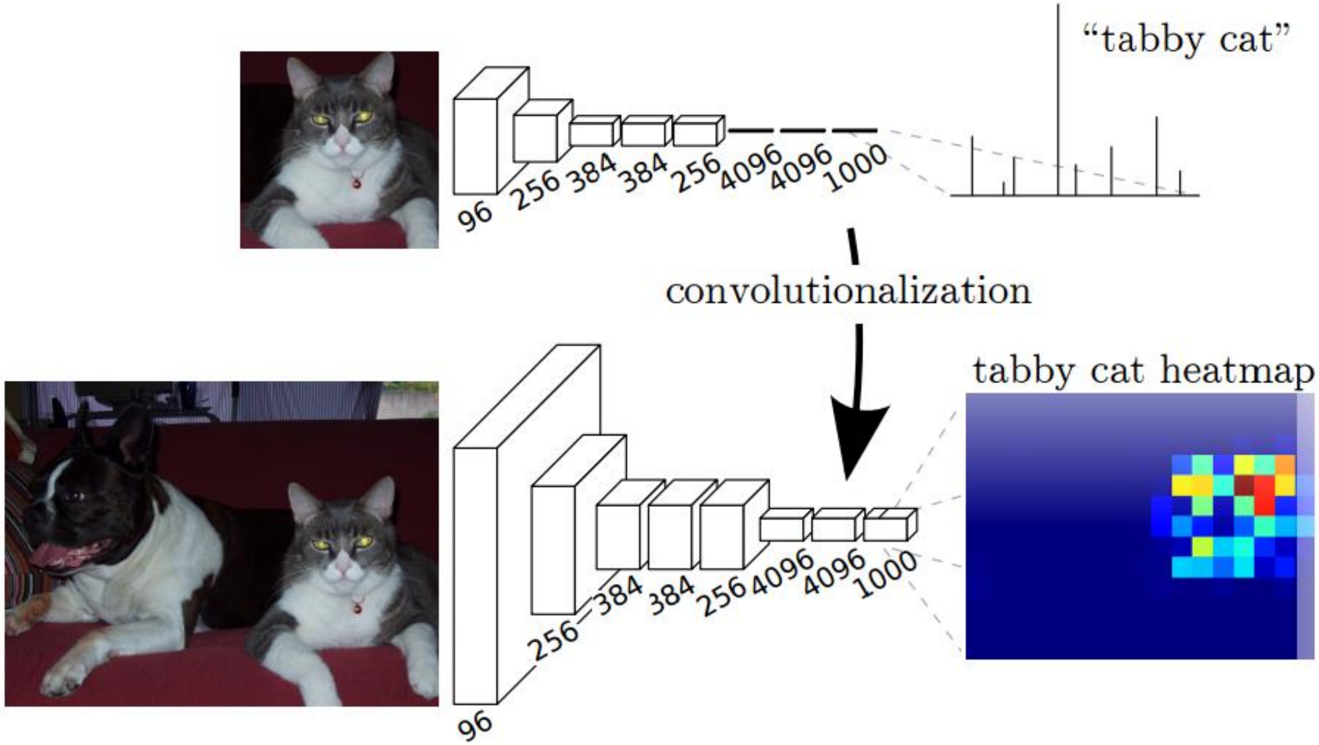
Convolution layers

For the larger Input field

Deconvolutions in Convolutional Neural Networks
By Prof. Bohyung Han

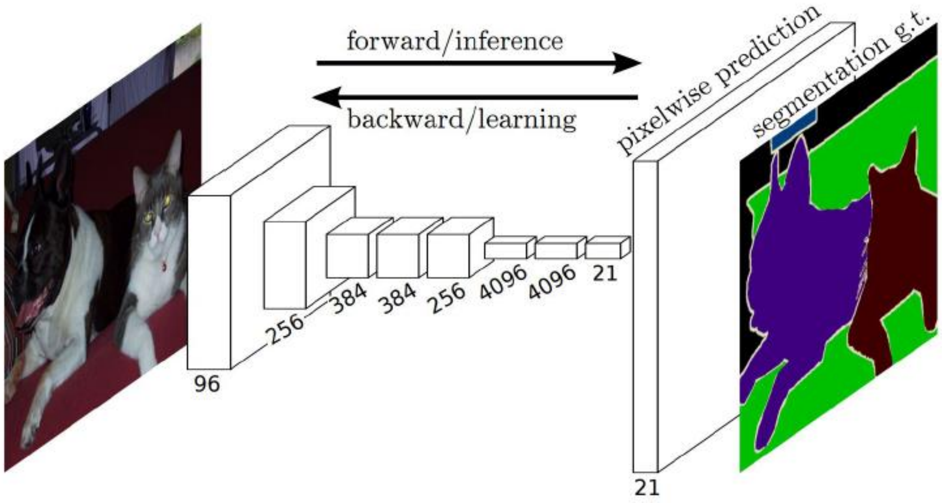
# FC layer as Conv layer

- Transforming fully connected layers into convolution layers enables a classification net to output a heatmap



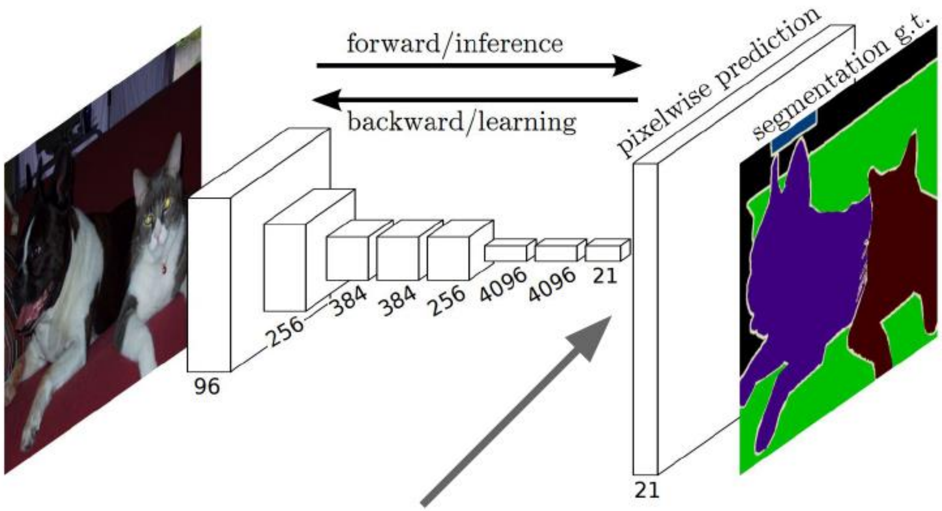
# SEMANTIC SEGMENTATION WITH UPSAMPLING

# Semantic segmentation: upsampling



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

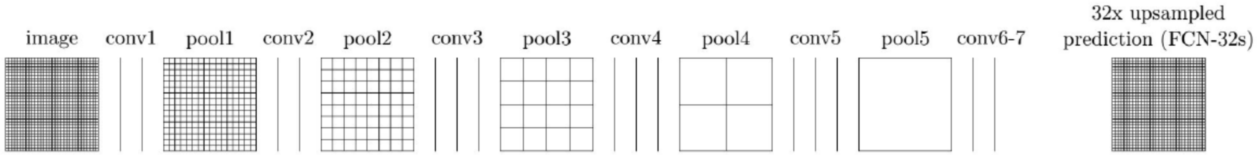
# Semantic segmentation: upsampling



Learnable upsampling!

