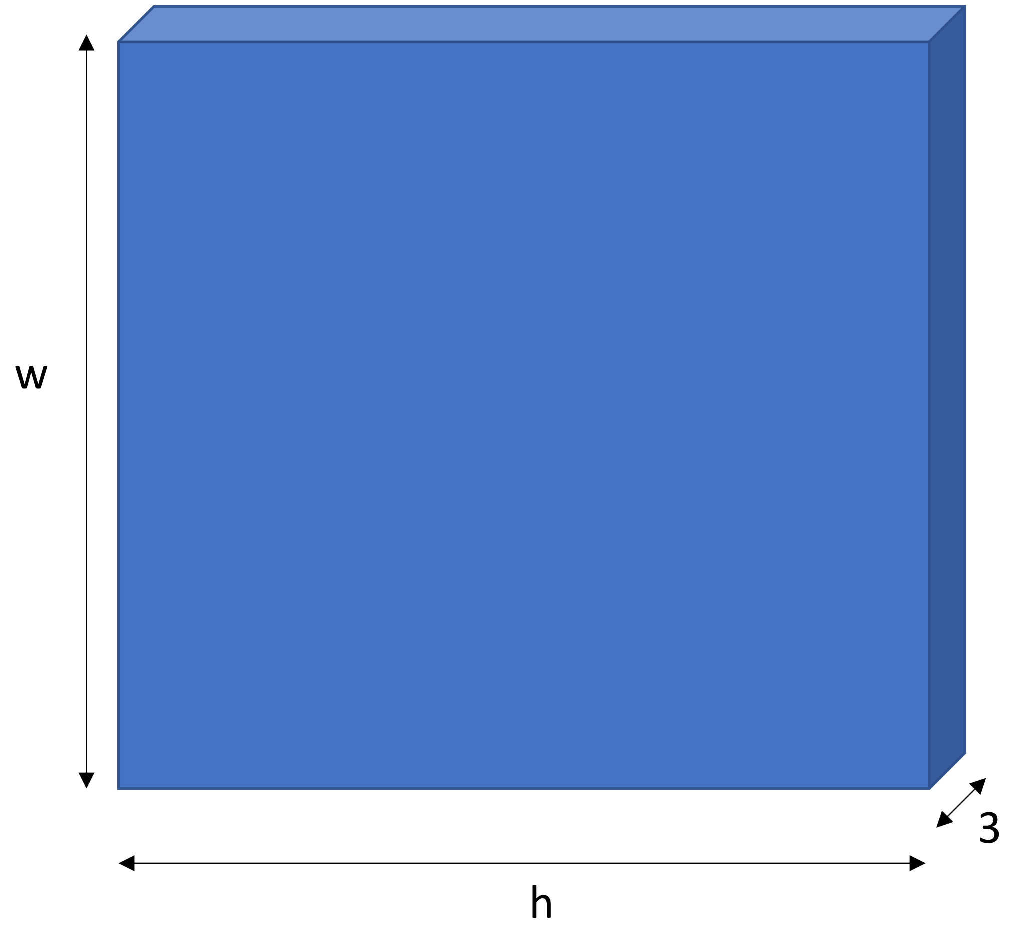
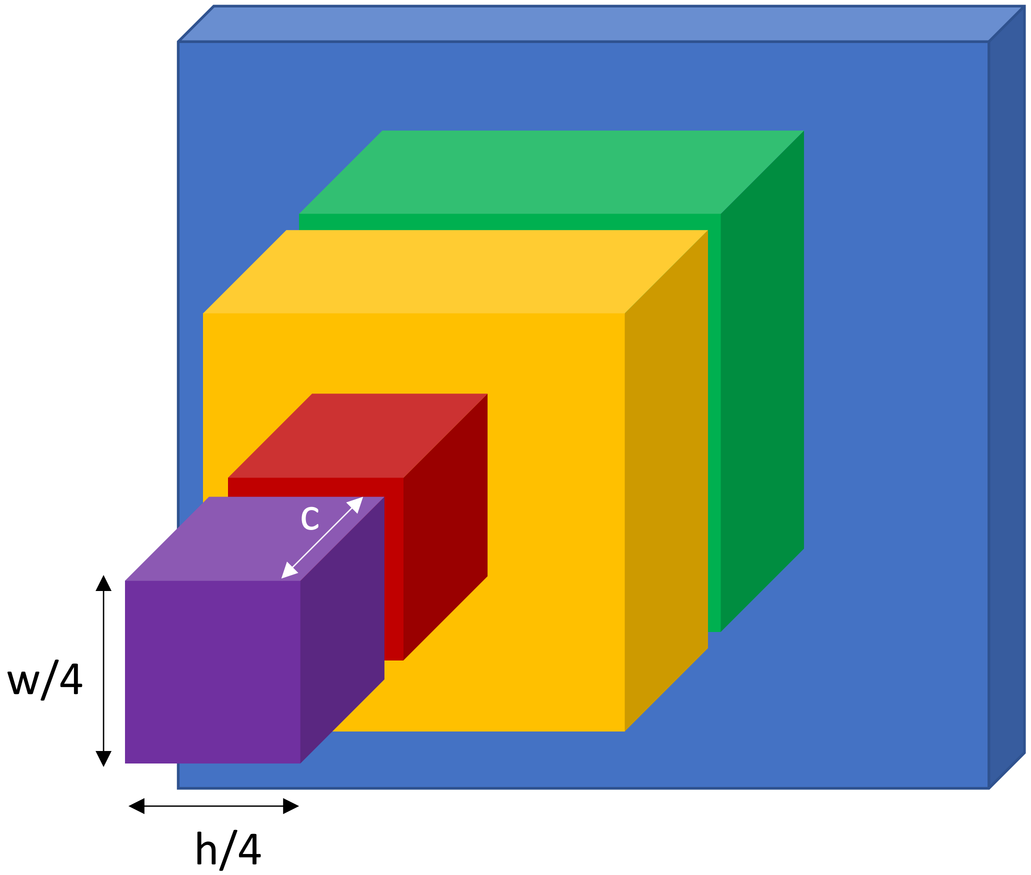
Semantic segmentation

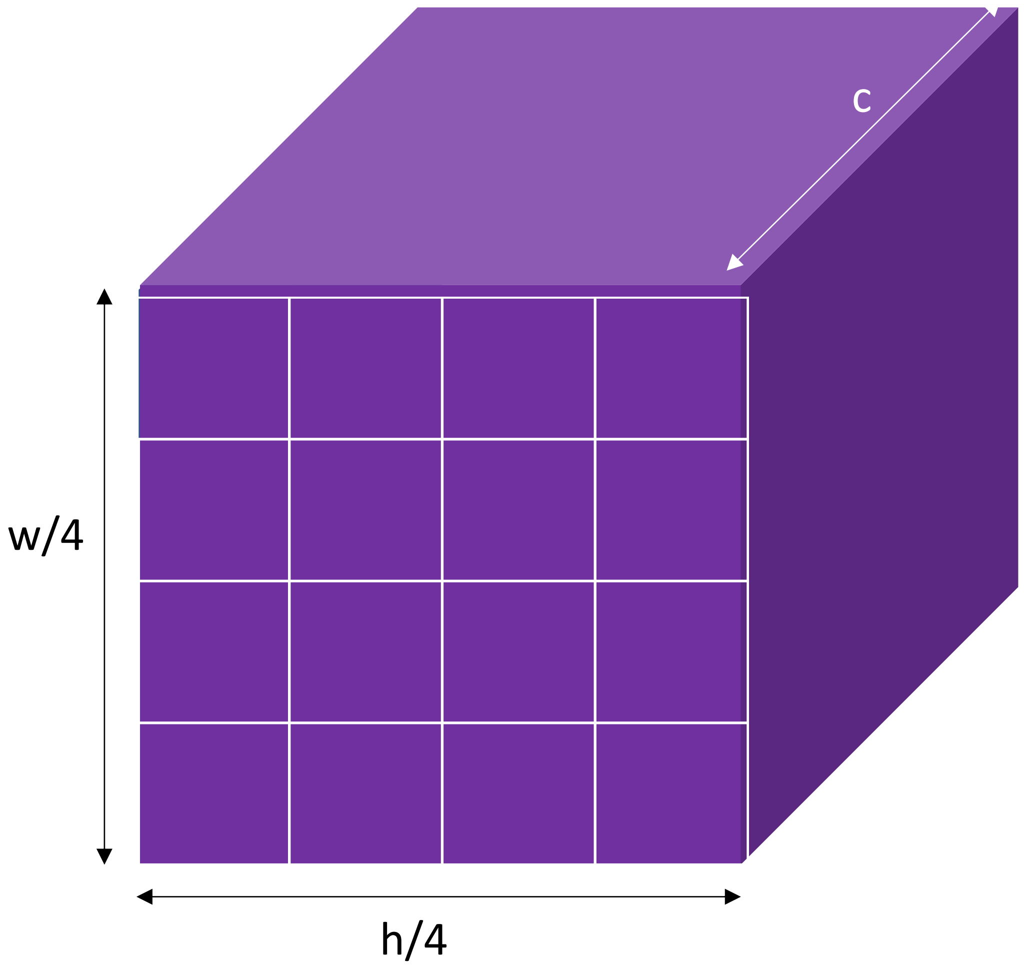
Semantic segmentation using convolutional networks



