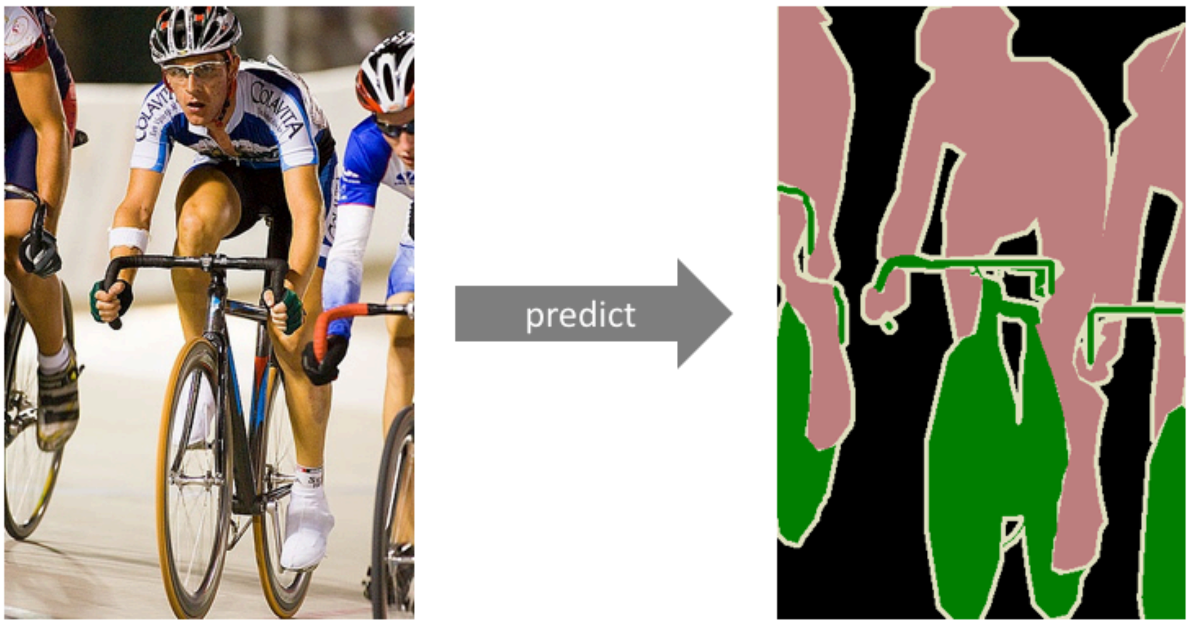
Semantic Segmentation

ISTD 50.035

Computer Vision

Acknowledgement: Some images are from various sources: UCF, Stanford cs231n, etc.

Semantic Segmentation



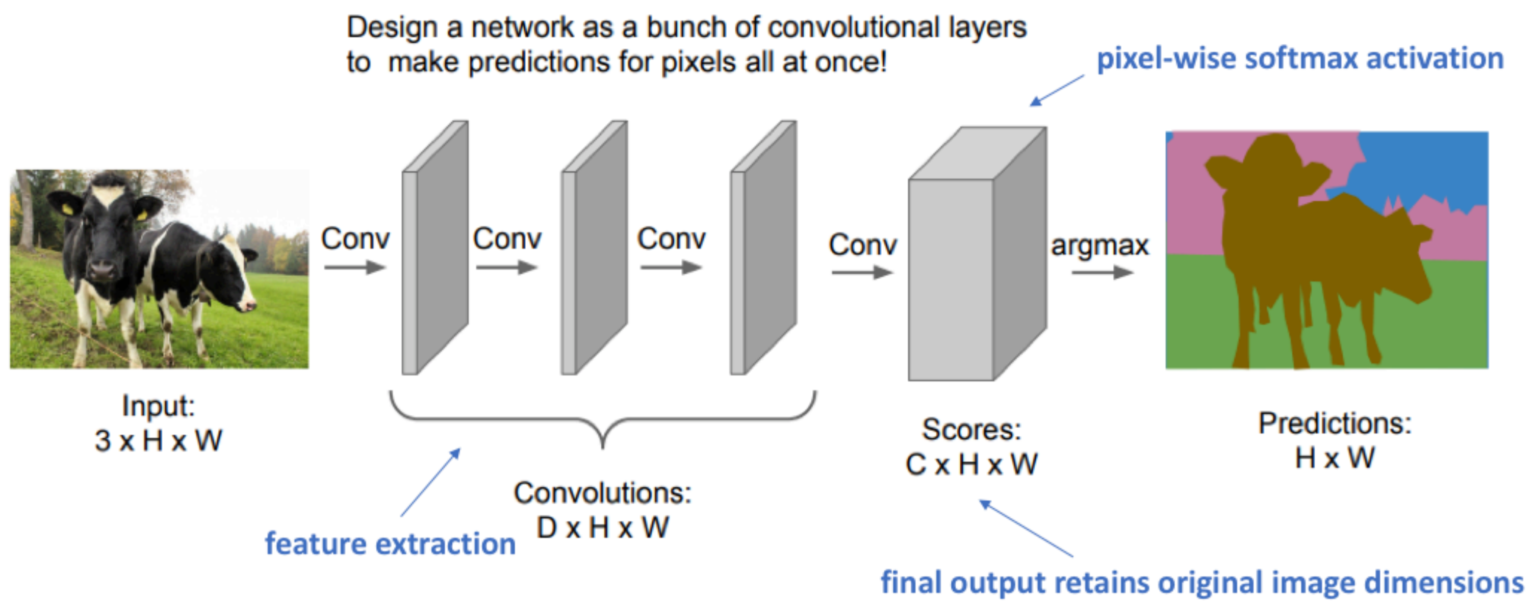
Person
Bicycle
Background

Label each pixel of an image with a class value -> dense prediction

Semantic Segmentation

Input Image | Semantic Segmentation | Instance Segmentation

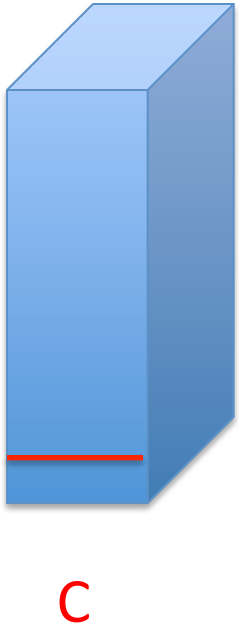
Semantic Segmentation with ConvNet



Downside: Preserving image dimensions throughout entire network will be computationally expensive.