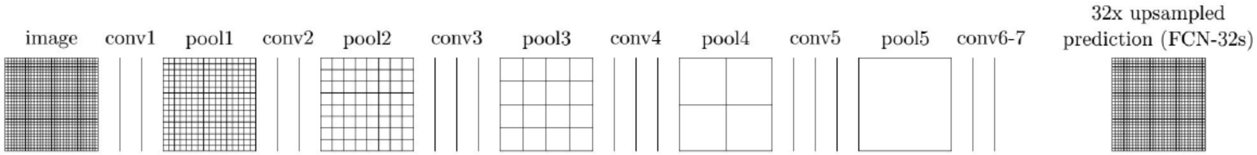
Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

# Semantic segmentation: upsampling



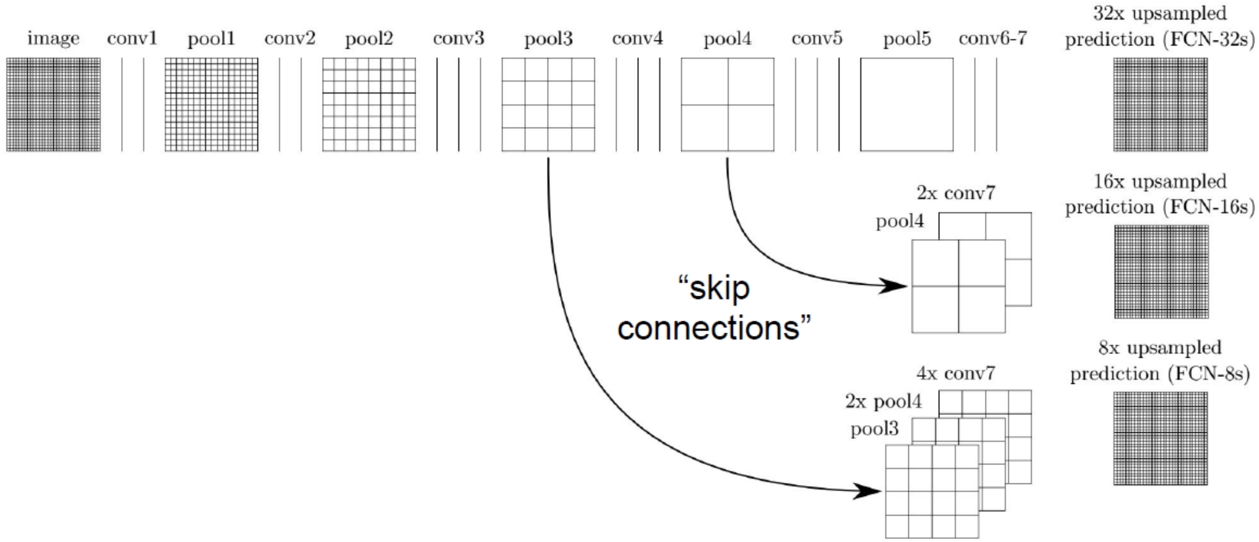
Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

# Semantic segmentation: upsampling



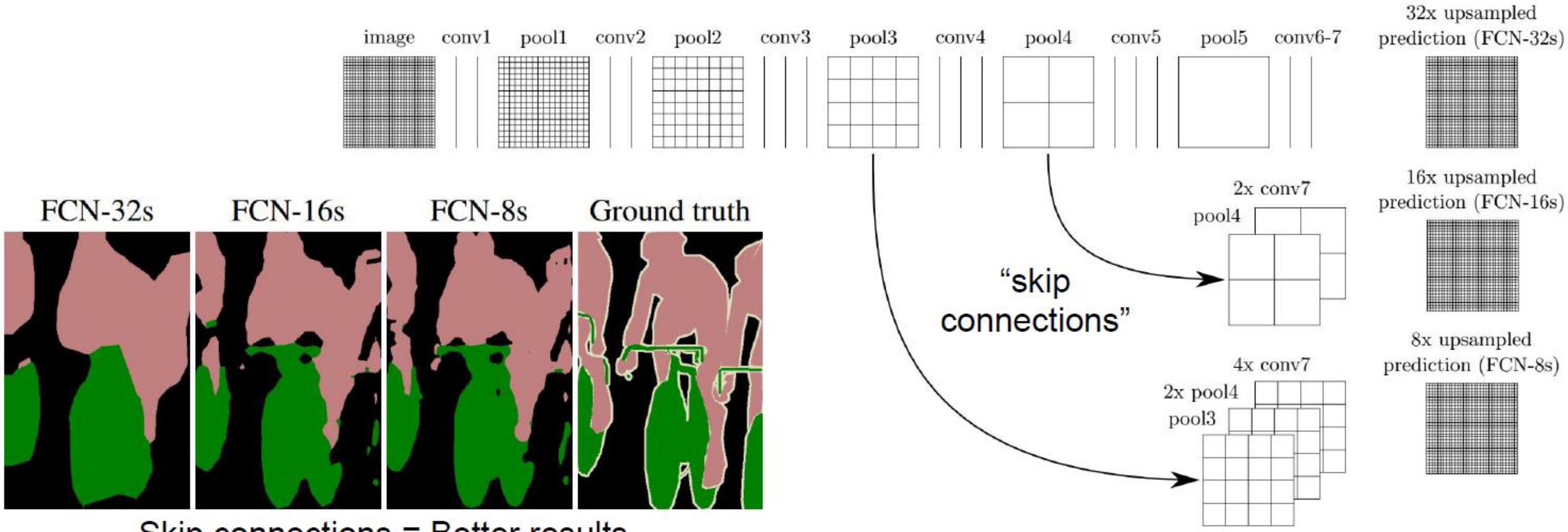
Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

# Semantic segmentation: upsampling



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

# Semantic segmentation: upsampling

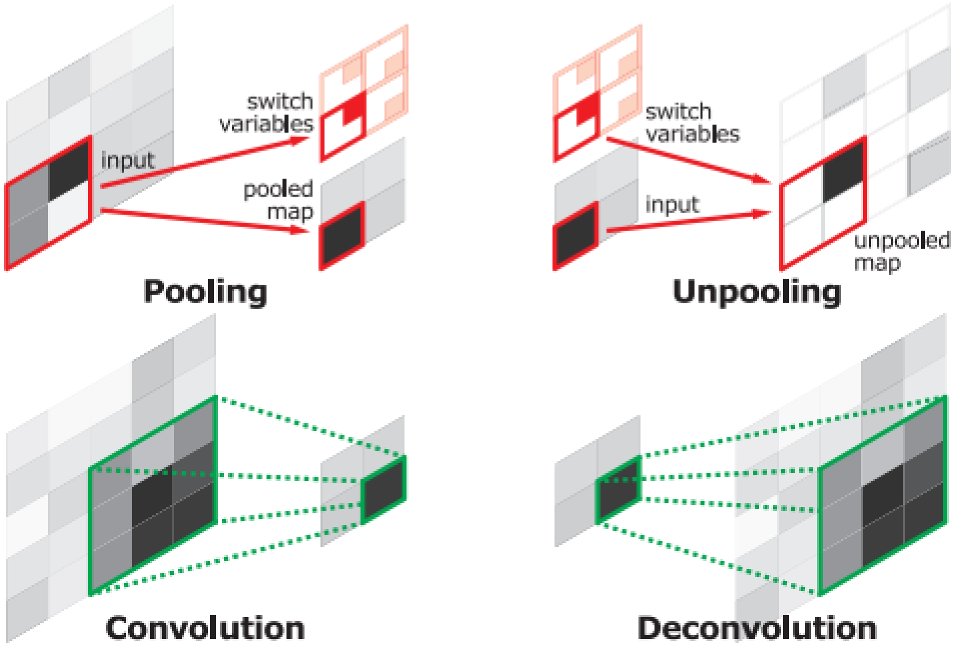


Skip connections = Better results

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

# LEARNING DECONVOLUTION NETWORK FOR SEMANTIC SEGMENTATION

# Illustration of deconvolution and unpooling operations.



Noh. et al. “Learning Deconvolution Network for Semantic Segmentation,” ICCV 2015

# Overall architecture

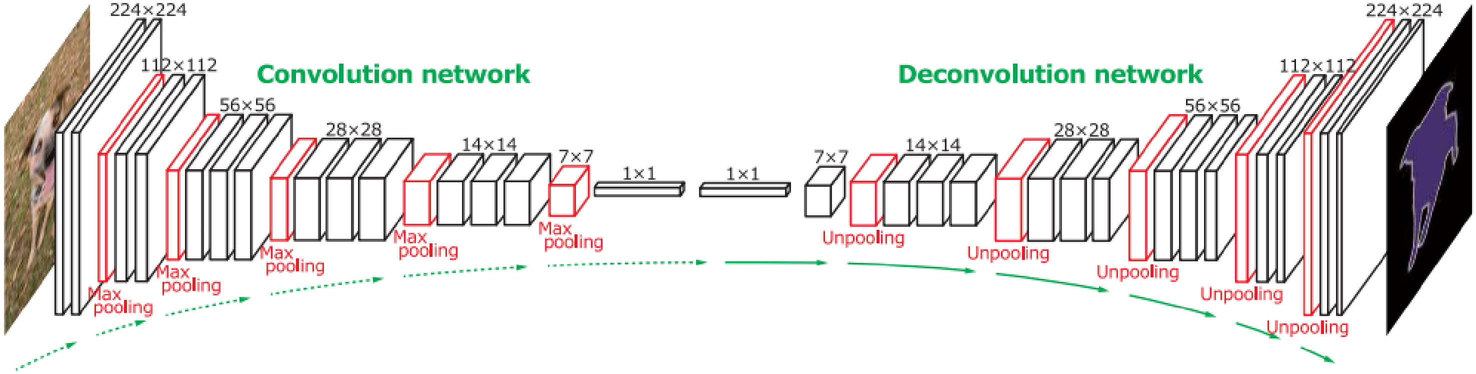


Figure 2. Overall architecture of the proposed network. On top of the convolution network based on VGG 16-layer net, we put a multi-layer deconvolution network to generate the accurate segmentation map of an input proposal. Given a feature representation obtained from the convolution network, dense pixel-wise class prediction map is constructed through multiple series of unpooling, deconvolution and rectification operations.