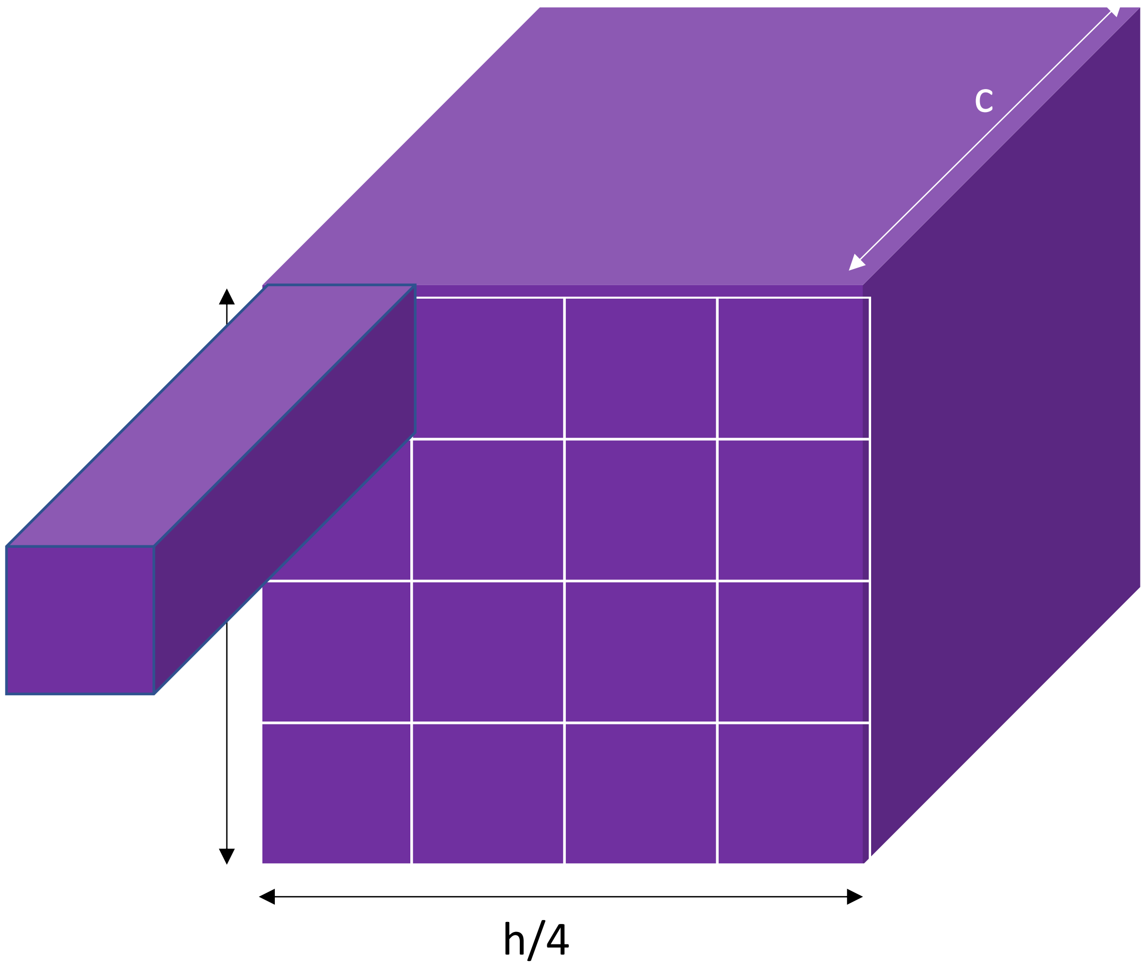
Semantic segmentation using convolutional networks



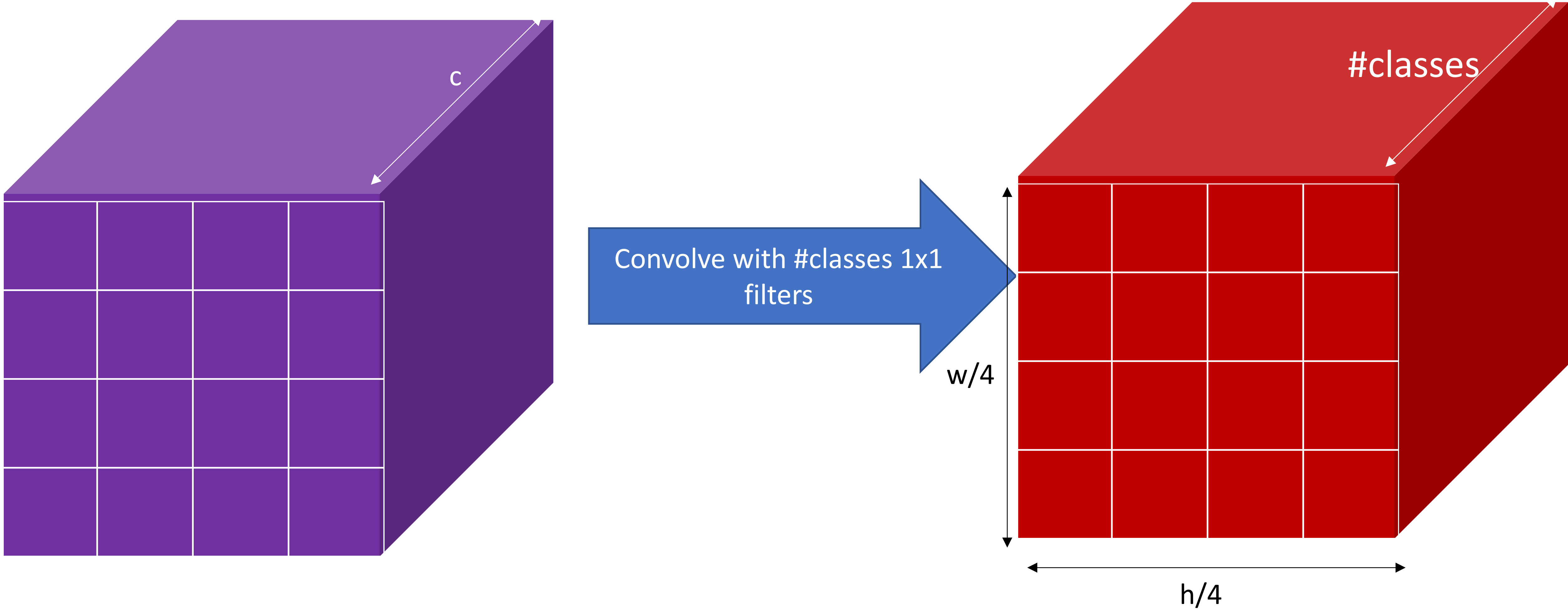
Semantic segmentation using convolutional networks



Semantic segmentation using convolutional networks



Semantic segmentation using convolutional networks



Semantic segmentation using convolutional networks

- Pass image through convolution and subsampling layers
- Final convolution with #classes outputs
- Get scores for *subsampled* image
- Upsample back to original size