Probability vector of C classes at each pixel location

Pixel wise softmax loss

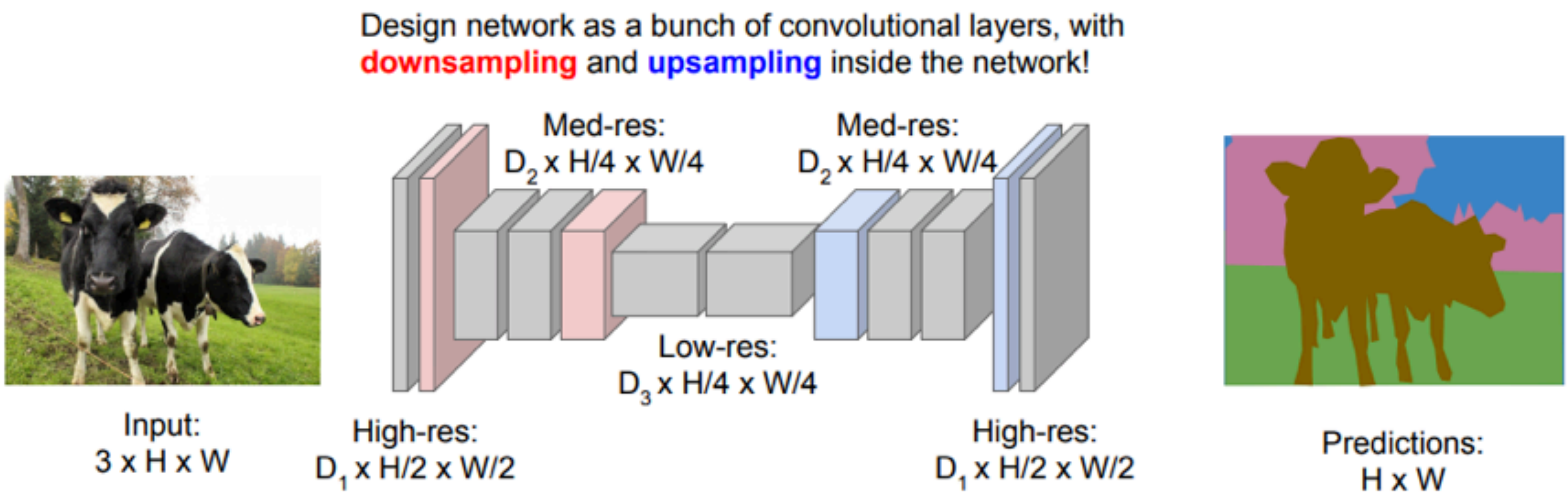


Probability vector of C classes at each pixel location

Loss at each pixel location = -log p_y

Loss = (spatial) sum of loss at each pixel

Semantic Segmentation with ConvNet

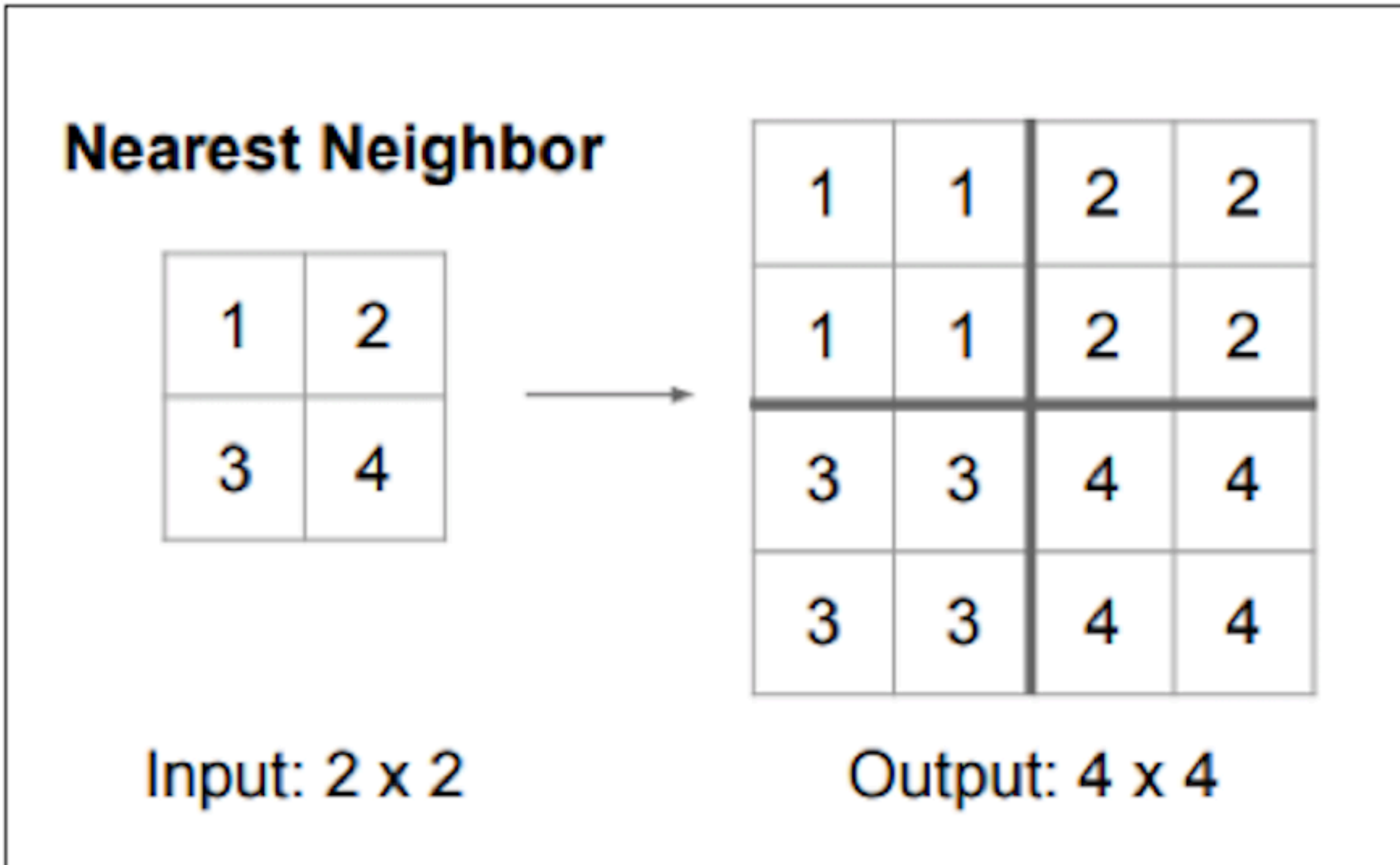


Solution: Make network deep and *work at a lower spatial resolution* for many of the layers.

