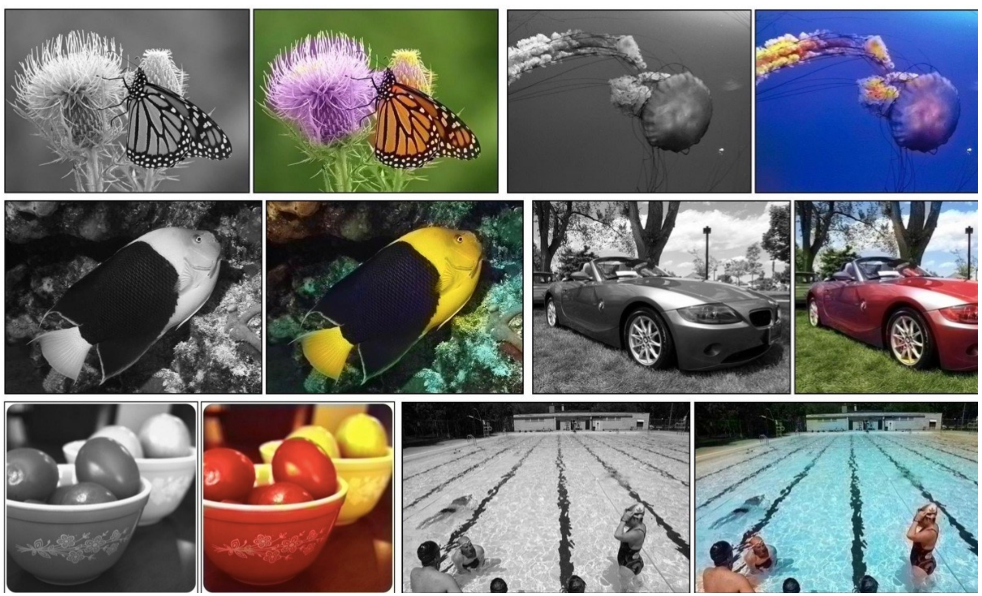
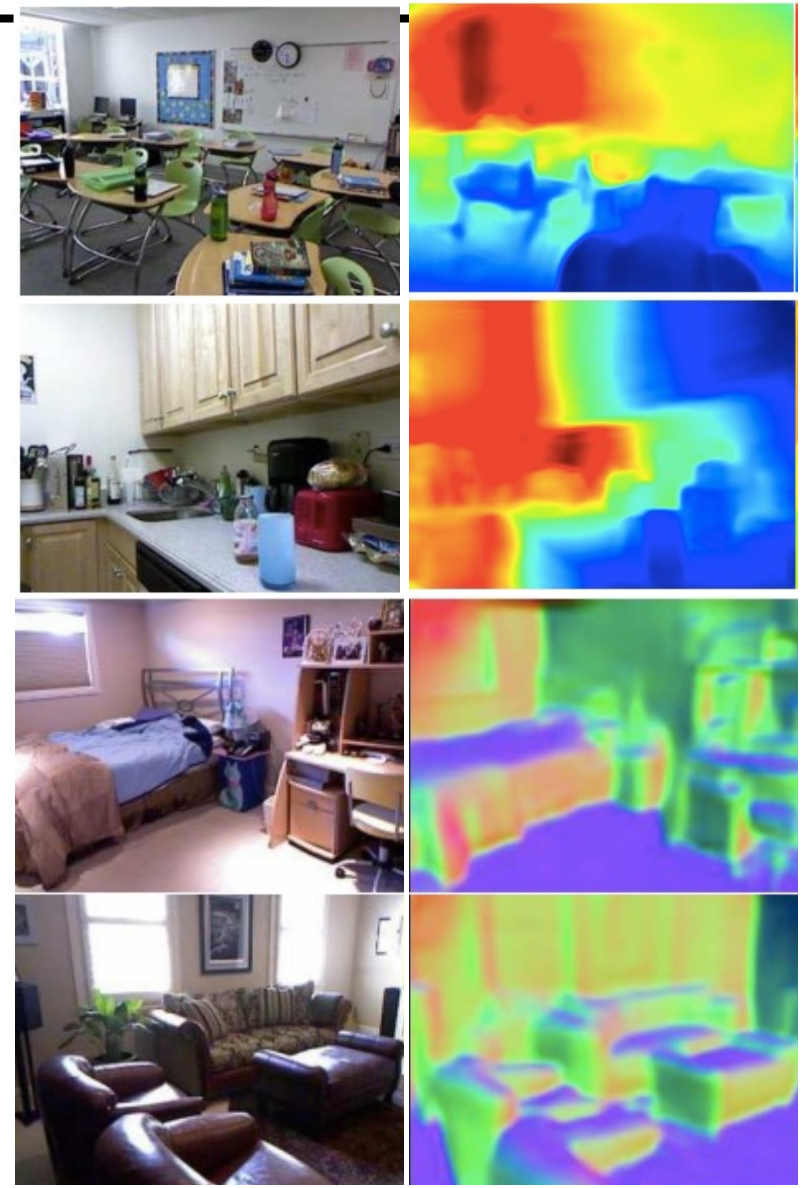
Pixel Prediction Tasks