Transfer learning for semantic segmentation

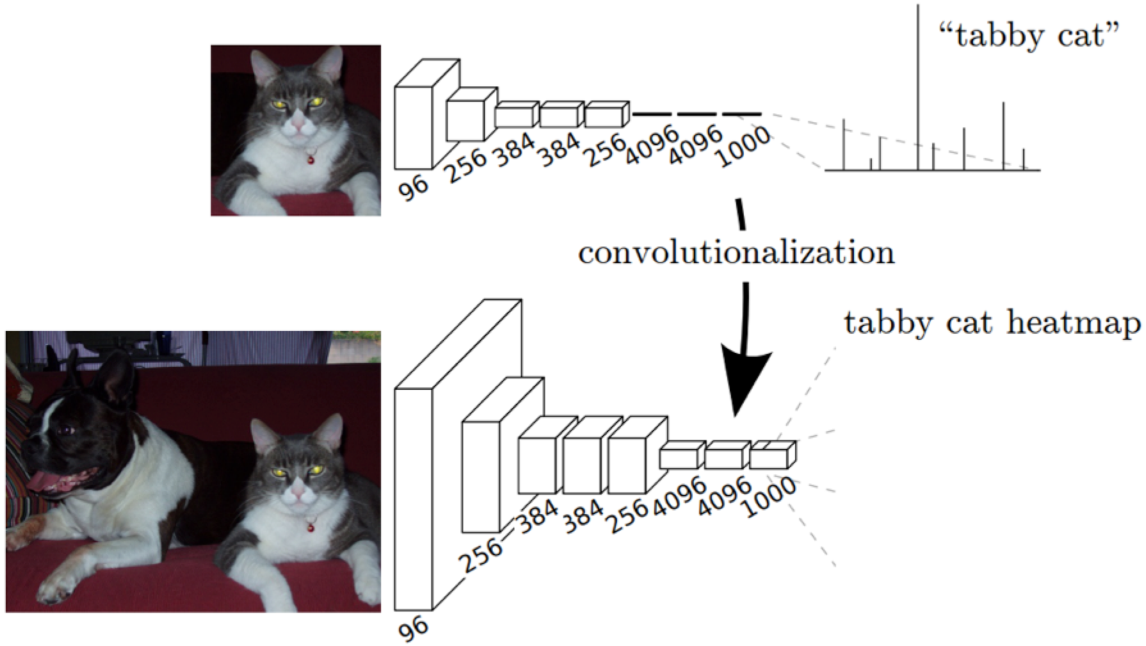


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

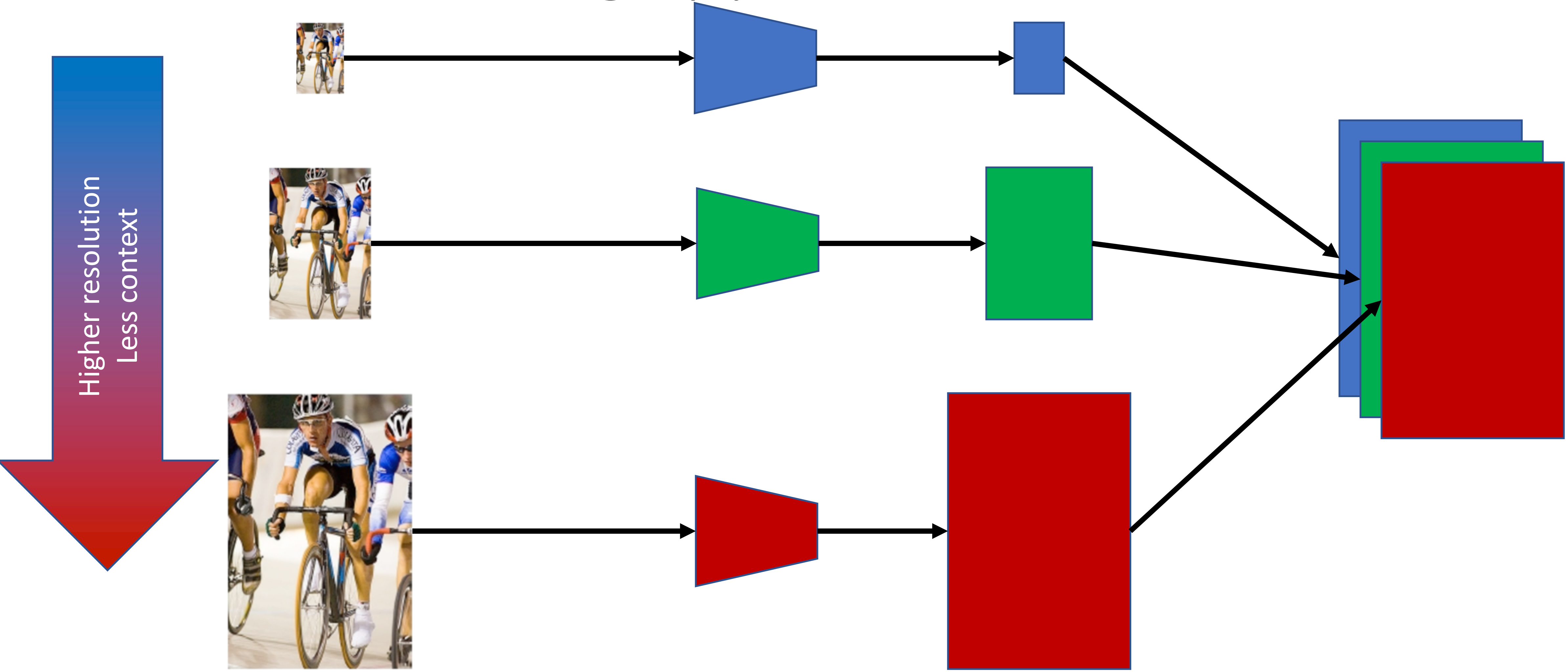
Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

Semantic segmentation using convolutional networks

The resolution issue

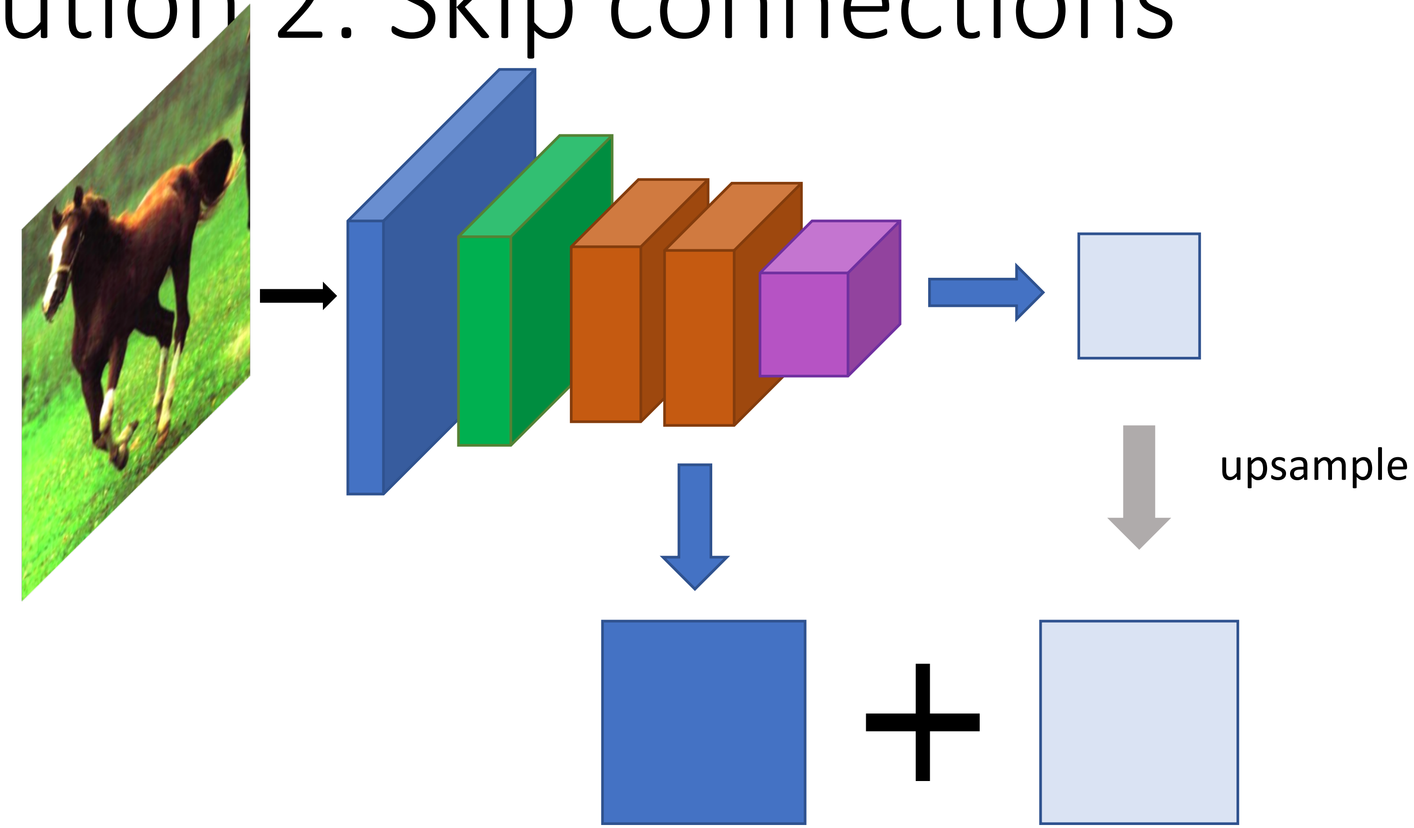
- Problem: Need fine details!
- Shallower network / earlier layers?
  - Not very semantic!
- Remove subsampling?
  - Looks at only a small window!