Noh. et al. “Learning Deconvolution Network for Semantic Segmentation,” ICCV 2015

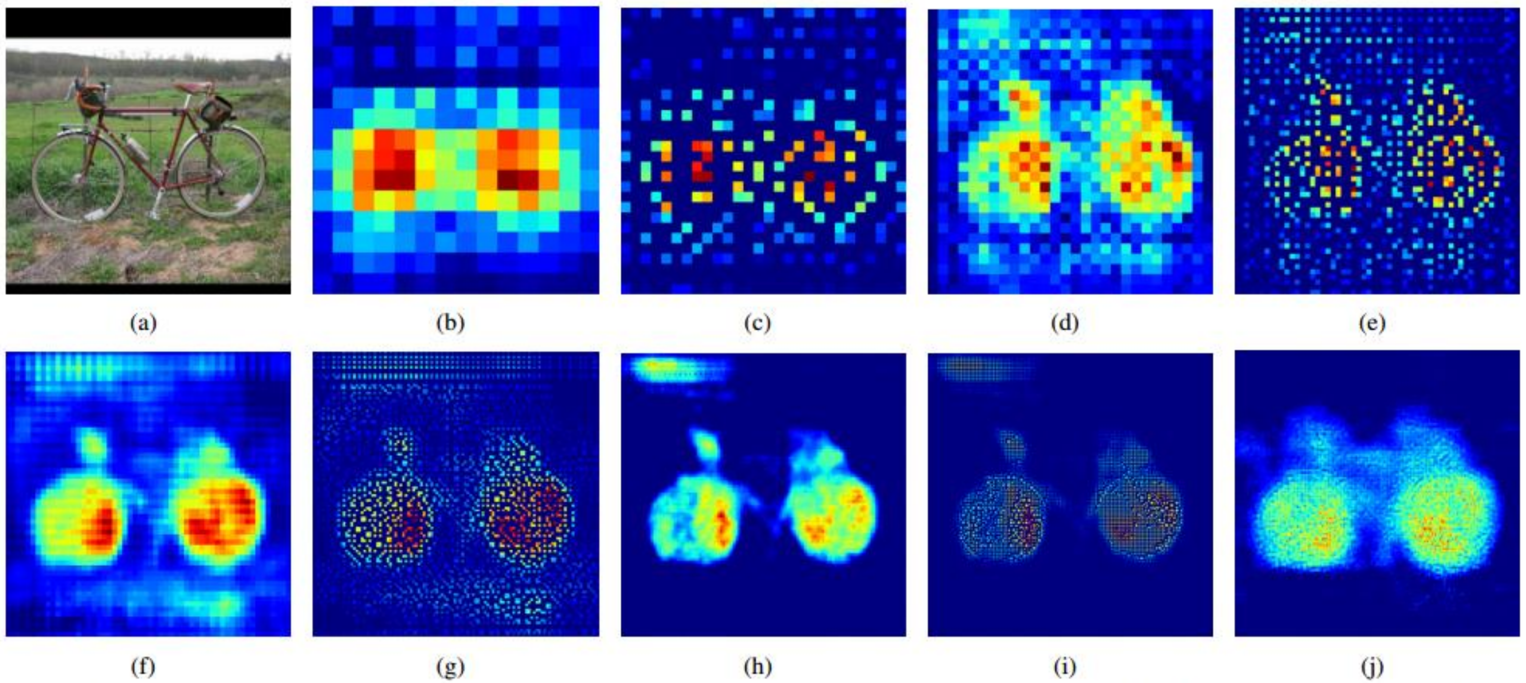


Figure 4. Visualization of activations in our deconvolution network. The activation maps from top left to bottom right correspond to the output maps from lower to higher layers in the deconvolution network. We select the most representative activation in each layer for effective visualization. The image in (a) is an input, and the rest are the outputs from (b) the last 14 × 14 deconvolutional layer, (c) the 28 × 28 unpooling layer, (d) the last 28 × 28 deconvolutional layer, (e) the 56 × 56 unpooling layer, (f) the last 56 × 56 deconvolutional layer, (g) the 112 × 112 unpooling layer, (h) the last 112 × 112 deconvolutional layer, (i) the 224 × 224 unpooling layer and (j) the last 224 × 224 deconvolutional layer. The finer details of the object are revealed, as the features are forward-propagated through the layers in the deconvolution network. Note that noisy activations from background are suppressed through propagation while the activations closely related to the target classes are amplified. It shows that the learned filters in higher deconvolutional layers tend to capture class-specific shape information.

Noh. et al. “Learning Deconvolution Network for Semantic Segmentation,” ICCV 2015

# Semantic segmentation as an instance-wise segmentation problem

- The proposed network is trained to perform semantic segmentation for individual instances.
  - Given an input image, we first generate a sufficient number of candidate proposals, and apply the trained network to obtain semantic segmentation maps of individual proposals. Then we aggregate the outputs of all proposals to produce semantic segmentation on a whole image.

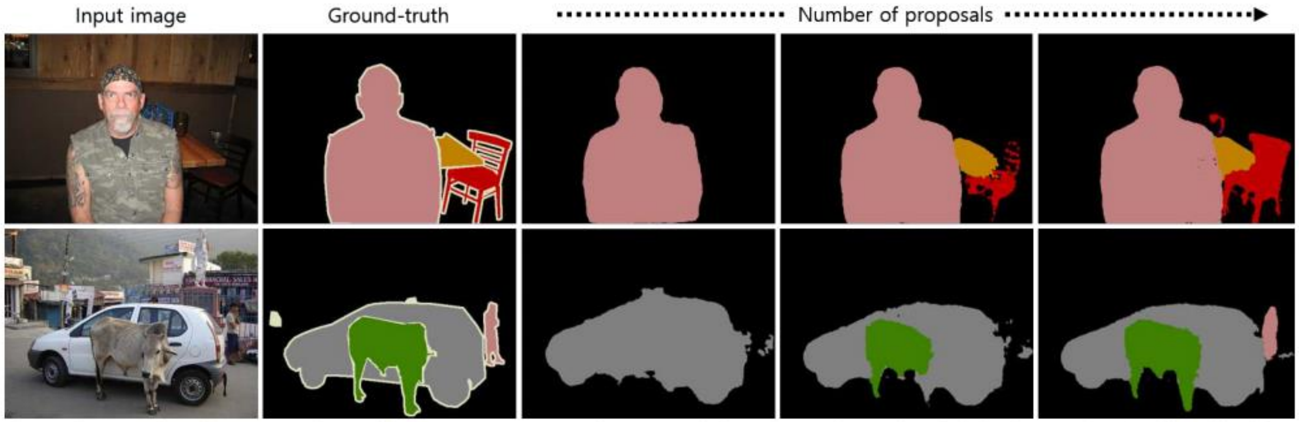
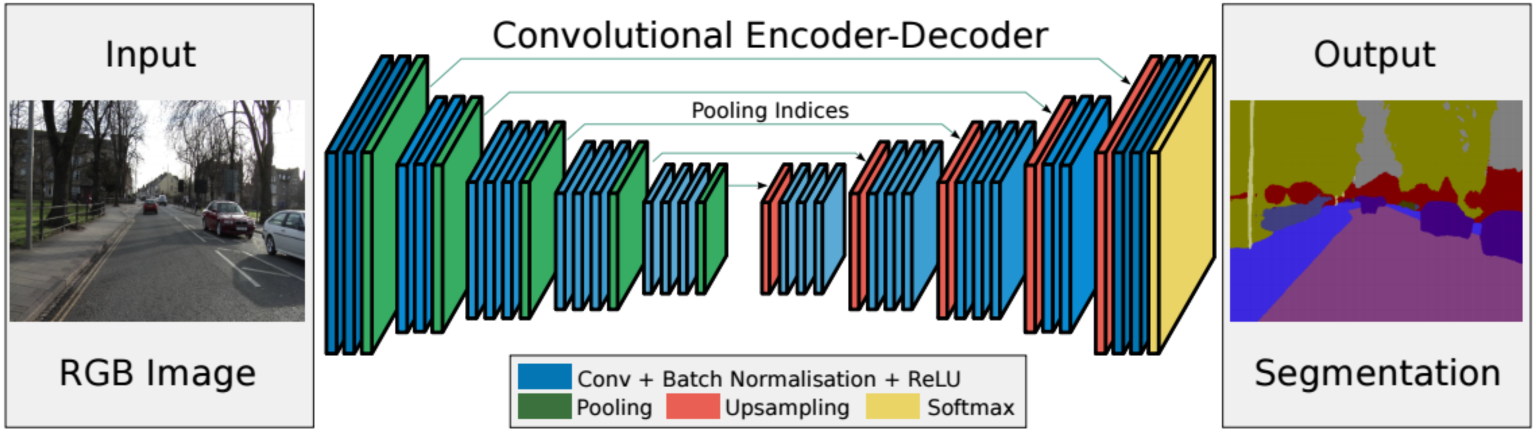