Probability vector of C classes at each pixel location

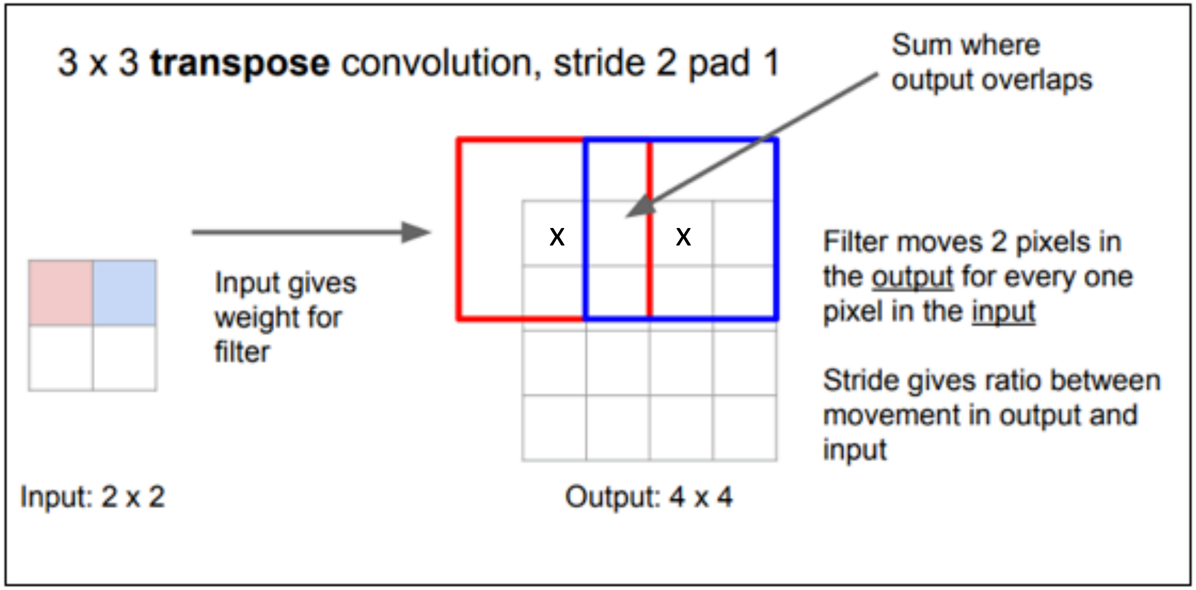
Upsampling

Unpooling: inverse of max/average pooling



Upsampling

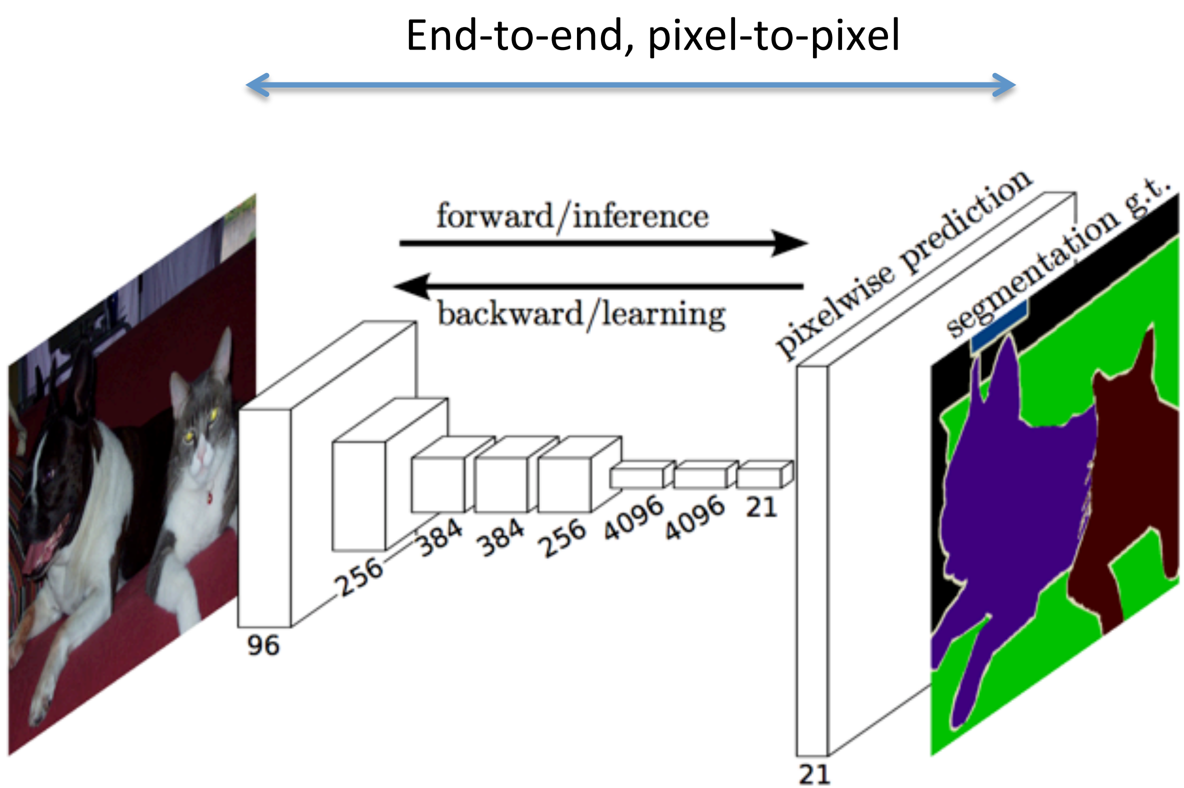
Transpose convolution



Standard convolution: input (matrix) * filter (matrix) -> output (scalar)
Transpose convolution: input (scalar) * filter (matrix) -> output (matrix)

Learn filter mask for optimal upsampling

Fully convolutional network (FCN)



[Long et al. 2014]

-Well pre-trained network as 'encoder'

-Transpose convolution layers to upsample the coarse feature map to full-resolution segmentation map