Solution 1: Image pyramids

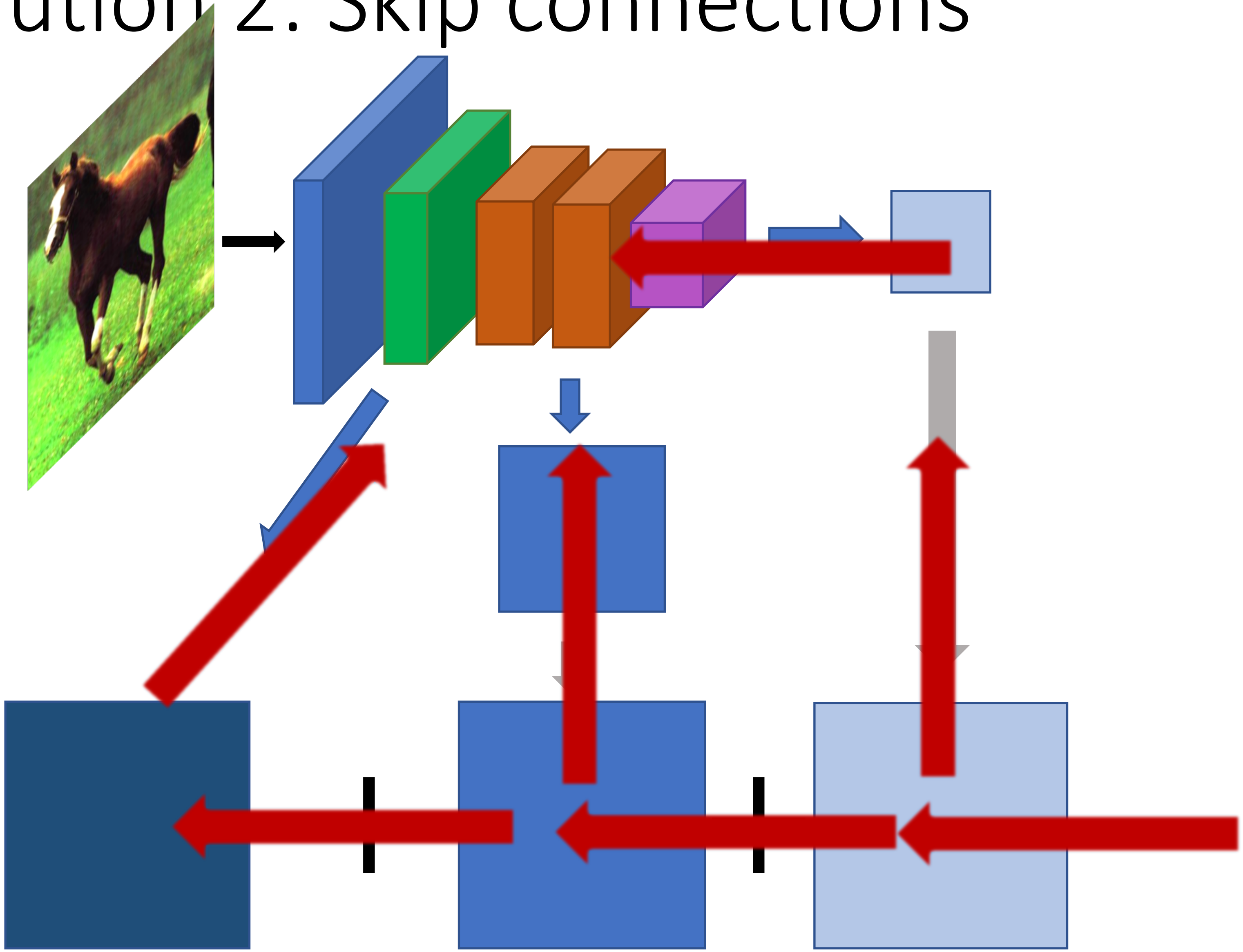


Learning Hierarchical Features for Scene Labeling. Clement Farabet, Camille Couprie, Laurent Najman, Yann LeCun. In *TPAMI,* 2013.

Solution 2: Skip connections



Solution 2: Skip connections

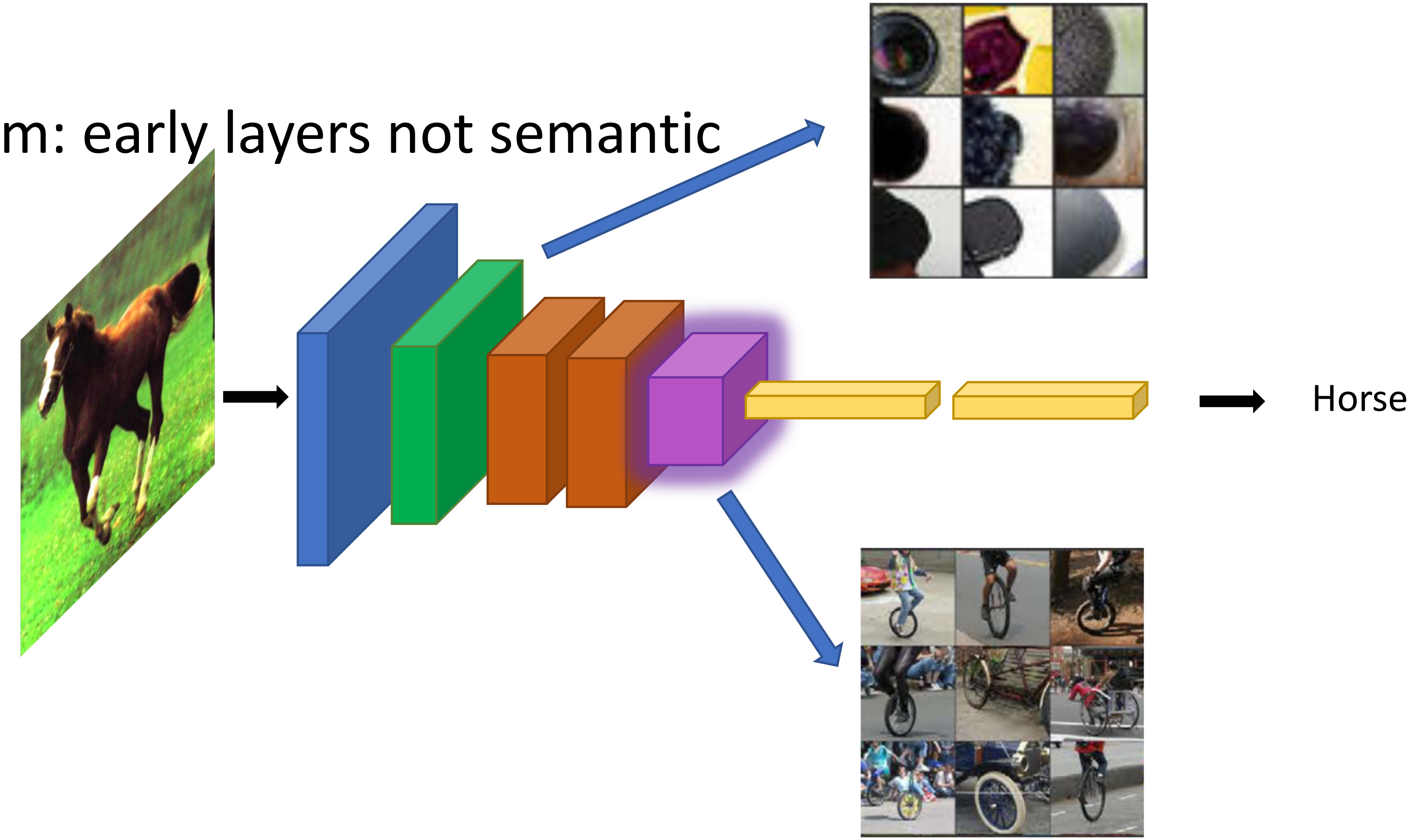


Skip connections

Fully convolutional networks for semantic segmentation. Evan Shelhamer, Jon Long, Trevor Darrell. In *CVPR* 2015

Skip connections

- Problem: early layers not semantic



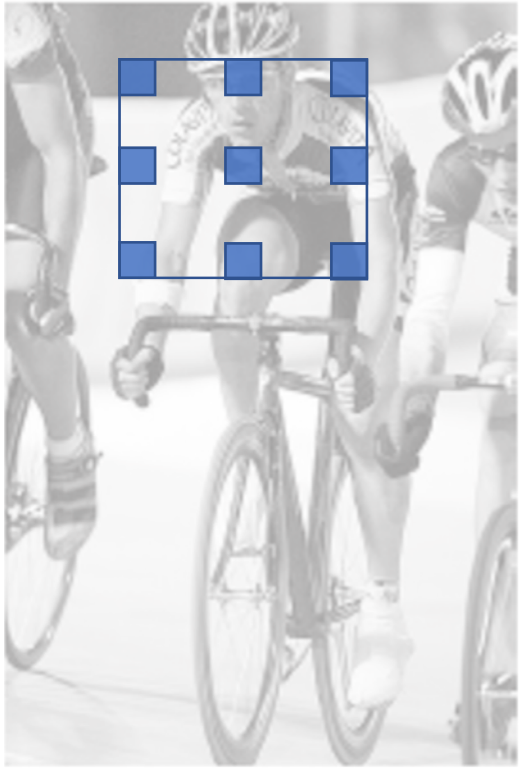
Visualizations from : M. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. In *ECCV* 2014.