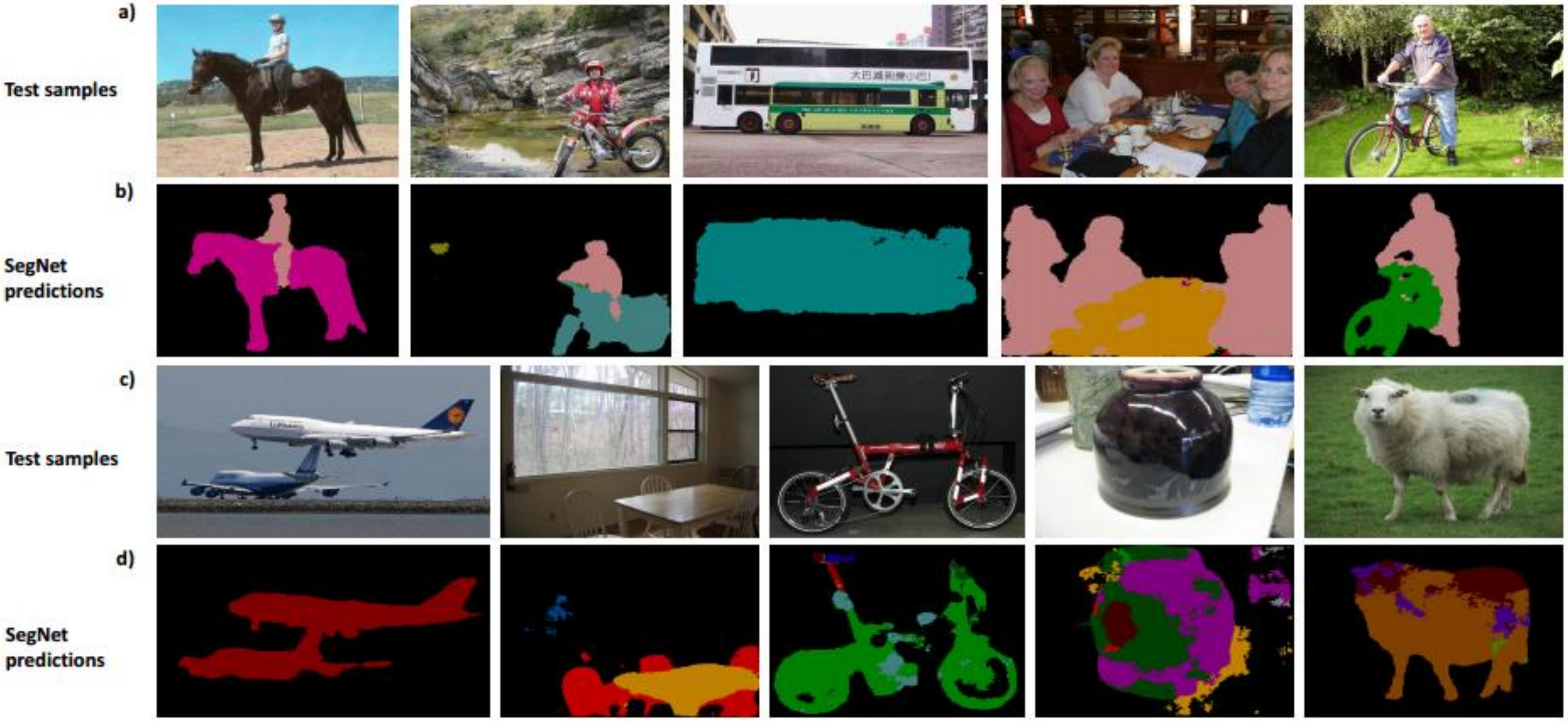
Figure 6. Benefit of instance-wise prediction. We aggregate the proposals in a decreasing order of their sizes. The algorithm identifies finer object structures through iterations by handling multi-scale objects effectively.

# SEGNET: A DEEP CONVOLUTIONAL ENCODER-DECODER ARCHITECTURE FOR IMAGE SEGMENTATION

# Encoder/Decoder



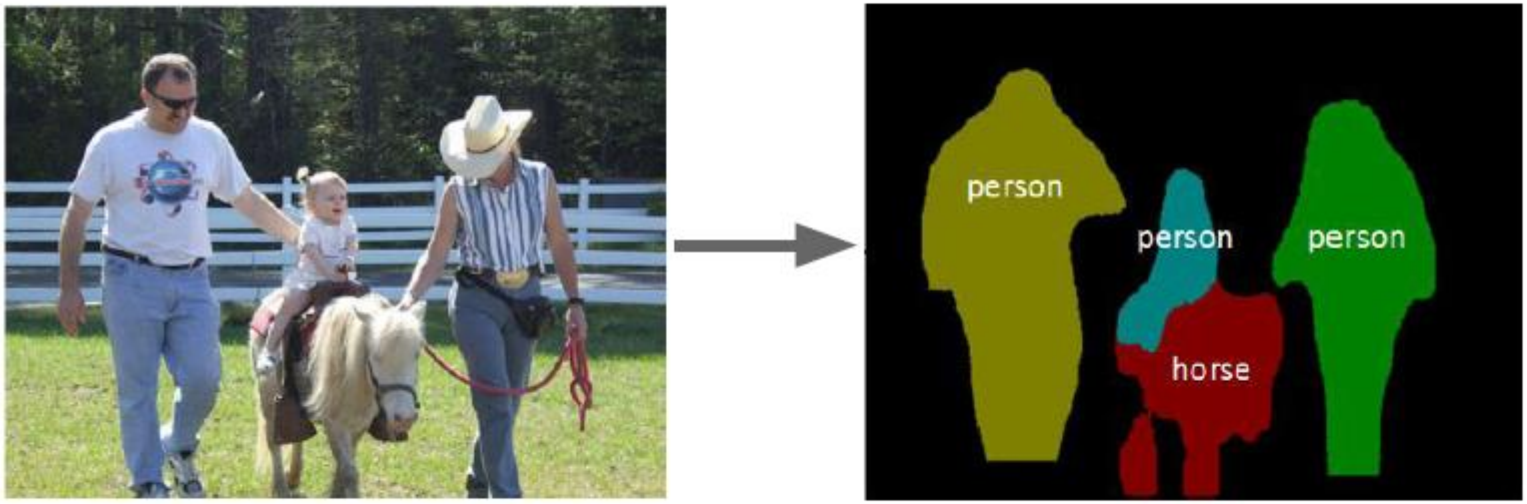
# Results


a)
Test samples
b)
SegNet predictions
c)
Test samples
d)
SegNet predictions

# INSTANCE SEGMENTATION

# Instance Segmentation

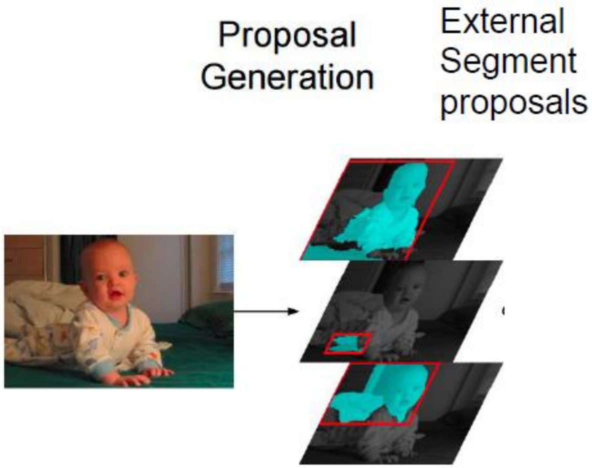
- Detect instances, give category, label pixels