Semantic segmentation



Colorization



Depth / Surface Normal Estimation

Many Slides from L. Lazebnik.

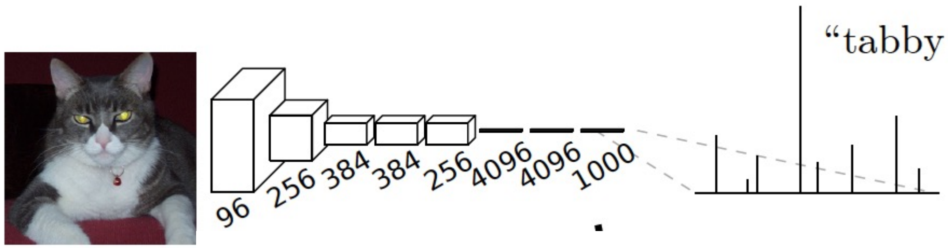
Outline

- Semantic segmentation
  - Architectures
    - “Convolutionalization”
    - Dilated convolutions
    - Hyper-columns / skip-connections
    - Learned up-sampling architectures
- Other dense prediction problems

From Image Classifiers to Semantic Segmentation

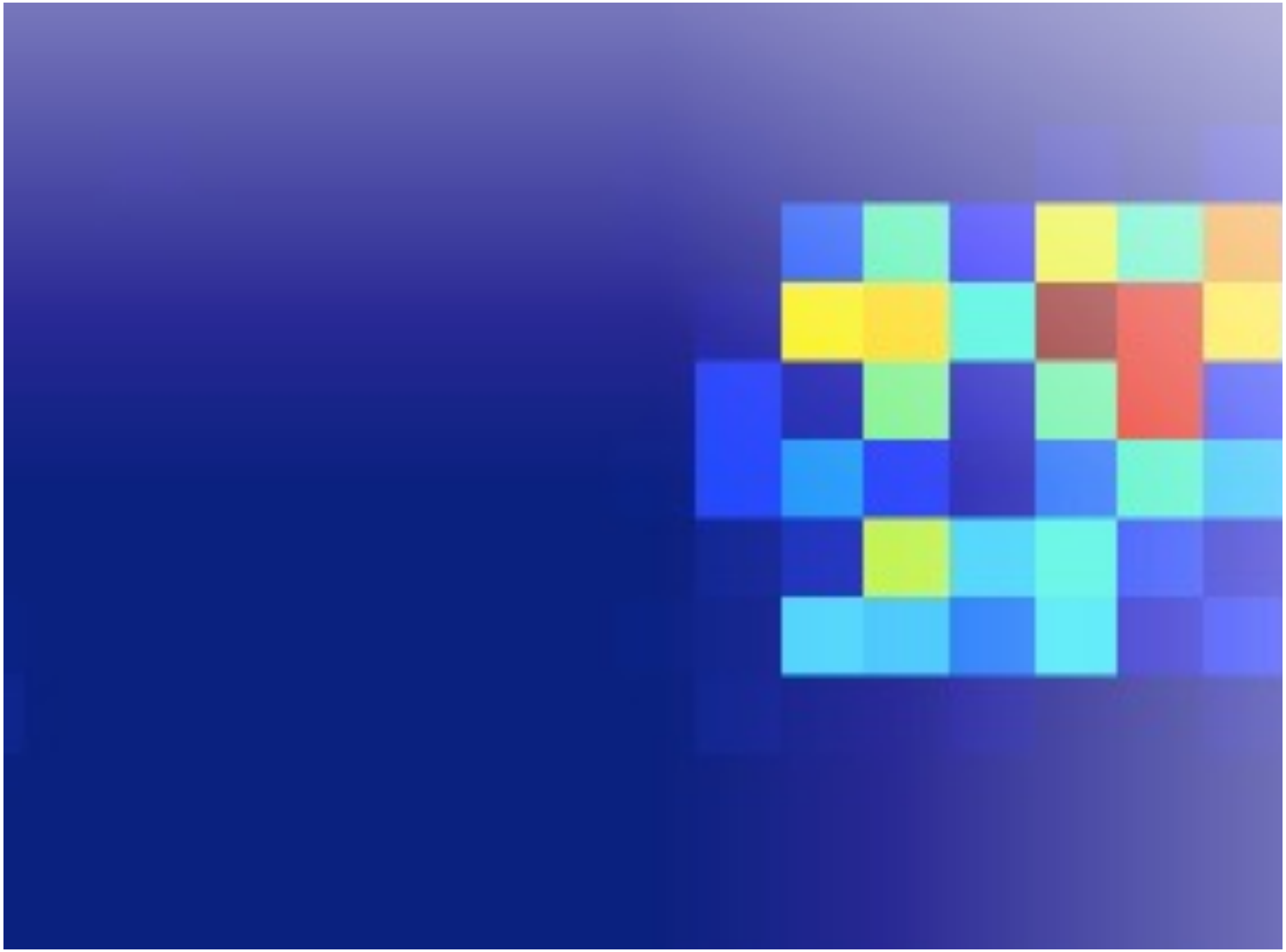
Have: an pre-trained image classification network