Solution 3: Dilation

- Need subsampling to allow convolutional layers to capture large regions with small filters
  - Can we do this without subsampling?

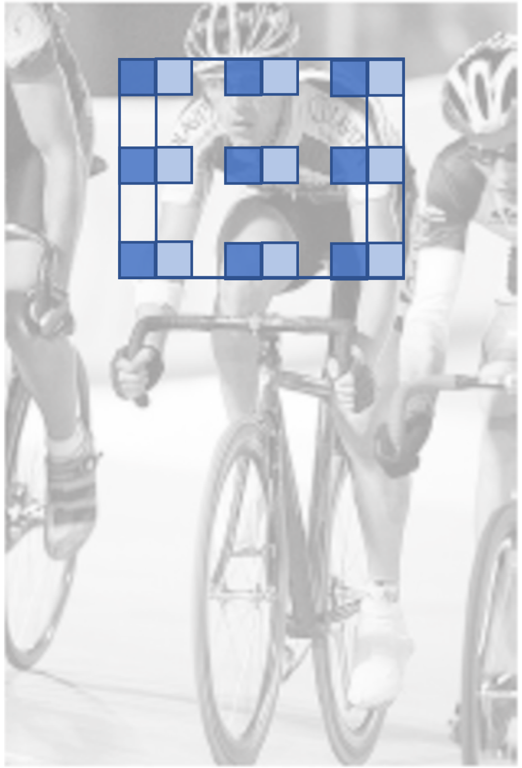
Solution 3: Dilation

- Need subsampling to allow convolutional layers to capture large regions with small filters
  - Can we do this without subsampling?



Solution 3: Dilation

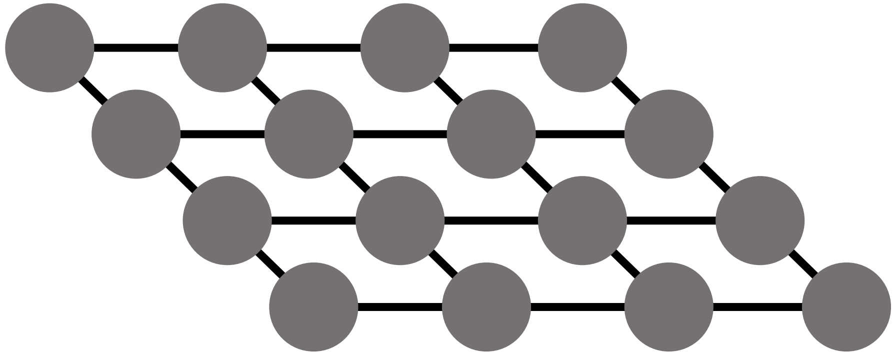
- Need subsampling to allow convolutional layers to capture large regions with small filters
  - Can we do this without subsampling?



Solution 3: Dilation

- Instead of subsampling by factor of 2: dilate by factor of 2
- Dilation can be seen as:
    - Using a much larger filter, but with most entries set to 0
    - Taking a small filter and “exploding”/ “dilating” it
- Not panacea: without subsampling, feature maps are much larger: memory issues

Solution 4: Conditional random fields



$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} e^{-E(\mathbf{y},\mathbf{x})}$$

$$\begin{aligned} \mathbf{y}^* &= \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \\ &= \arg\min_{\mathbf{y}} E(\mathbf{y},\mathbf{x}) \end{aligned}$$

$$E(\mathbf{y},\mathbf{x}) = \sum_i E_{data}(y_i,\mathbf{x}) + \sum_{i,j \in \mathcal{N}} E_{smooth}(y_i, y_j, \mathbf{x})$$

Solution 4: Conditional Random Fields

- Idea: take convolutional network prediction and sharpen using classic techniques
- *Conditional Random Field*

$$\mathbf{y}^* = \arg\min_{\mathbf{y}} \sum_i E_{data}(y_i, \mathbf{x}) + \sum_{i,j \in \mathcal{N}} E_{smooth}(y_i, y_j, \mathbf{x})$$

$$E_{smooth}(y_i, y_j, \mathbf{x}) = \underbrace{\mu(y_i, y_j)}_{\text{Label compatibility}} \underbrace{w_{ij}(\mathbf{x})}_{\text{Pixel similarity}}$$

Inference in CRFs

- Problem: combinatorial optimization
- Variational methods: Approximate complex distribution p(y) with simple distribution q(y)
- Mean-field approximation: q(y) is independent distribution for each pixel:

$$q(\mathbf{y}) = \prod_i q_i(y_i)$$

- If N pixels and K classes, basically N K-dimensional vectors

Mean field inference

- If we can find best q, solution is highest probability output for each pixel
- Try to match p with q by minimizing *Kulback-Leibler Divergence*

$$KL(q||p) = \sum_{\mathbf{y}} q(\mathbf{y}) \log p(\mathbf{y}) - \sum_{\mathbf{y}} q(\mathbf{y}) \log q(\mathbf{y})$$

- Iterative process: in each iteration, do coordinate ascent on one $q(y_i)$

Mean field inference

- Coordinate descent on $q_i(y_i)$
- At each step, keep other pixels fixed and update one
- Each step (approximately):
  - Take current $q_j(y_j)$ on all $j \neq i$
  - Use this to compute $p(y_i|y_{-i})$ where $y_{-i} = \{y_j : j \neq i\}$
  - Set q_i to this

$$q_i \propto \mathbb{E}_{q_{-i}}[\log p(y_i|y_{-i})]$$

Fully Connected CRFs

- Typically, only adjacent pixels connected
  - Fewer connections => Easier to optimize
- Dense connectivity: every pixel connected to everything else
- Intractable to optimize except if pairwise potential takes specific form

$$E_{smooth}(y_i, y_j, \mathbf{x}) = \mu(y_i, y_j) w_{ij}(\mathbf{x})$$

$$w_{ij}(\mathbf{x}) = \sum_m w_m e^{-\|\mathbf{f}_m(i) - \mathbf{f}_m(j)\|^2}$$

Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. Philipp Krahenbuhl, Vladlen Koltun. In *NIPS,* 2011.

Gaussian edge potentials

$$w_{ij}(\mathbf{x}) = \sum_m w_m e^{-\|\mathbf{f}_m(i)-\mathbf{f}_m(j)\|^2}$$

- What should **f** be?
- simple answer: color, position

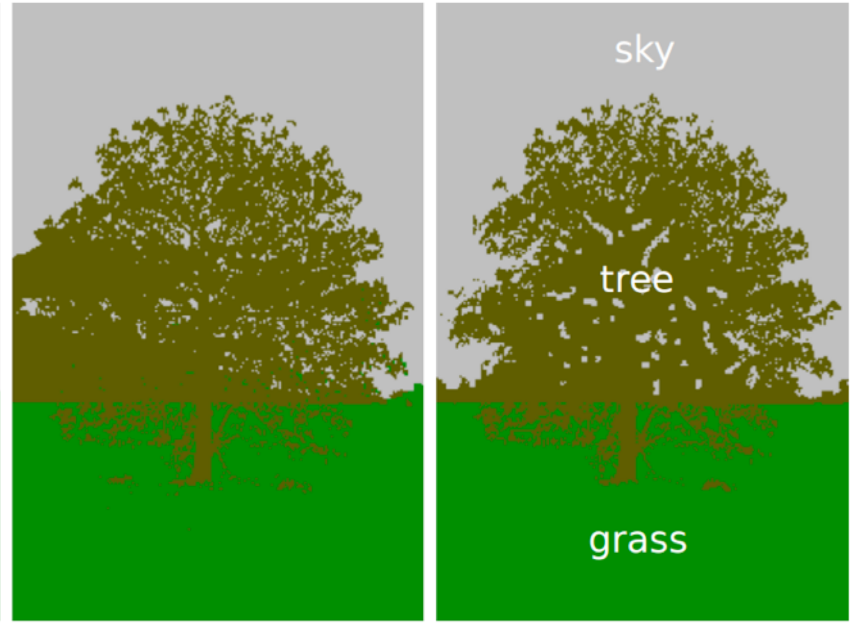
Mean field inference for Dense-CRF

$$q_i(y_i = l) \propto \exp[\underbrace{-\psi_u(y_i)}_{\text{Unary}} - \sum_{l'} \underbrace{\mu(l,l')}_{\text{Label compatibility transform}} \underbrace{\sum_m w_m \sum_{j \neq i} e^{-\|\mathbf{f}_m(i) - \mathbf{f}_m(j)\|^2} q_j(y_j = l')]}_{\text{Message passing}}$$

$$\mathbf{q}_i \propto \exp[-\boldsymbol{\psi}_u^{(i)} - \boldsymbol{\mu} \sum_j \mathbf{m}_{j \to i}]$$