# Instance Segmentation

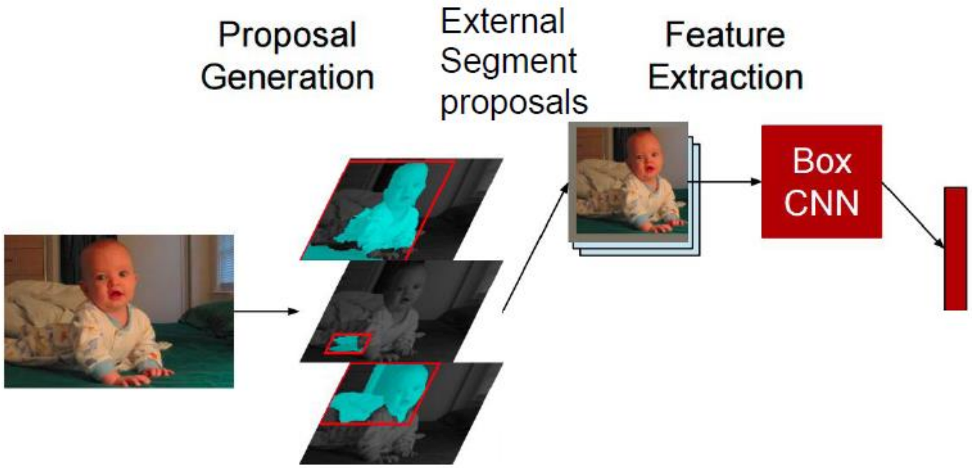
Hariharan et al, “Simultaneous Detection and Segmentation”, ECCV 2014

# Instance Segmentation



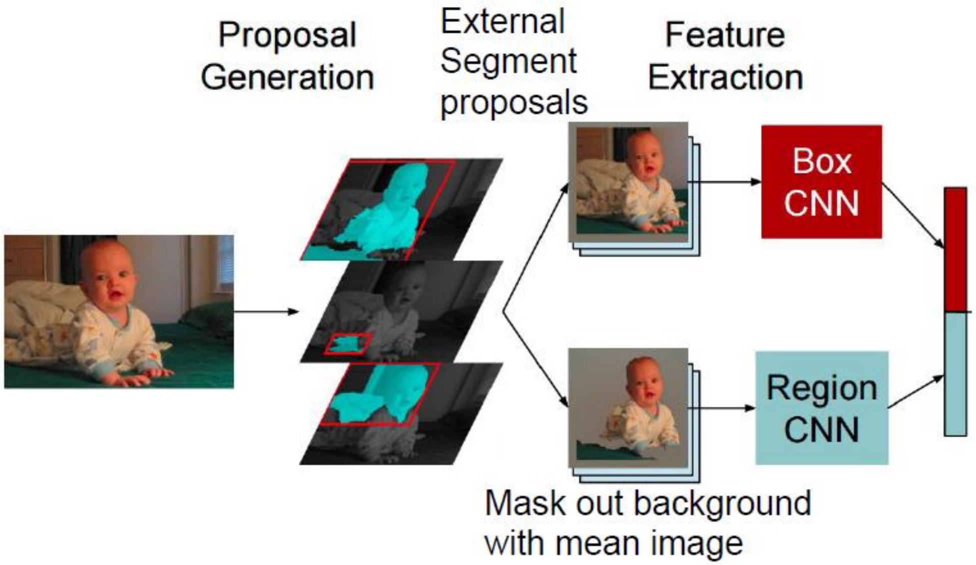
Hariharan et al, “Simultaneous Detection and Segmentation”, ECCV 2014

# Instance Segmentation



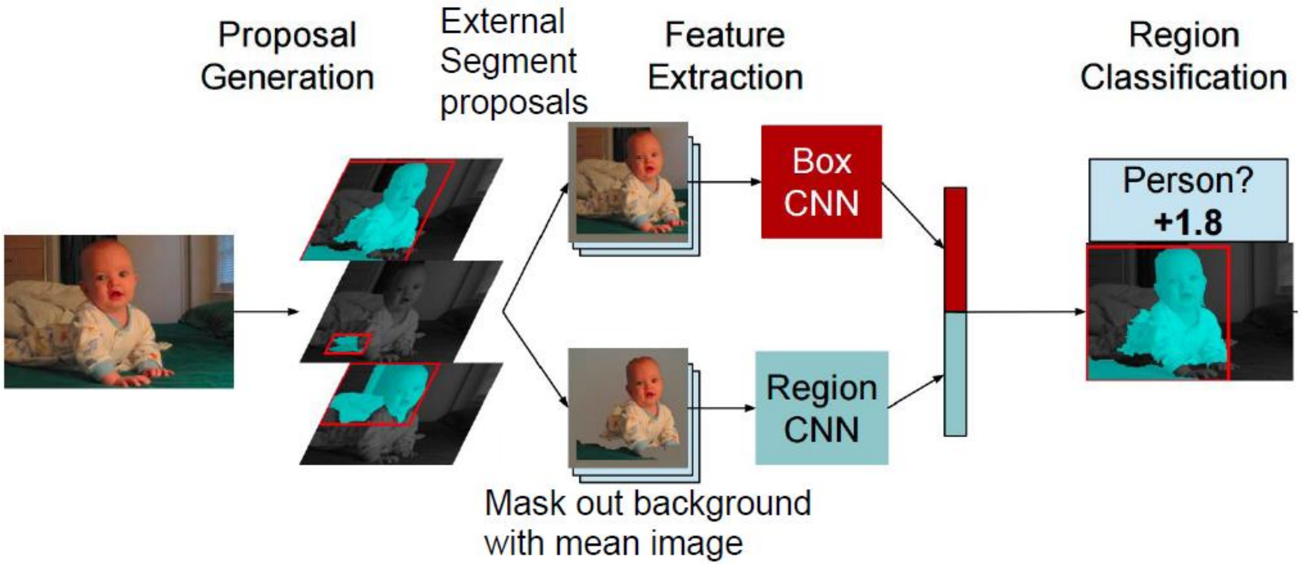
Hariharan et al, "Simultaneous Detection and Segmentation", ECCV 2014

# Instance Segmentation


Proposal Generation
External Segment proposals
Feature Extraction
Box CNN
Region CNN
Mask out background with mean image

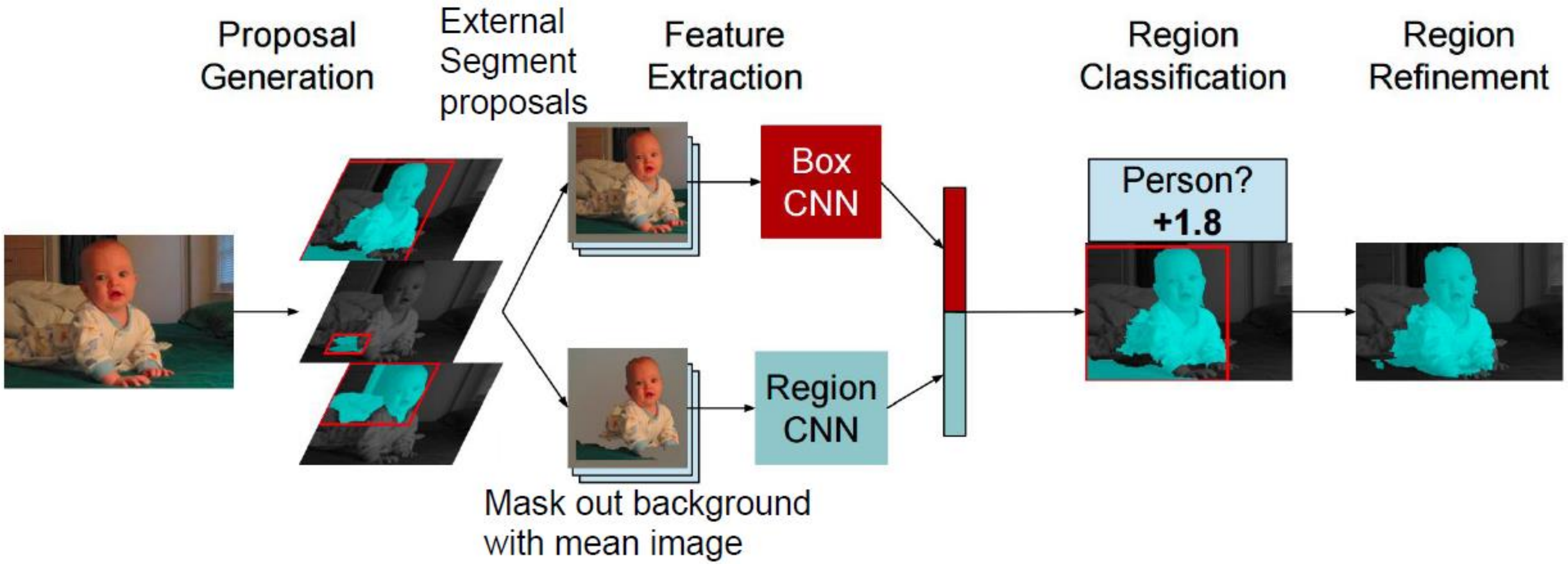
Hariharan et al, “Simultaneous Detection and Segmentation”, ECCV 2014