-Trained end-to-end, pixel-to-pixel

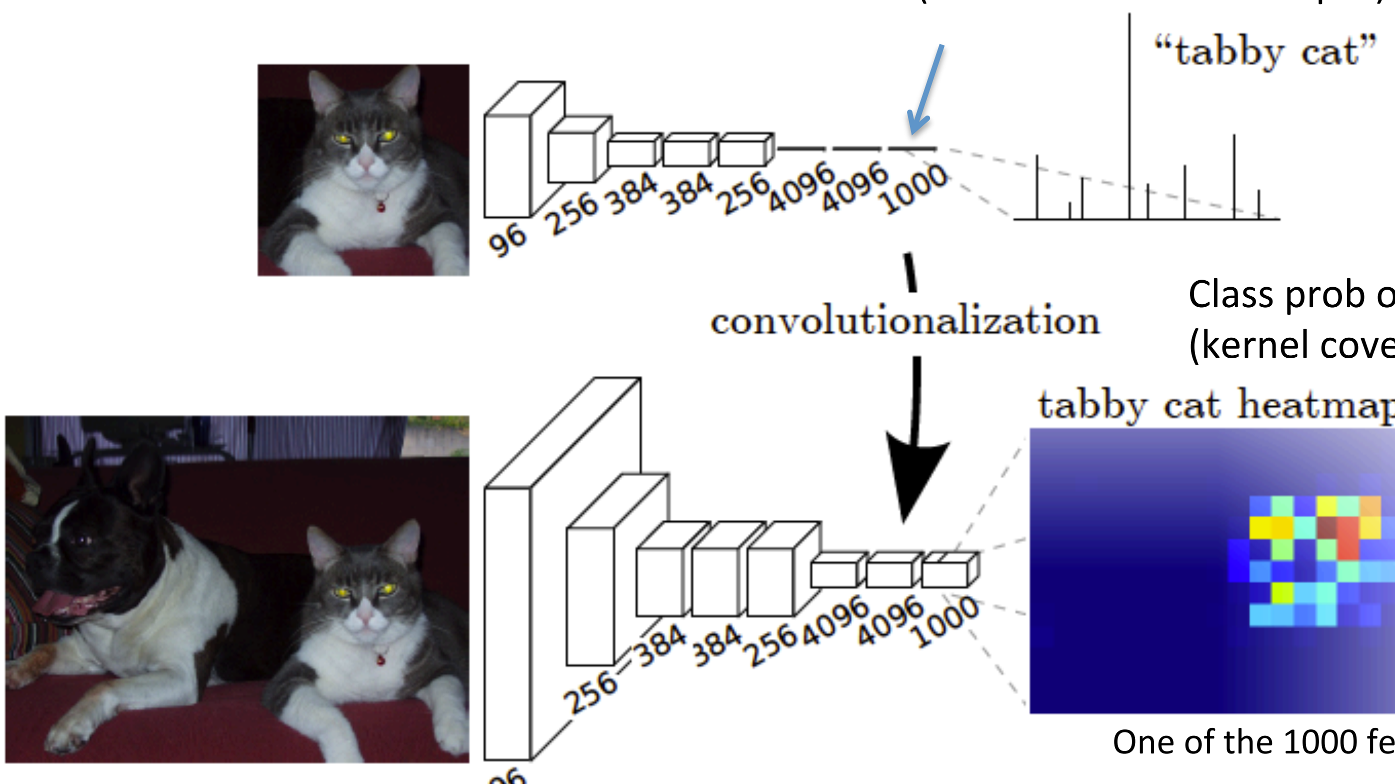
Fully convolutional network (FCN)

Fully connected layer

-Discard spatial information -> not suitable for SS

-Can be viewed as convolutions with kernels that cover the entire input region (so no spatial info)

Class prob of whole image
(kernel covers entire input)

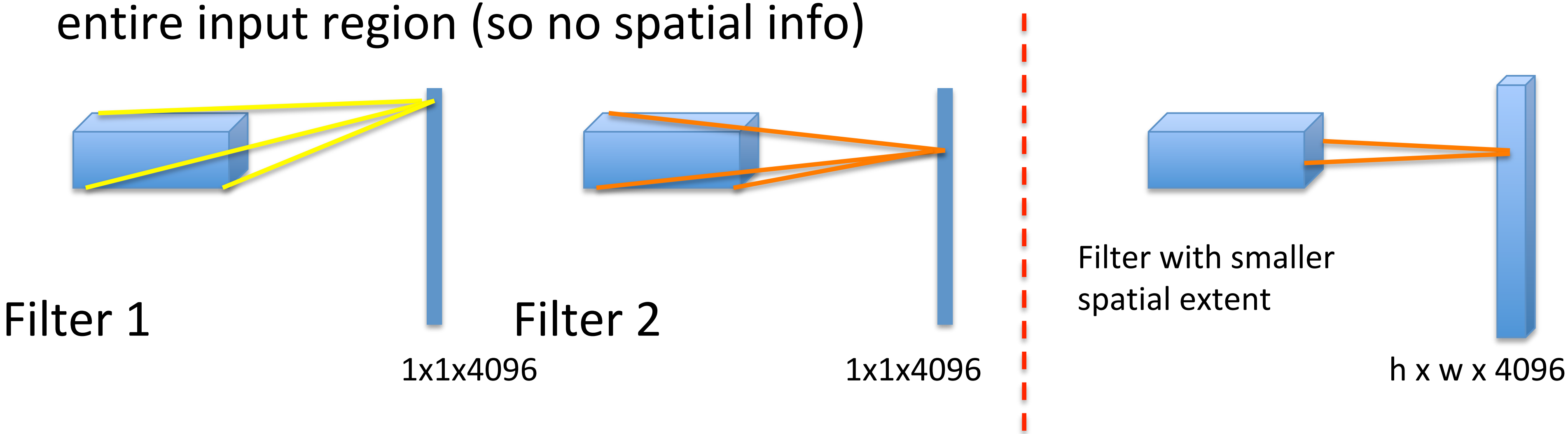


Class prob of different areas
(kernel covers a local area)

One of the 1000 feature maps

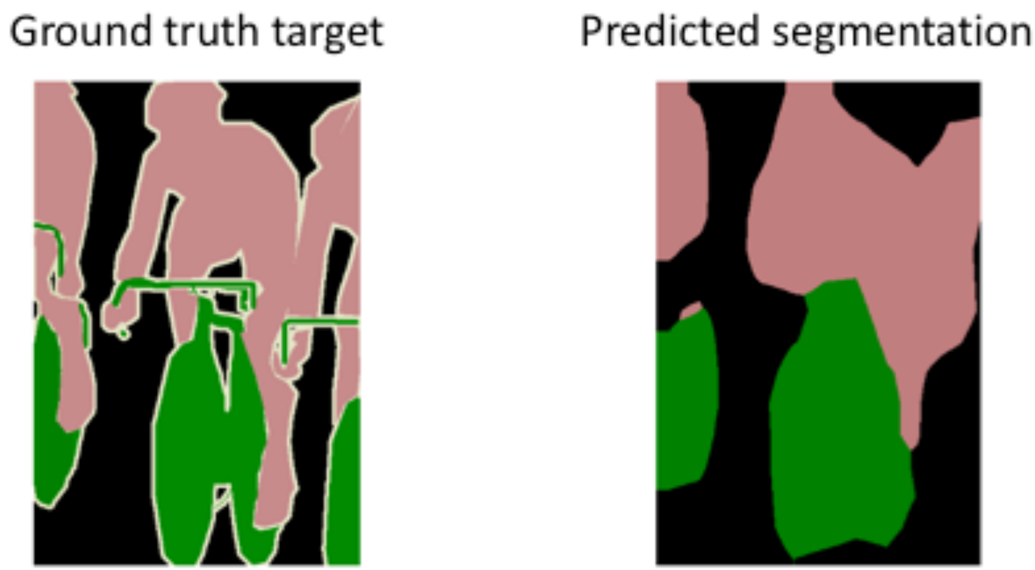
Fully convolutional network (FCN)