Want: pixel-wise predictions on arbitrary sized images



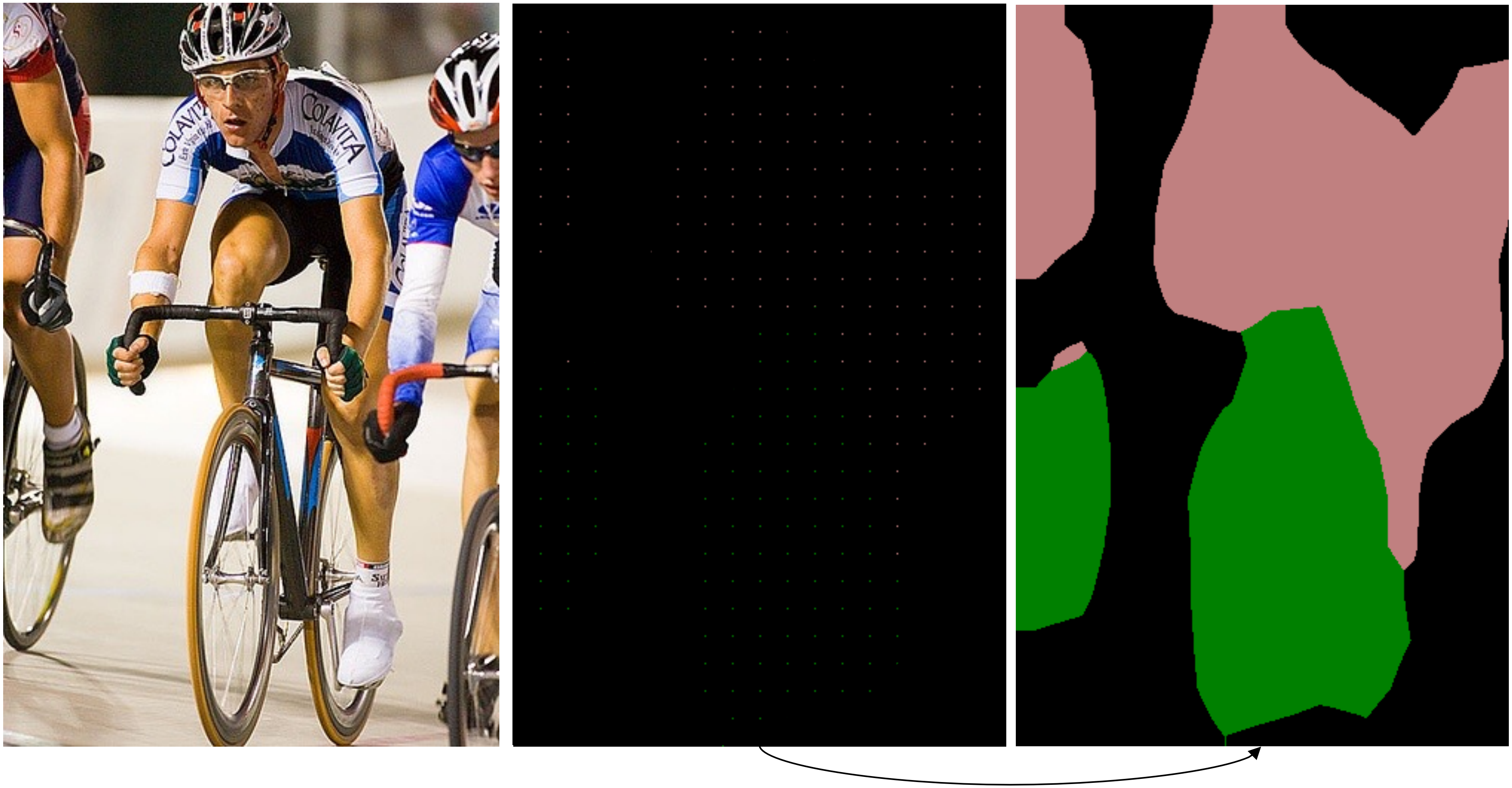
J. Long, E. Shelhamer, and T. Darrell, [Fully Convolutional Networks for Semantic Segmentation](), CVPR 2015