# Instance Segmentation



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

# Instance Segmentation



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

# Instance Segmentation: Cascades

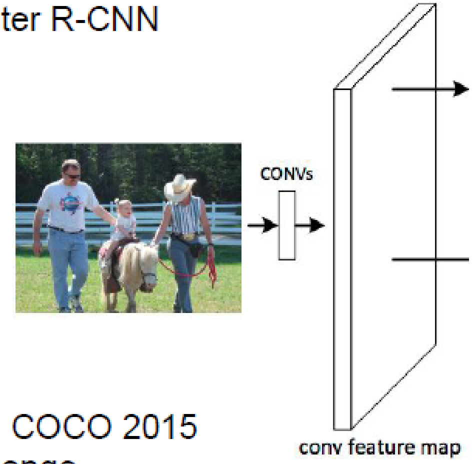
Similar to
Faster R-CNN

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

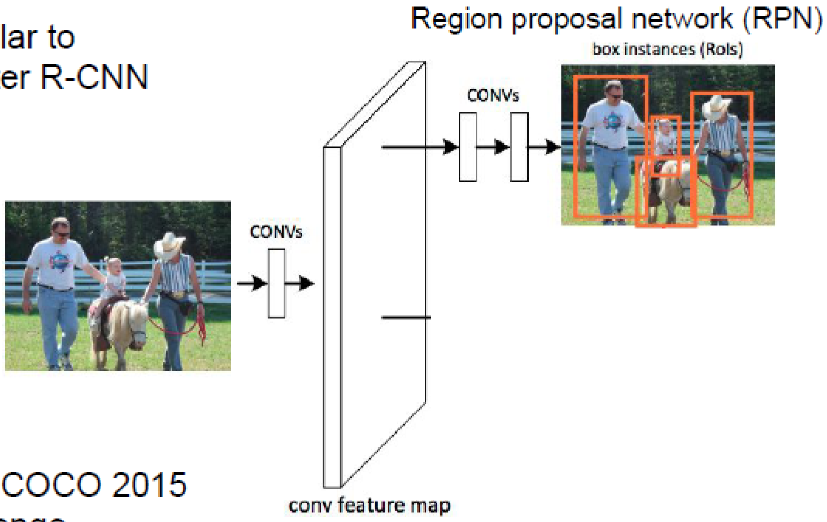


Won COCO 2015 challenge (with ResNet)

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

# Instance Segmentation: Cascades

Similar to
Faster R-CNN

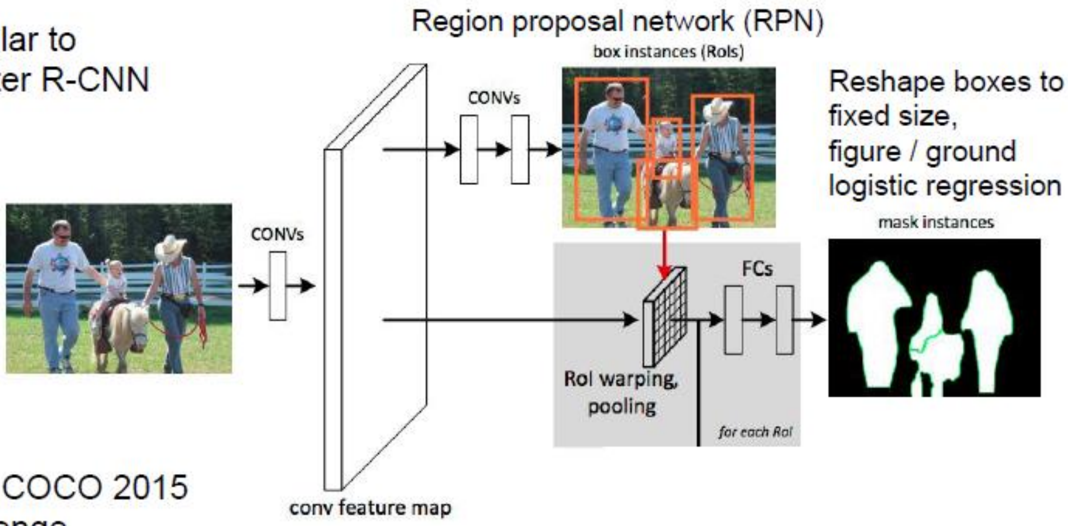


Won COCO 2015
challenge
(with ResNet)

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

# Instance Segmentation: Cascades

Similar to Faster R-CNN



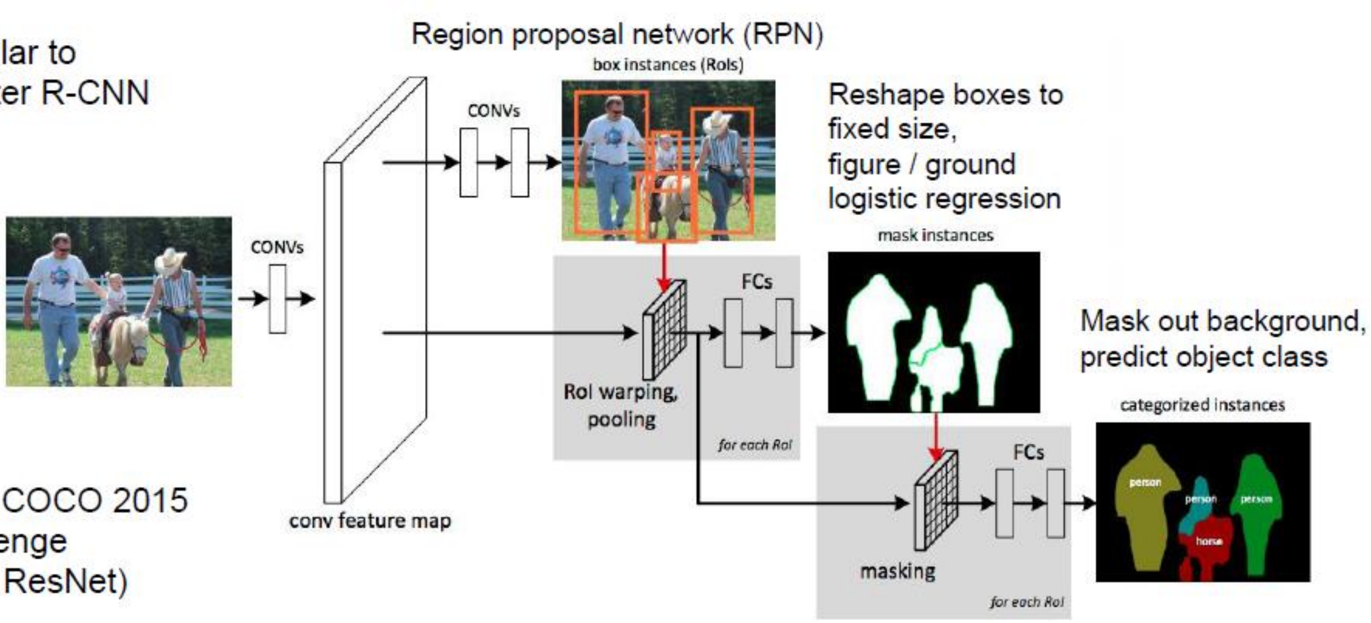
Reshape boxes to fixed size, figure / ground logistic regression