Fully connected layer
-Discard spatial information -> not suitable for SS
-Can be viewed as convolutions with kernels that cover the entire input region (so no spatial info)



Fully connected layer: 4096 filter masks; filter size = input
Output of fully connected layer: 1x1 feature map (total 4096 channels)
Fully convolutional: use small filter

Issue in baseline FCN



"Semantic segmentation faces an inherent tension between semantics and location: global information resolves what while local information resolves where"

"Combining fine layers and coarse layers lets the model make local predictions that respect global structure"

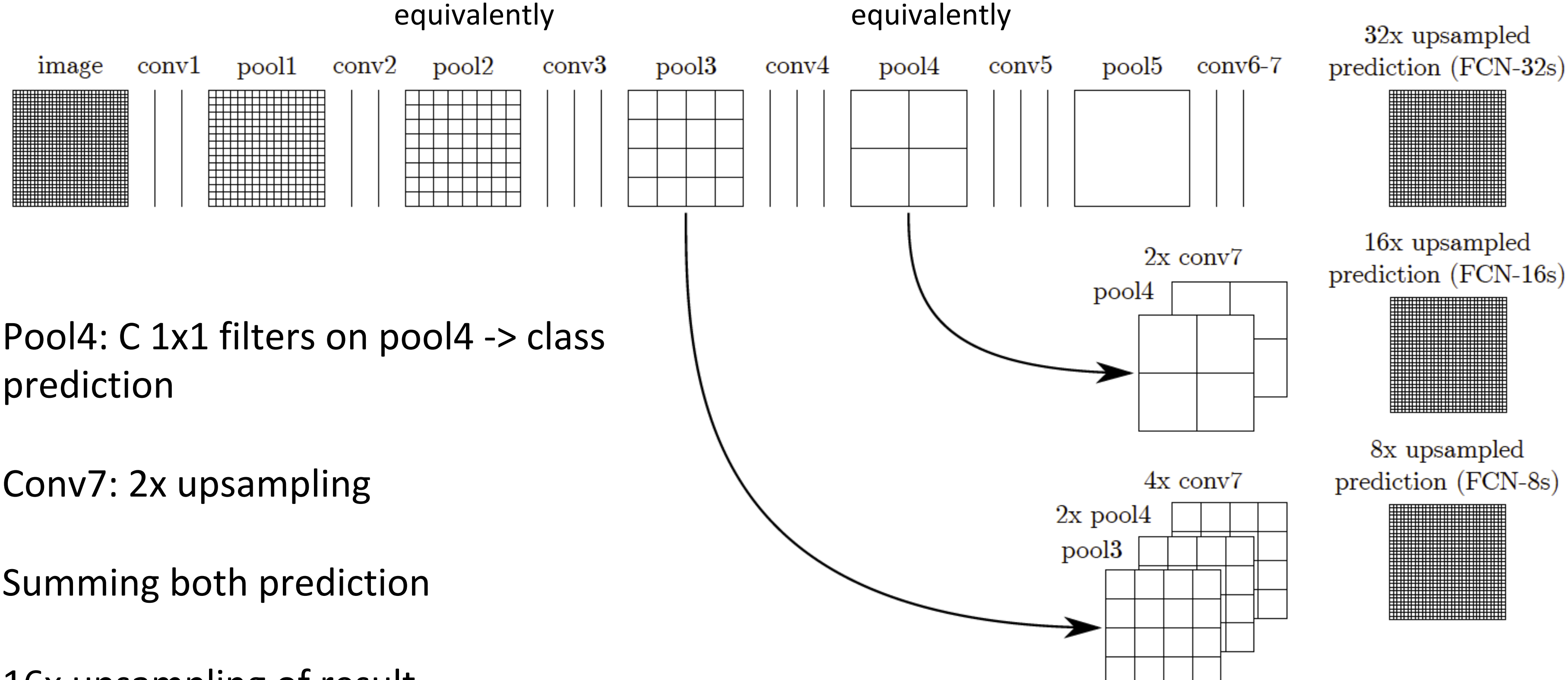
Combining what and where

- Deep, coarse semantic information
- Shallow, fine appearance information

Pooling with stride 2

Pooling with stride 4 equivalently

Pooling with stride 16 equivalently



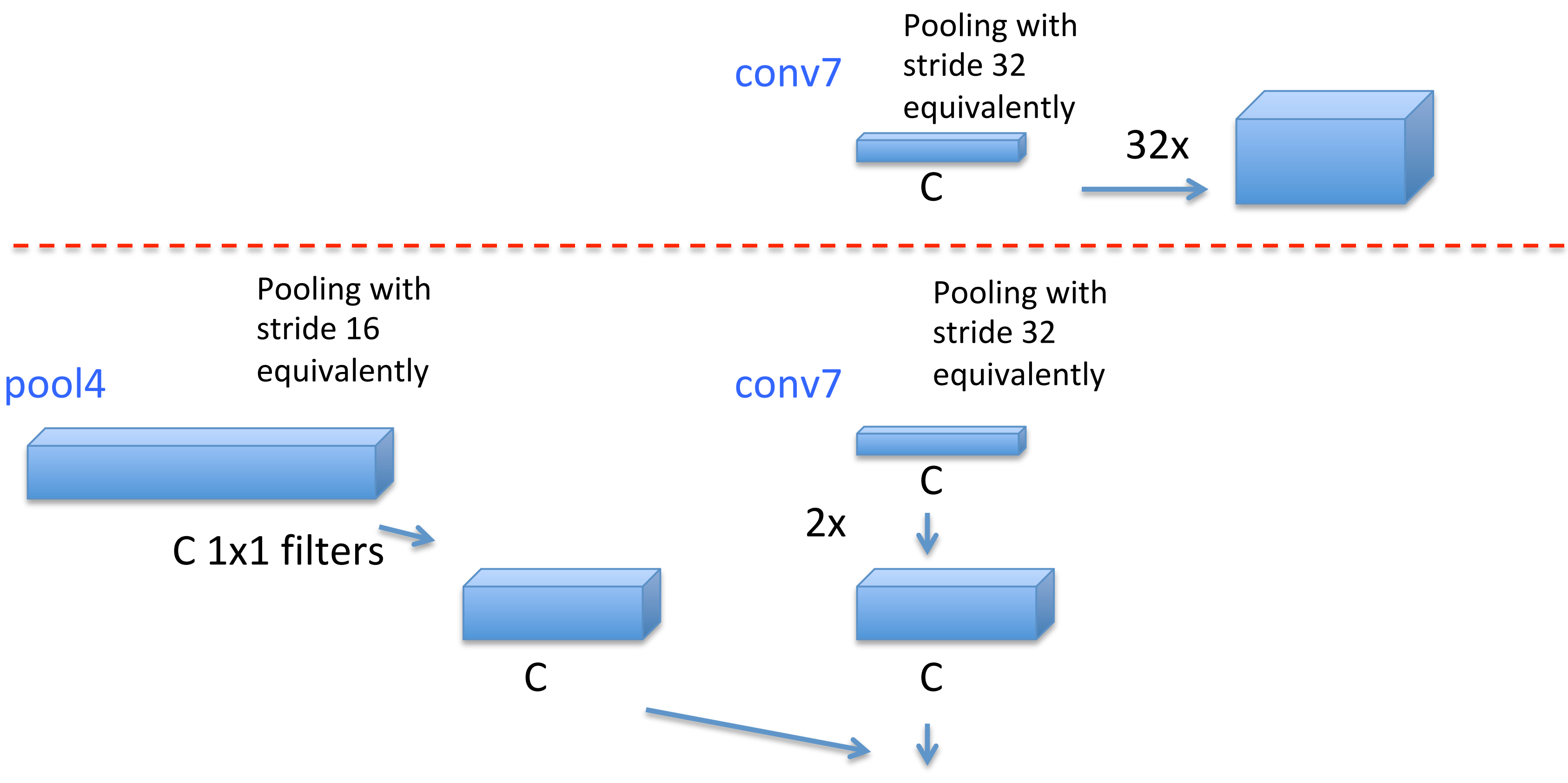
Pool4: C 1x1 filters on pool4 -> class prediction