Sparse, Low-resolution Output



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

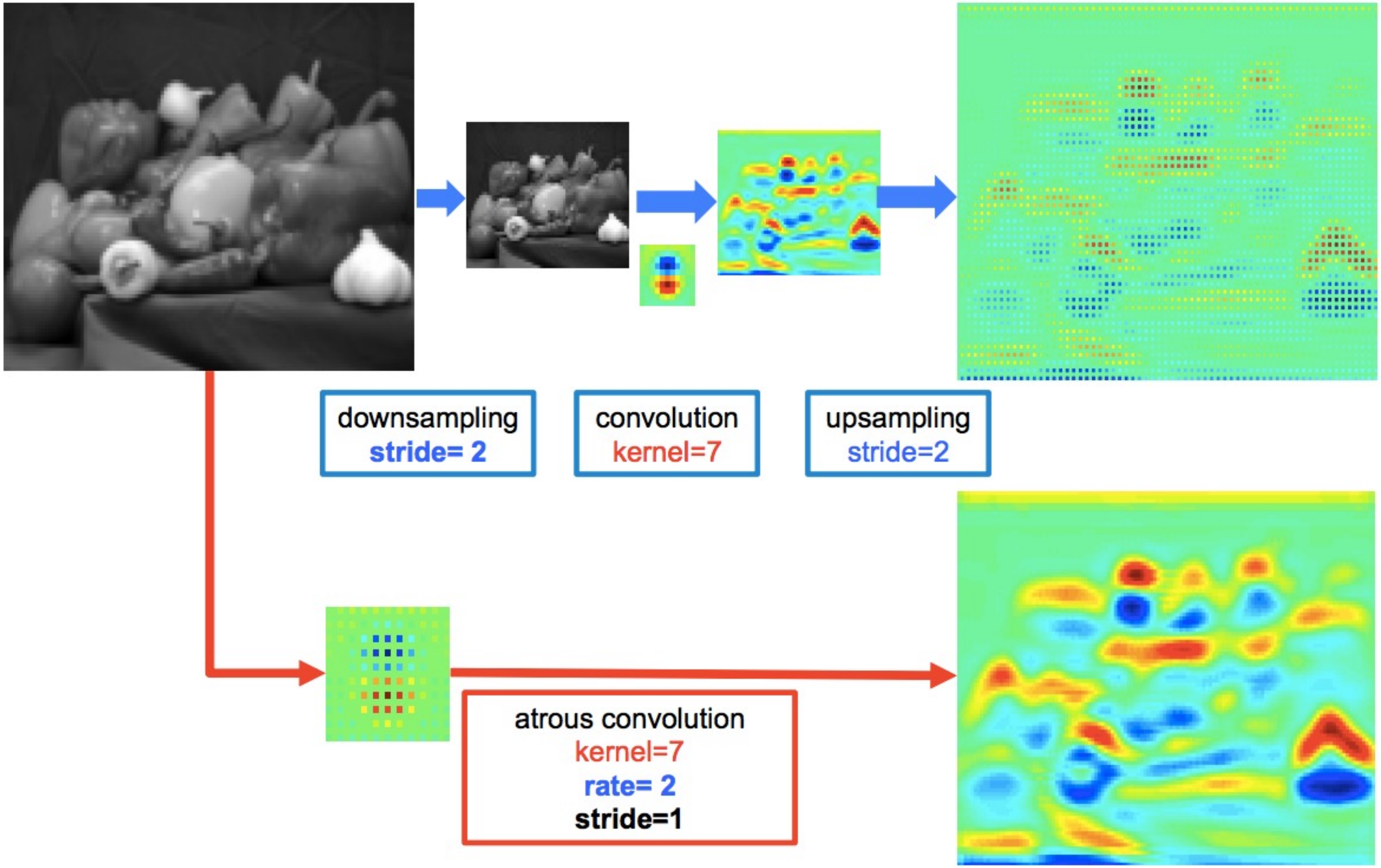
Sparse, Low-resolution Output