Fully Connected CRFs

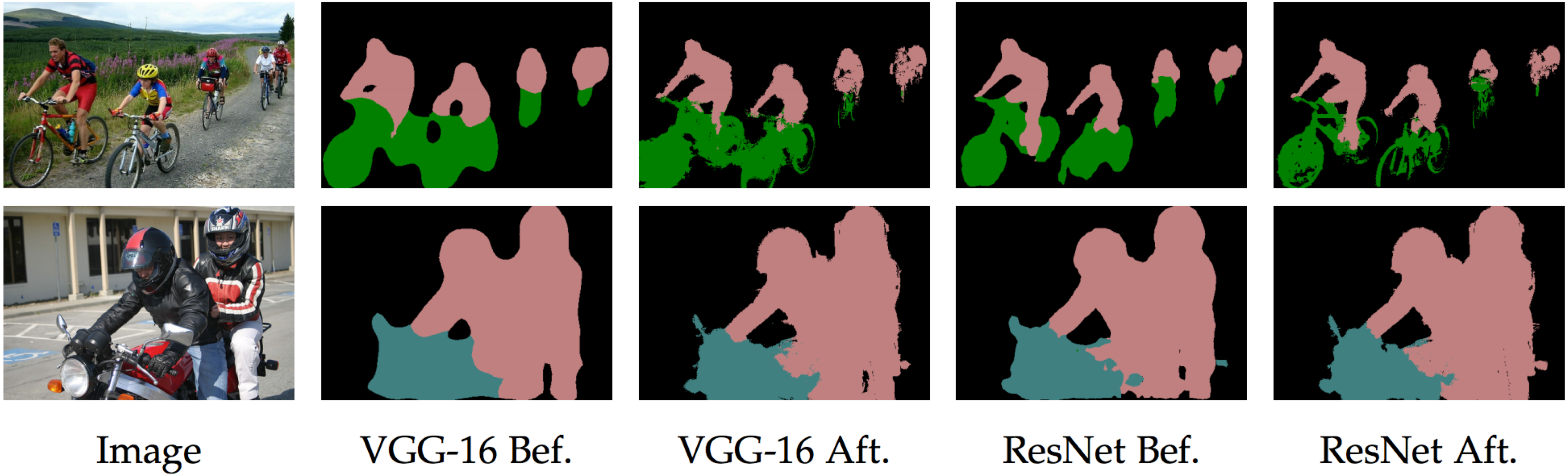
Grid CRF



Fully connected CRF

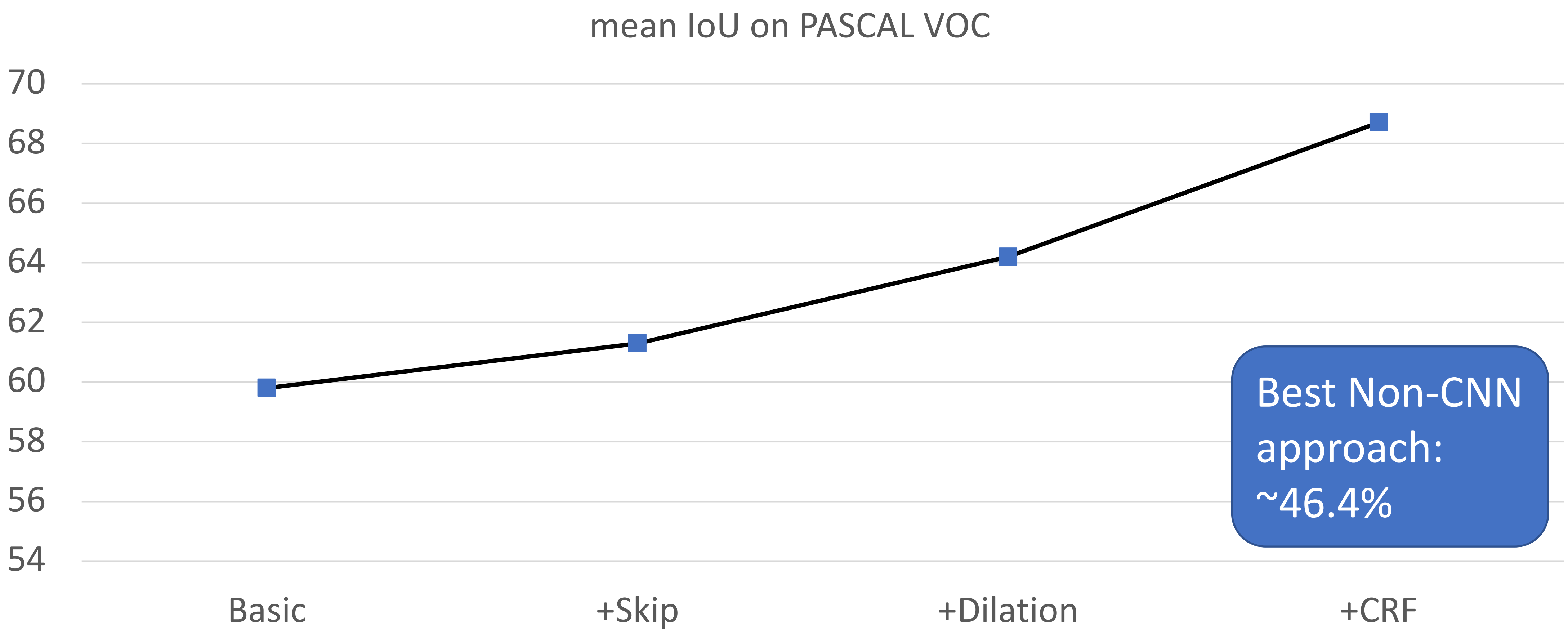
Ground truth

Fully connected CRFs



Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan Yuille. In *ICLR,* 2015.

Putting it all together



Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan Yuille. In *ICLR,* 2015.

Other additions

Method	mean IoU (%)
VGG16 + Skip + Dilation	65.8
ResNet101	68.7
ResNet101 + Pyramid	71.3
ResNet101 + Pyramid + COCO	74.9

DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan Yuille. Arxiv 2016.