Conv7: 2x upsampling

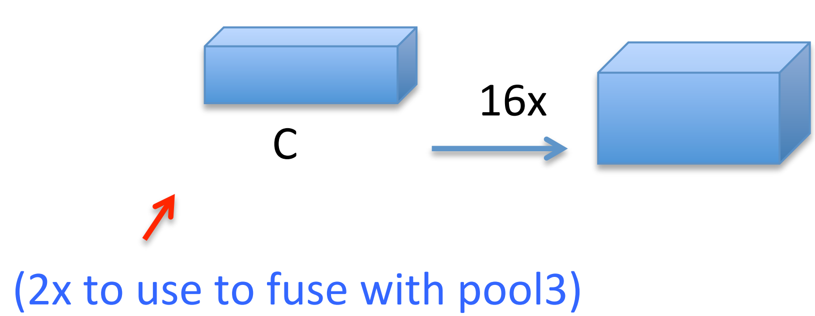
Summing both prediction

16x upsampling of result

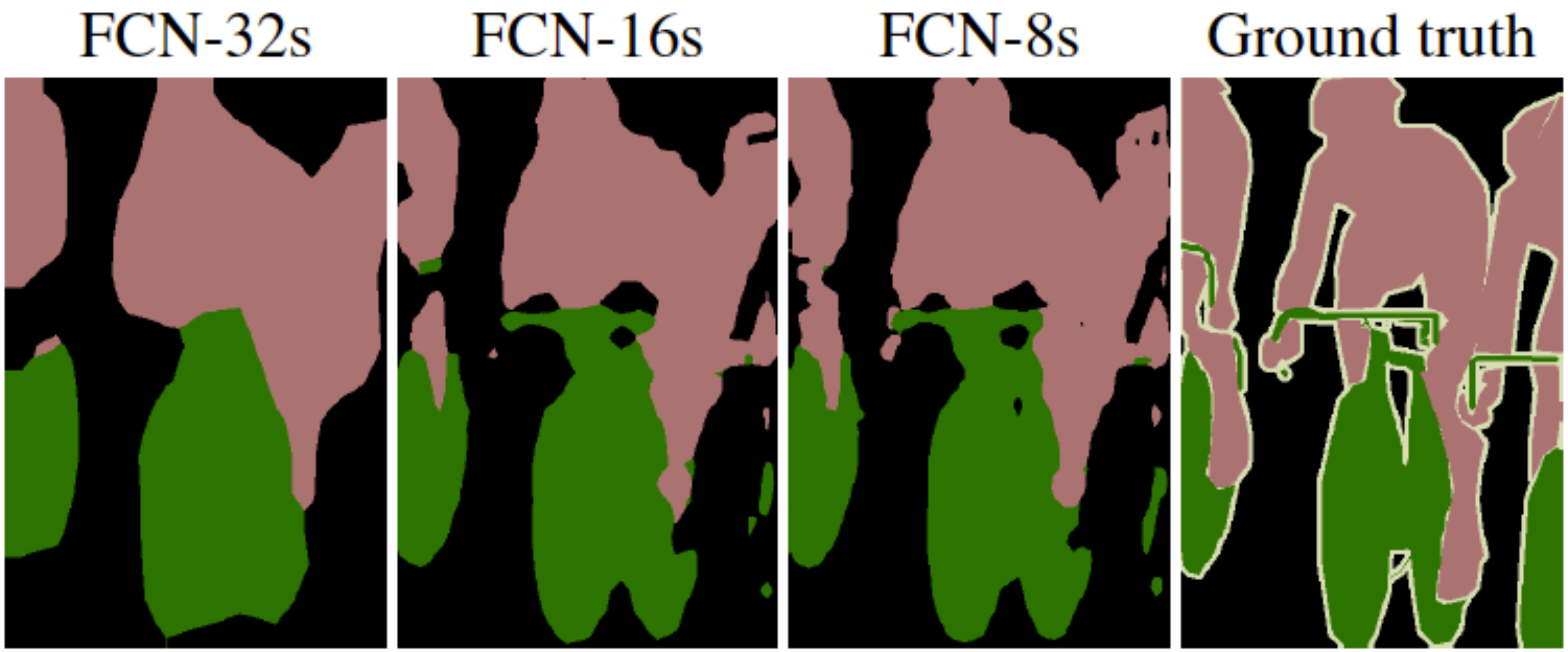
Combining what and where



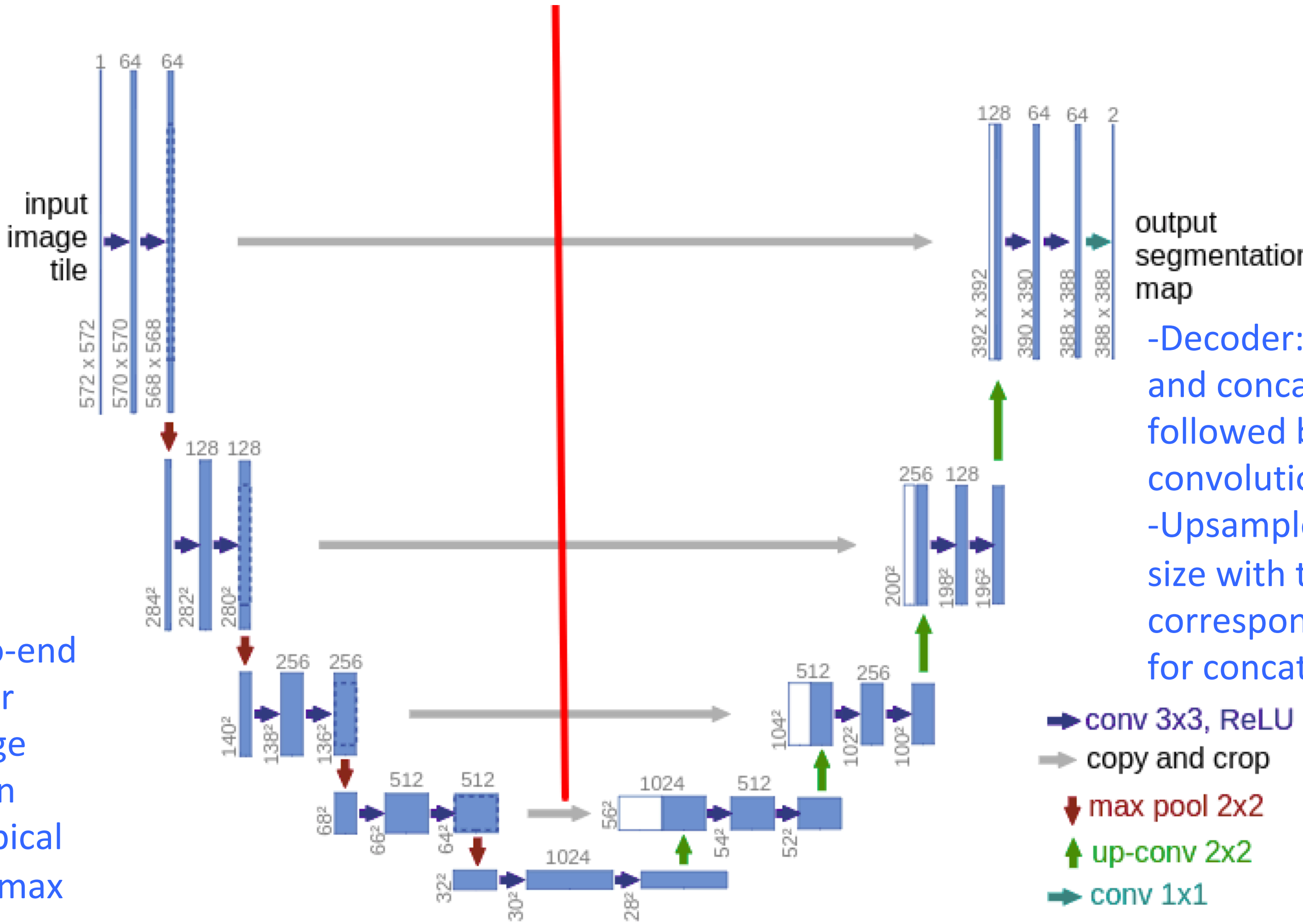
- Pool4: C 1x1 filters on pool4 -> class prediction
- Conv7: 2x upsampling
- Summing both prediction
- 16x upsampling of result



Combining what and where



U-Net: high capacity decoder