Bilinear Up sampling: Differentiable,
train through up-sampling.

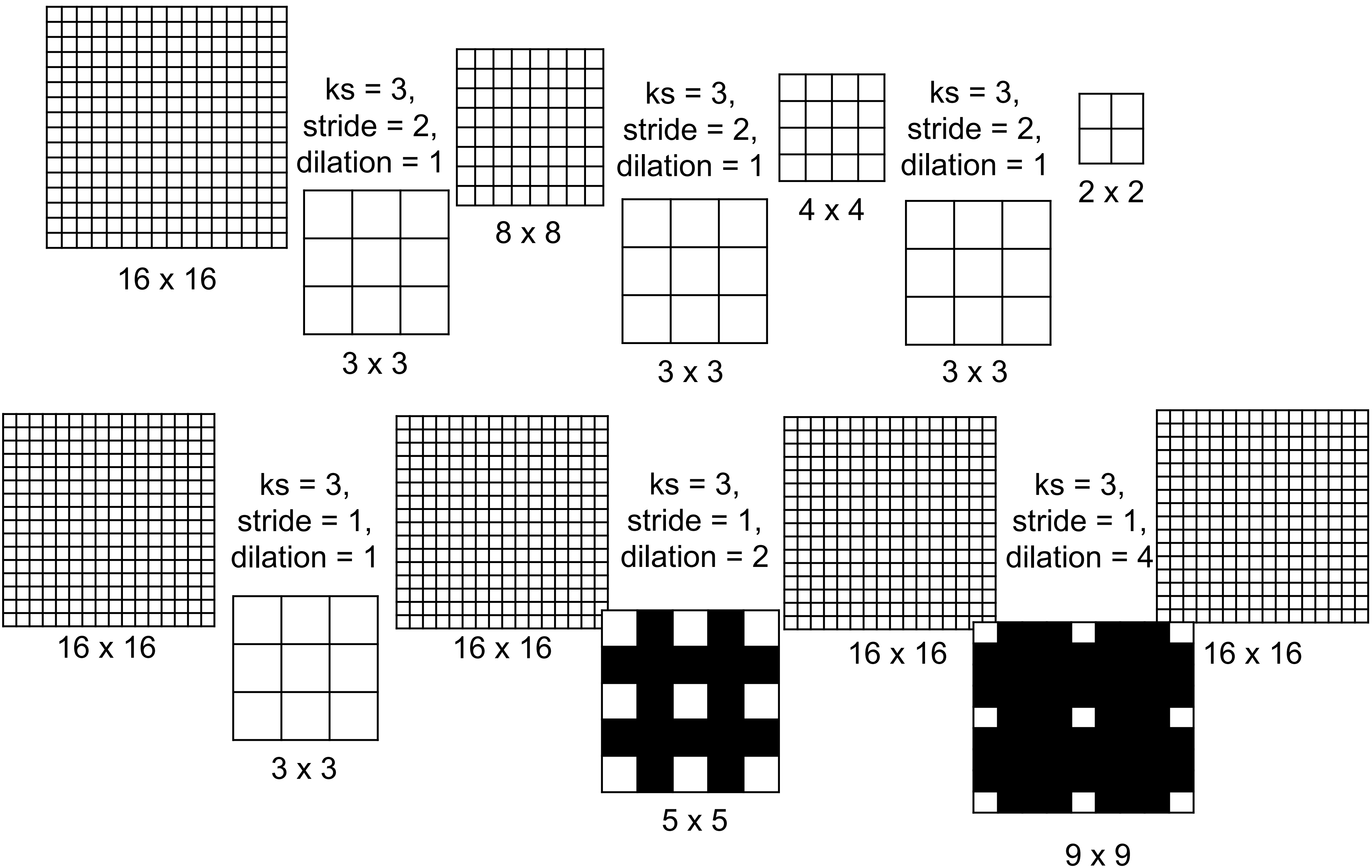
J. Long, et al., Fully Convolutional Networks for Semantic Segmentation, CVPR 2015