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



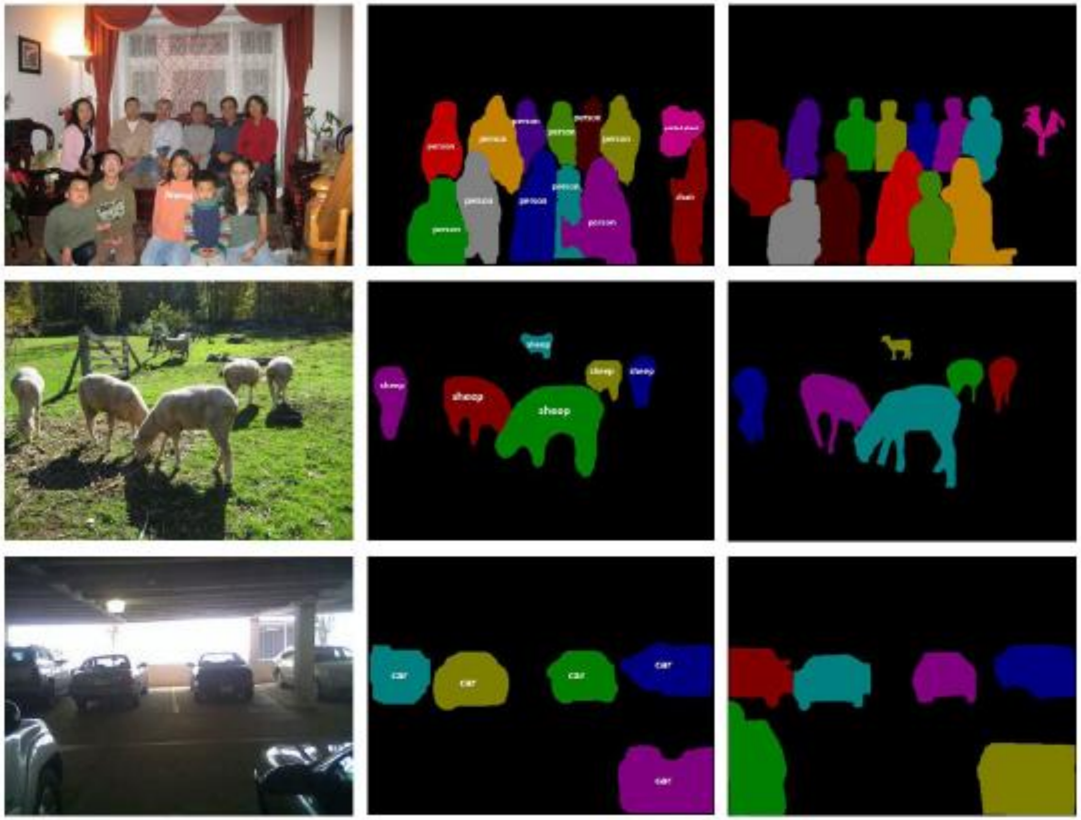
Reshape boxes to fixed size, figure / ground logistic regression

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



Predictions

Ground truth

# Summary

- 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

# BACKUPS

# LEARNABLE UPSAMPLING?

# Learnable upsampling

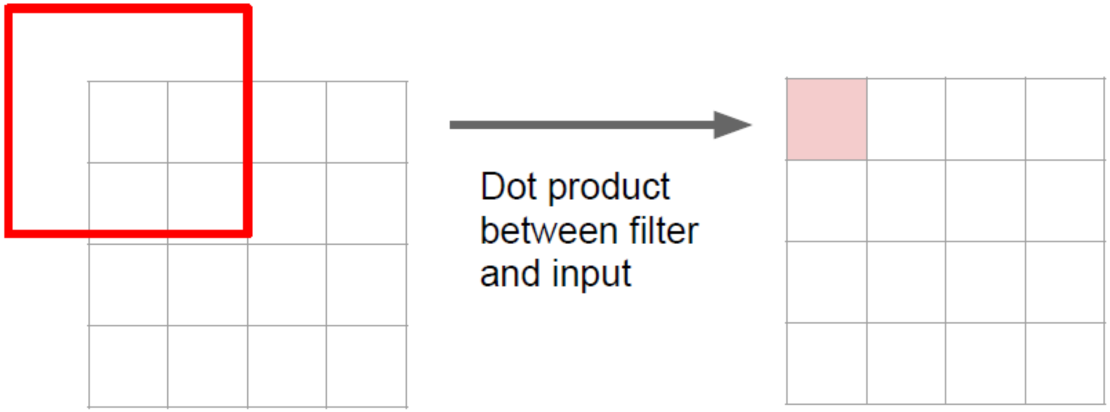
Typical 3 x 3 convolution, stride 1 pad 1

Input: 4 x 4

Output: 4 x 4

# Learnable upsampling

Typical 3 x 3 convolution, stride 1 pad 1



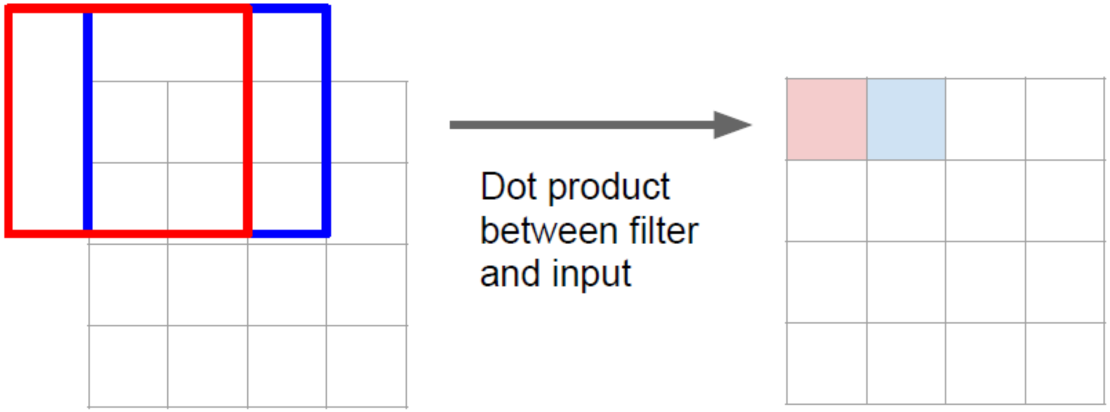
Dot product between filter and input

Input: 4 x 4

Output: 4 x 4

# Learnable upsampling

Typical 3 x 3 convolution, stride 1 pad 1



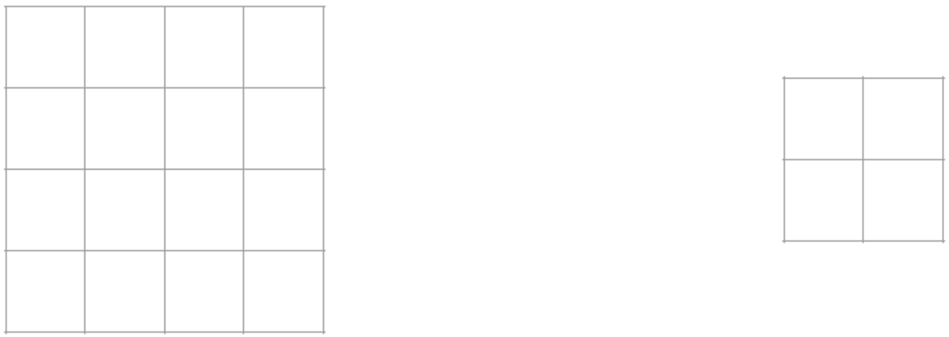
Dot product between filter and input

Input: 4 x 4

Output: 4 x 4

# Learnable upsampling

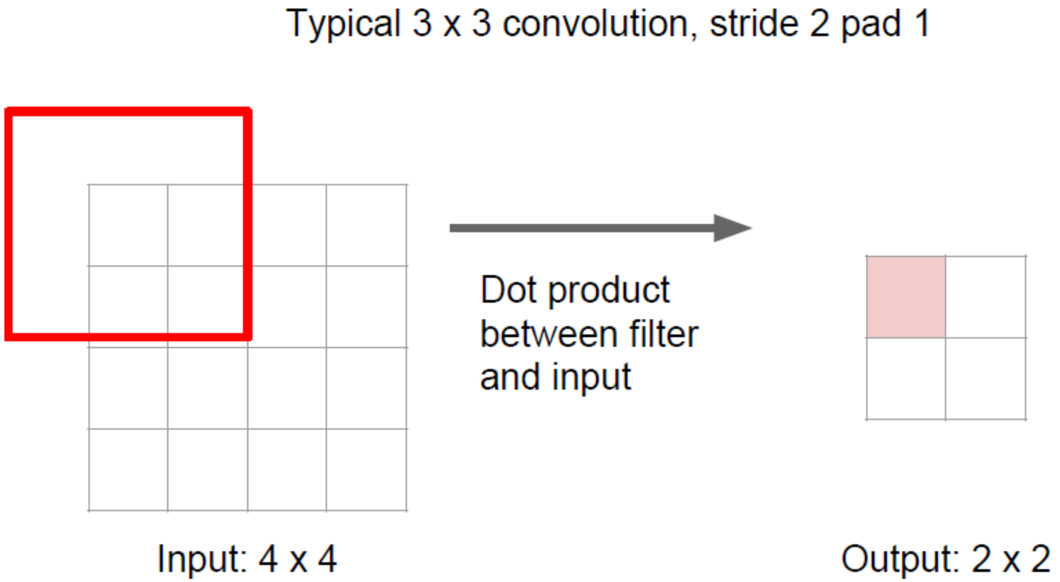
Typical 3 x 3 convolution, **stride 2** pad 1