-Decoder: upsample and concatenation, followed by regular convolution

-Upsample: same size with the corresponding block for concatenation

-Train end-to-end

-Originally for medical image segmentation

-Encoder: typical conv layer + max pool

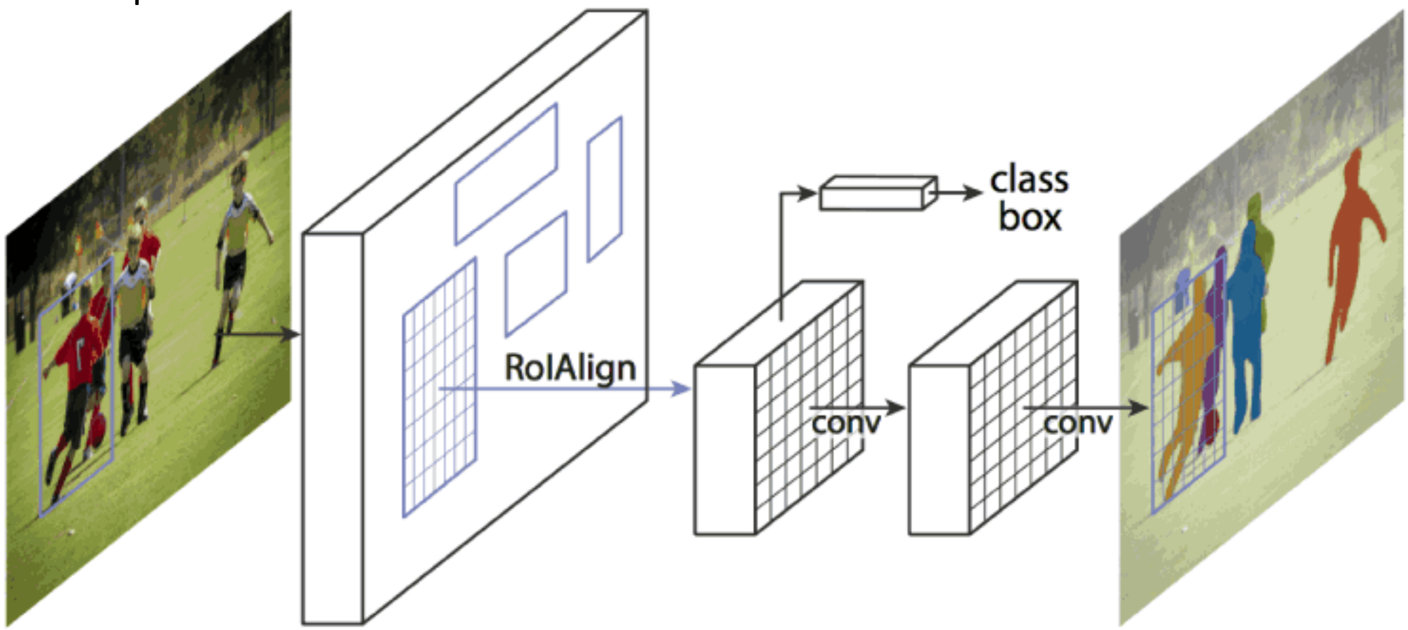
U-Net

- Upsampling: not accurate

- Use earlier stages to provide representation for localization: via concatenation