Mean field inference as a recurrent network

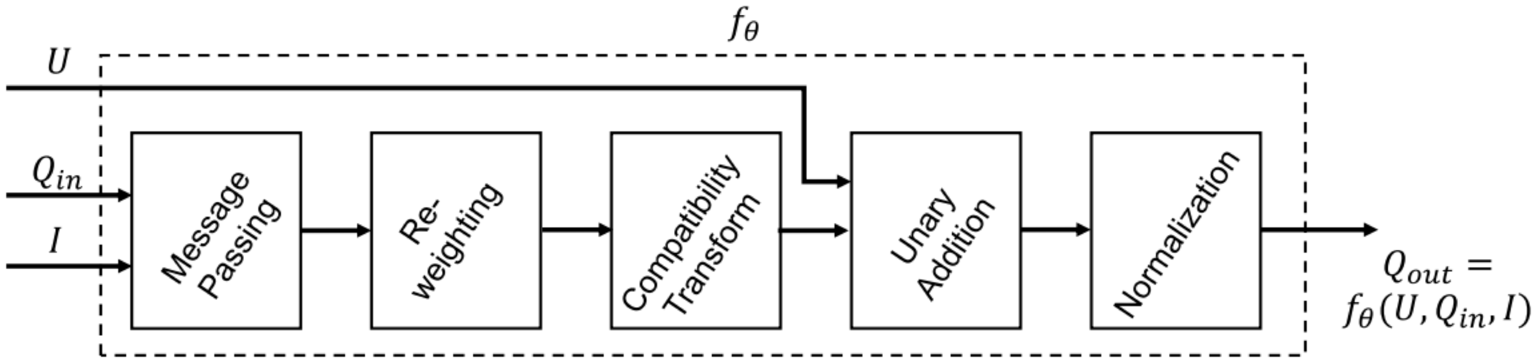
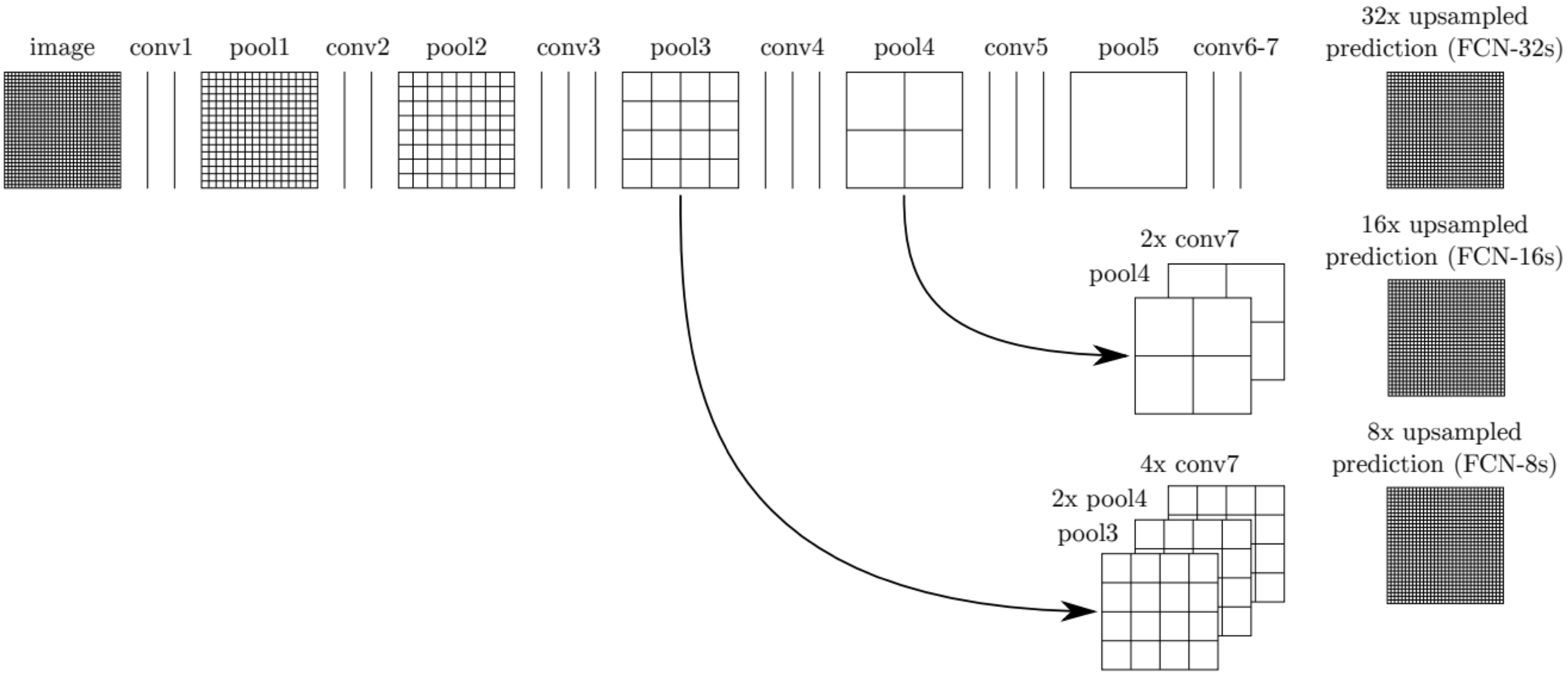


Figure 1. **A mean-field iteration as a CNN.** A single iteration of the mean-field algorithm can be modelled as a stack of common CNN layers.

Zheng, Shuai, et al. "Conditional random fields as recurrent neural networks." *Proceedings of the IEEE international conference on computer vision*. 2015.

Skip connections



Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

Alternative: coarse-to-fine prediction

- Inspiration 1: we are making independent predictions from each layer
- Inspiration 2: CRF-like approaches require iterated inference
- Inspiration 3: Coarse-to-fine refinement works because: coarse scales capture large scale structure coarsely, fine scales capture fine-scale structures

U-Net

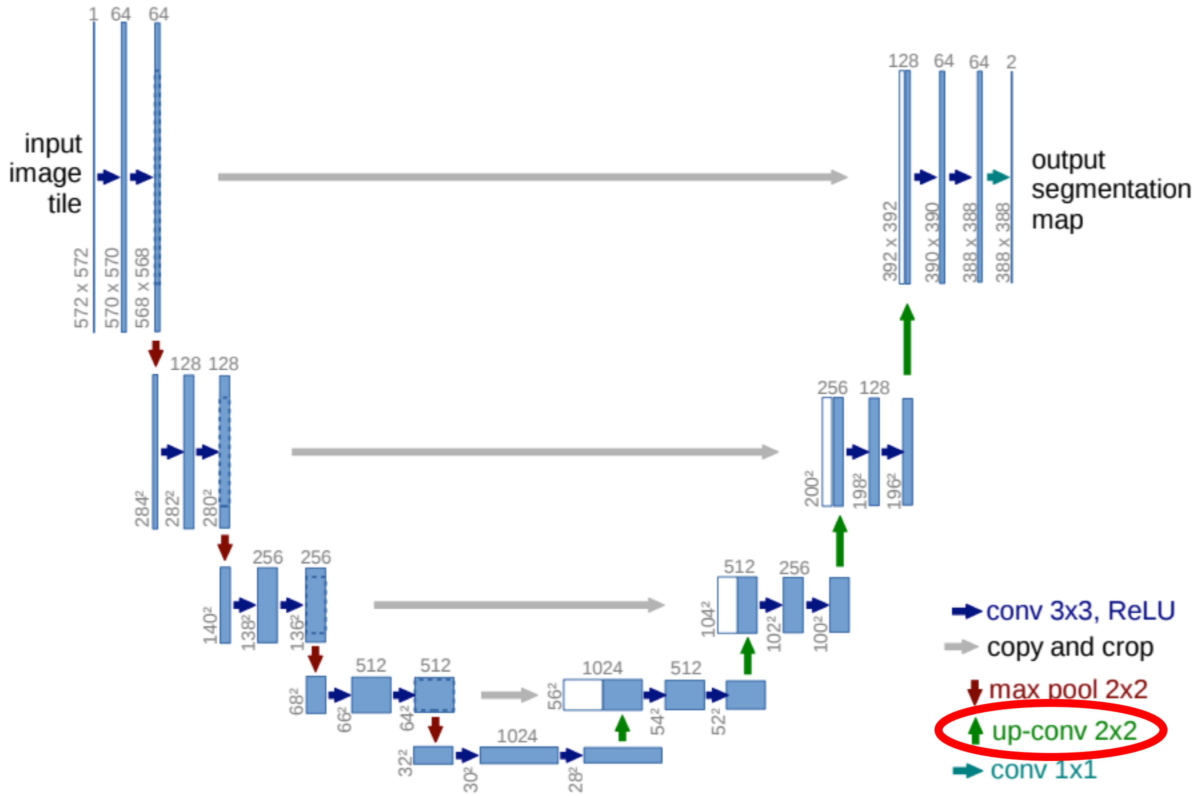


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

Learned upsampling

- Bilinear upsampling

$$y[i] = \begin{cases} x[i/2] & \text{if } i \text{ is even} \\ \frac{(x[(i+1)/2]+x[(i-1)/2])}{2} & \text{ow} \end{cases}$$

- Assume fractional indices in x are 0
- Assume w[-1] = w[1] = 0.5, w[0] = 1
- Then

$$y[i] = \sum_{k=-1}^{1} w[k]x[(i-k)/2]$$

Learned upsampling

$$y[i] = \sum_{k=-1}^{1} w[k]x[(i-k)/2]$$

- Looks remarkably like convolution
- But output size is twice input size
- Filter w can be learnt!
- “Up-convolution”, “Transposed convolution”, ~~“Deconvolution”~~