Fix 1: A trous Conv., Dilated Conv.



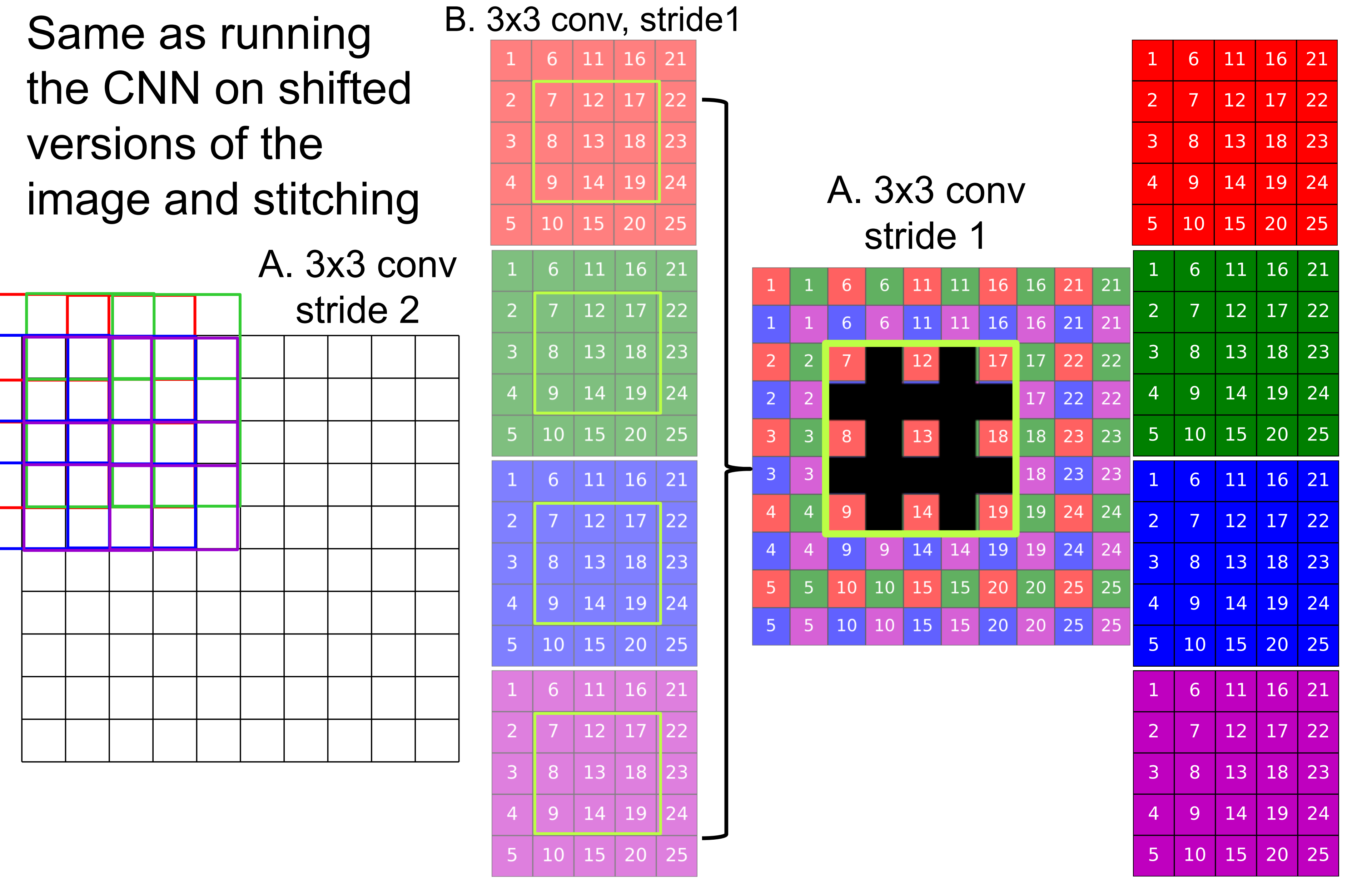
L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. Yuille, [DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs](), PAMI 2017