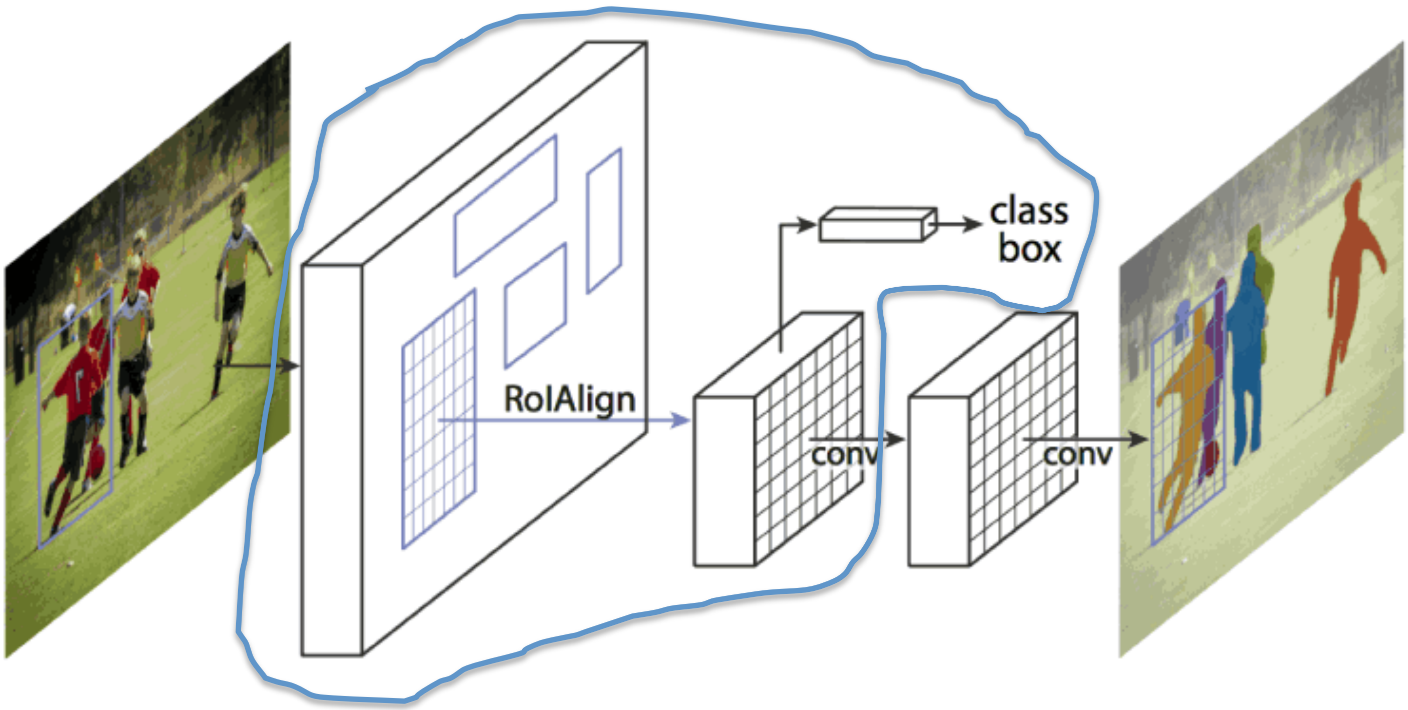
Mask R-CNN for instance segmentation

Mask R-CNN = Faster R-CNN + FCN
Add mask prediction in addition to class and box



The Mask R-CNN framework for instance segmentation

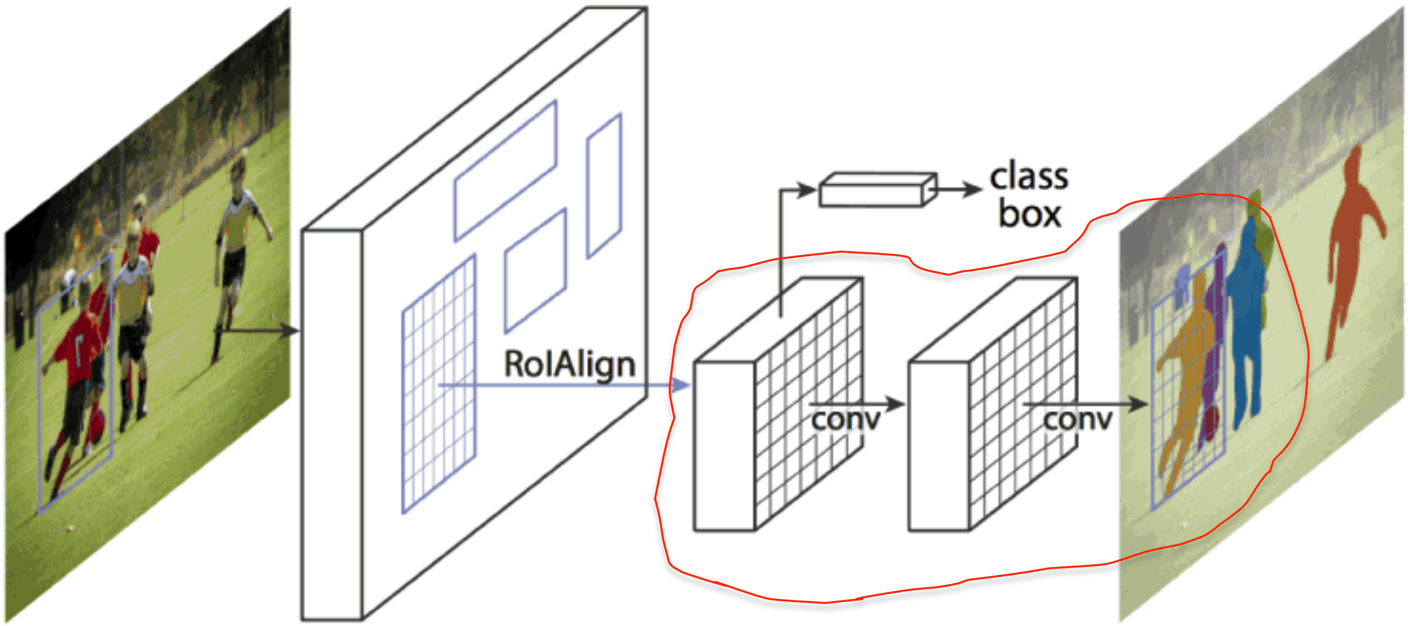
Mask R-CNN for instance segmentation



The Mask R-CNN framework for instance segmentation

Mask R-CNN for instance segmentation

Predict mxm mask from each ROI
Mask: encode input object's spatial layout



The Mask R-CNN framework for instance segmentation