Image-to-image translation problems

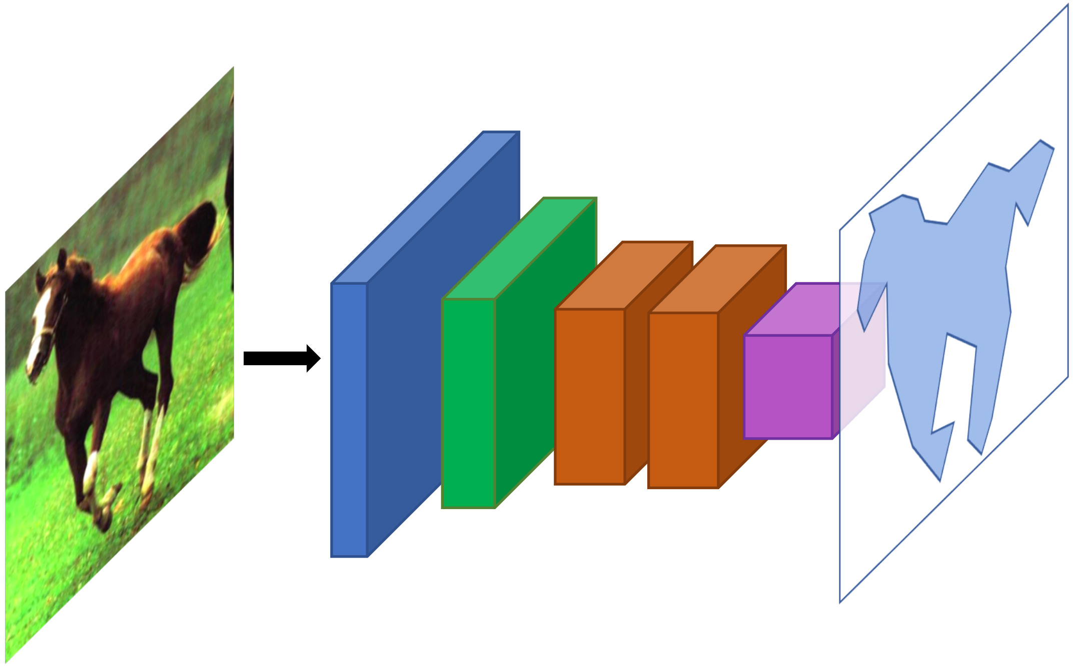
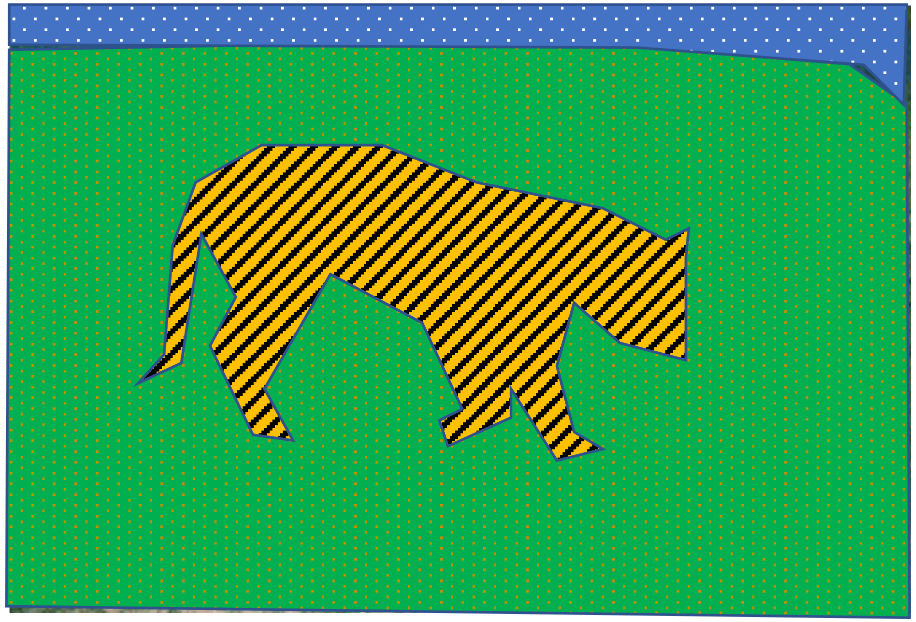
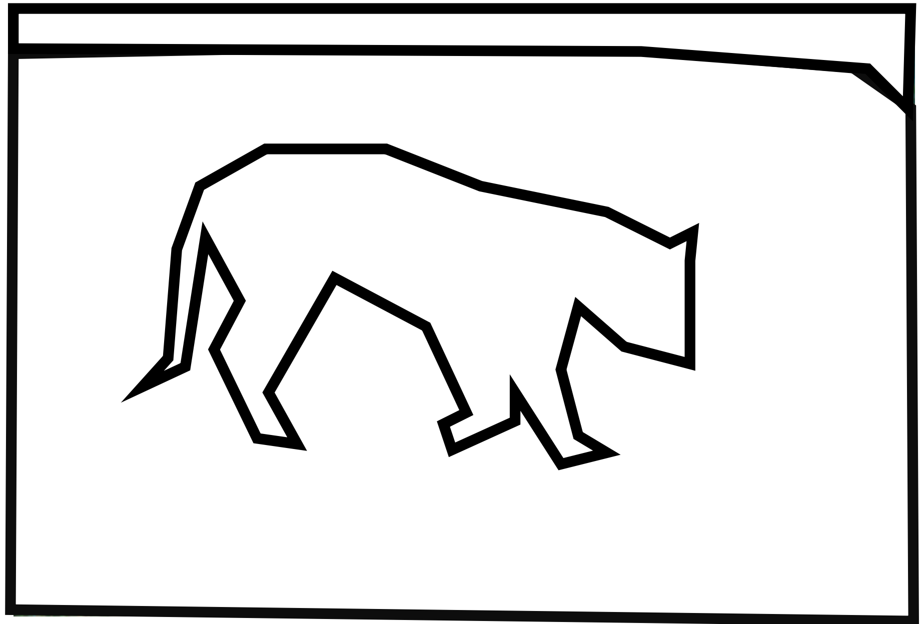


Image-to-image translation problems

- Segmentation
- Optical flow estimation
- Depth estimation
- Normal estimation
- Boundary detection
- …

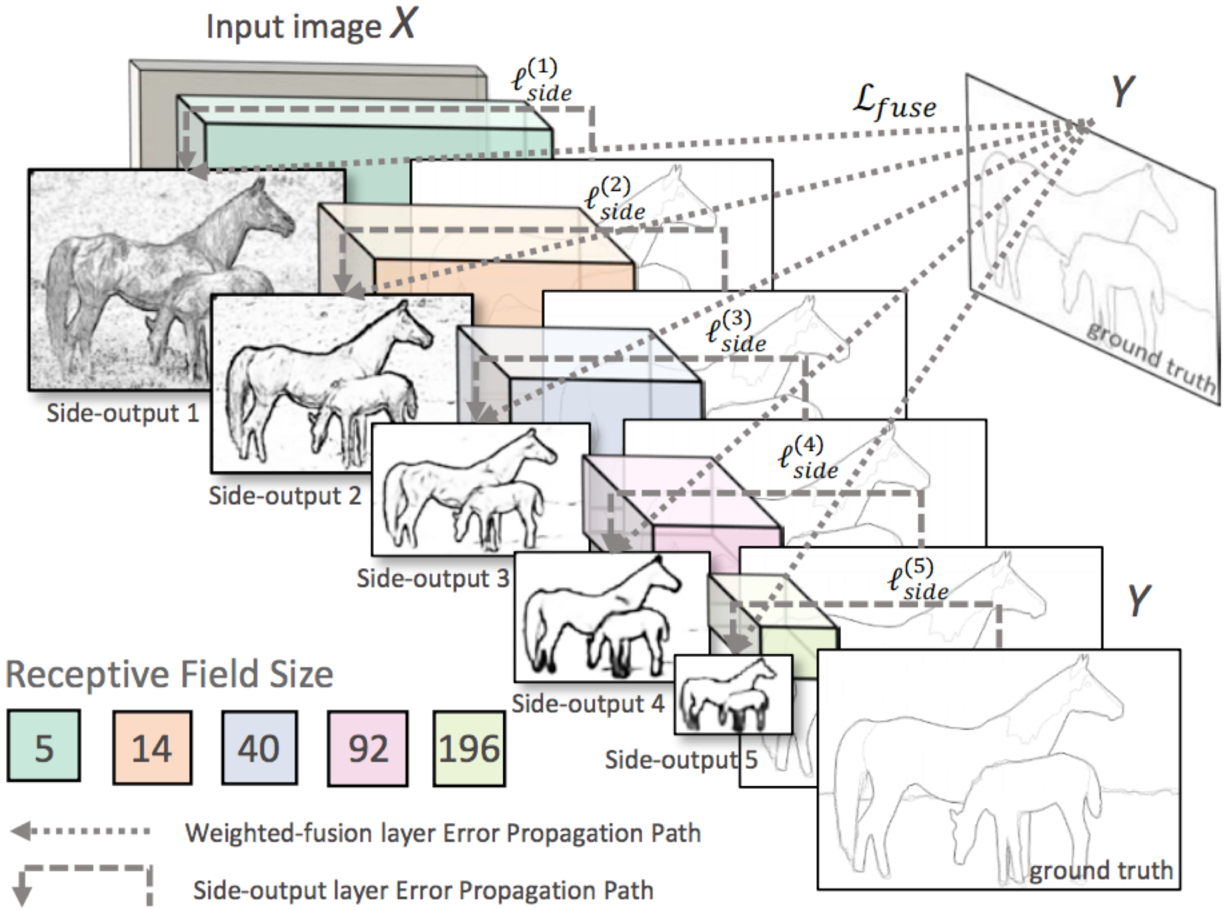
Revisiting contour detection





Convolutional network based edge detection

- Deep supervision: Skip connections, but additional loss at each layer



Method	Max F measure
Structured Edges	74.6
HED without deep supervision	77.1
HED with deep supervision	78.2
Humans	80

Holistically-Nested Edge Detection. Saining Xie, Zhuowen Tu. In *ICCV,* 2015.

Convolutional network based edge detection