Input: 4 x 4

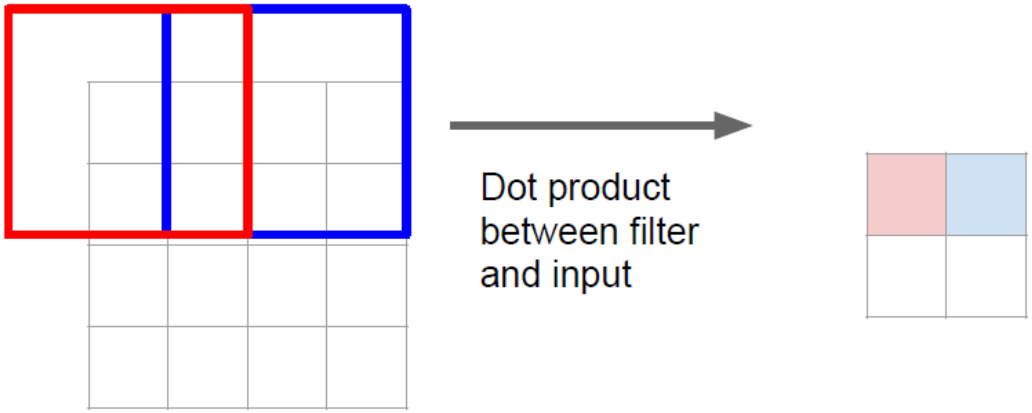
Output: 2 x 2

# Learnable upsampling


Typical 3 x 3 convolution, stride 2 pad 1
Dot product between filter and input
Input: 4 x 4
Output: 2 x 2

# Learnable upsampling

Typical 3 x 3 convolution, stride 2 pad 1



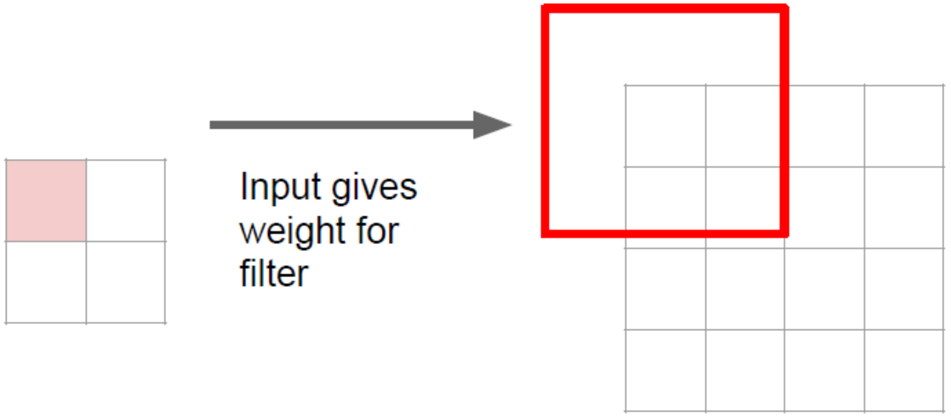
Dot product between filter and input

Input: 4 x 4

Output: 2 x 2

# Learnable upsampling

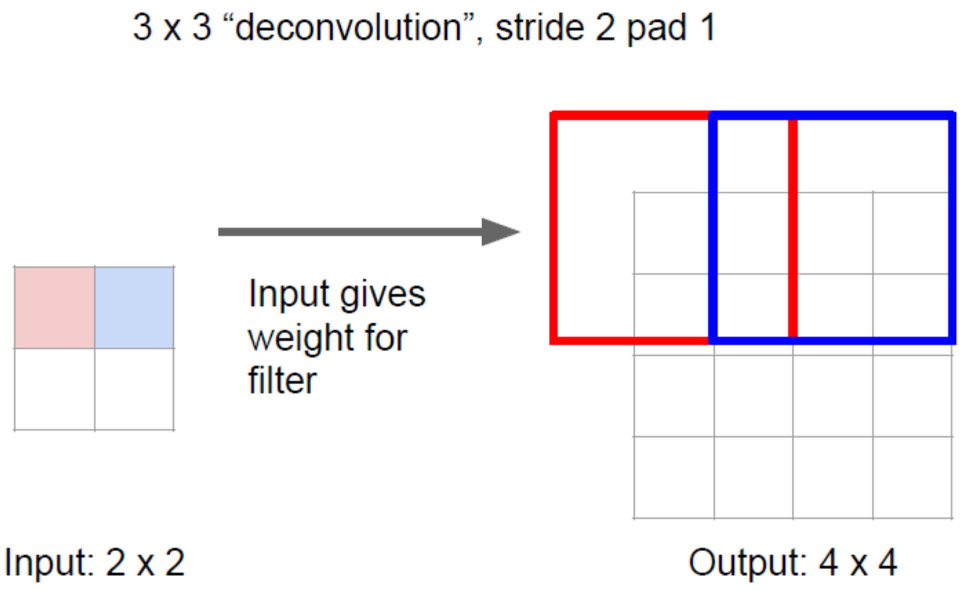
3 x 3 "deconvolution", stride 2 pad 1



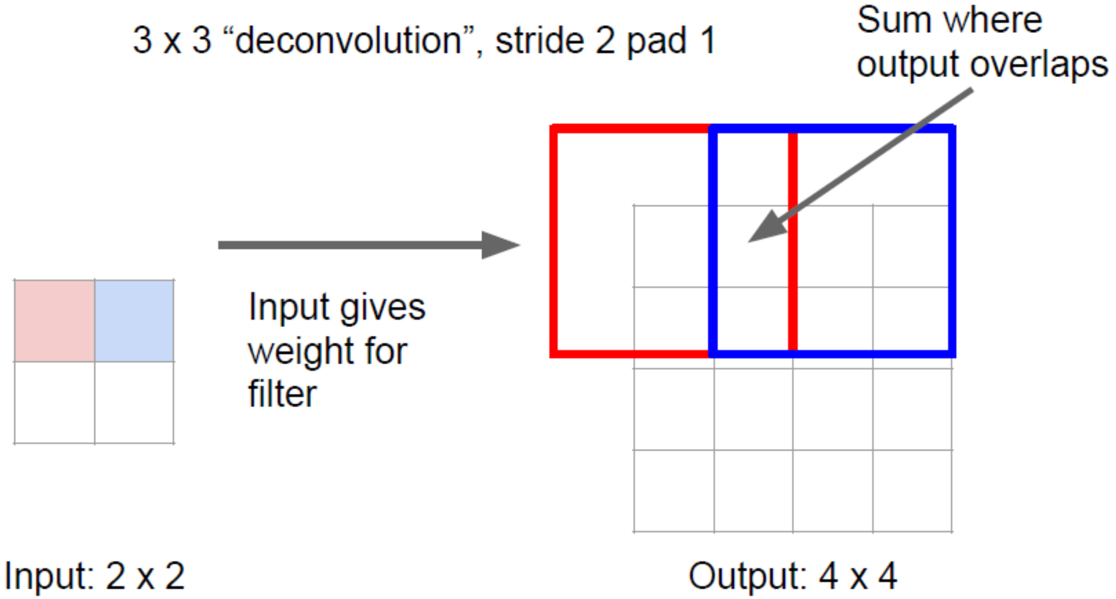
Input: 2 x 2

Output: 4 x 4

# Learnable upsampling


3 x 3 "deconvolution", stride 2 pad 1
Input gives weight for filter
Input: 2 x 2
Output: 4 x 4

# Learnable upsampling


3 x 3 “deconvolution”, stride 2 pad 1
Sum where output overlaps
Input gives weight for filter
Input: 2 x 2
Output: 4 x 4