Fix 1: A trous Conv., Dilated Conv.



Fix 1: A trous Conv., Dilated Conv.

Same as running the CNN on shifted versions of the image and stitching



Fix 1: A trous Conv., Dilated Conv.

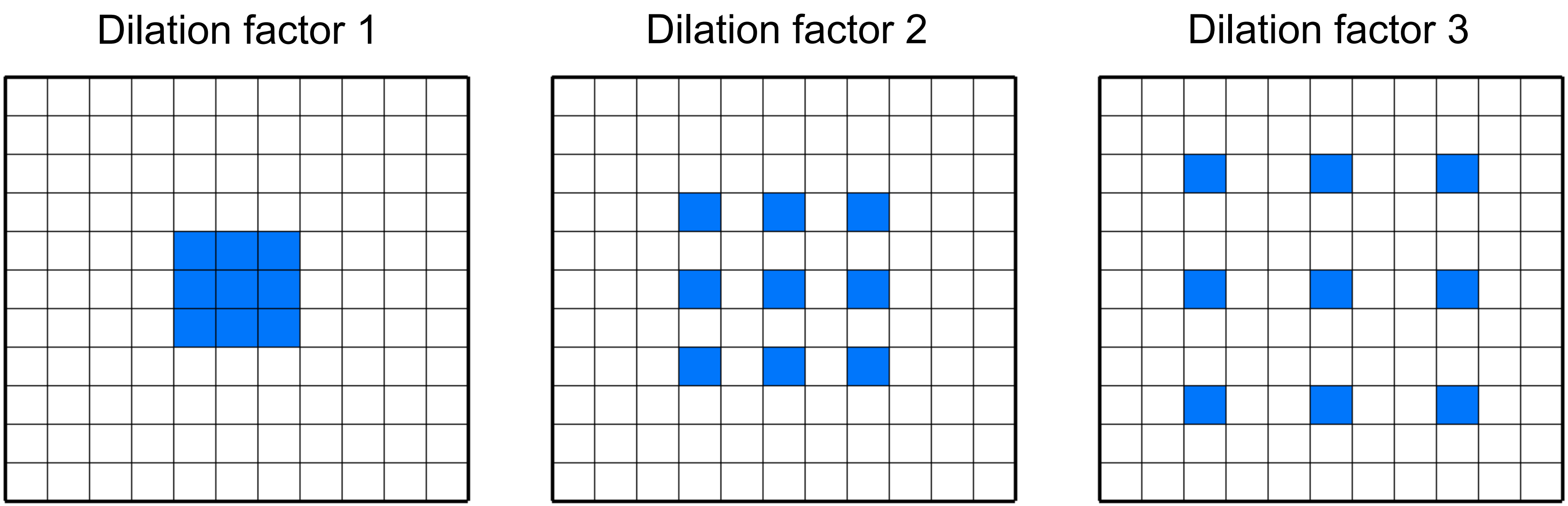


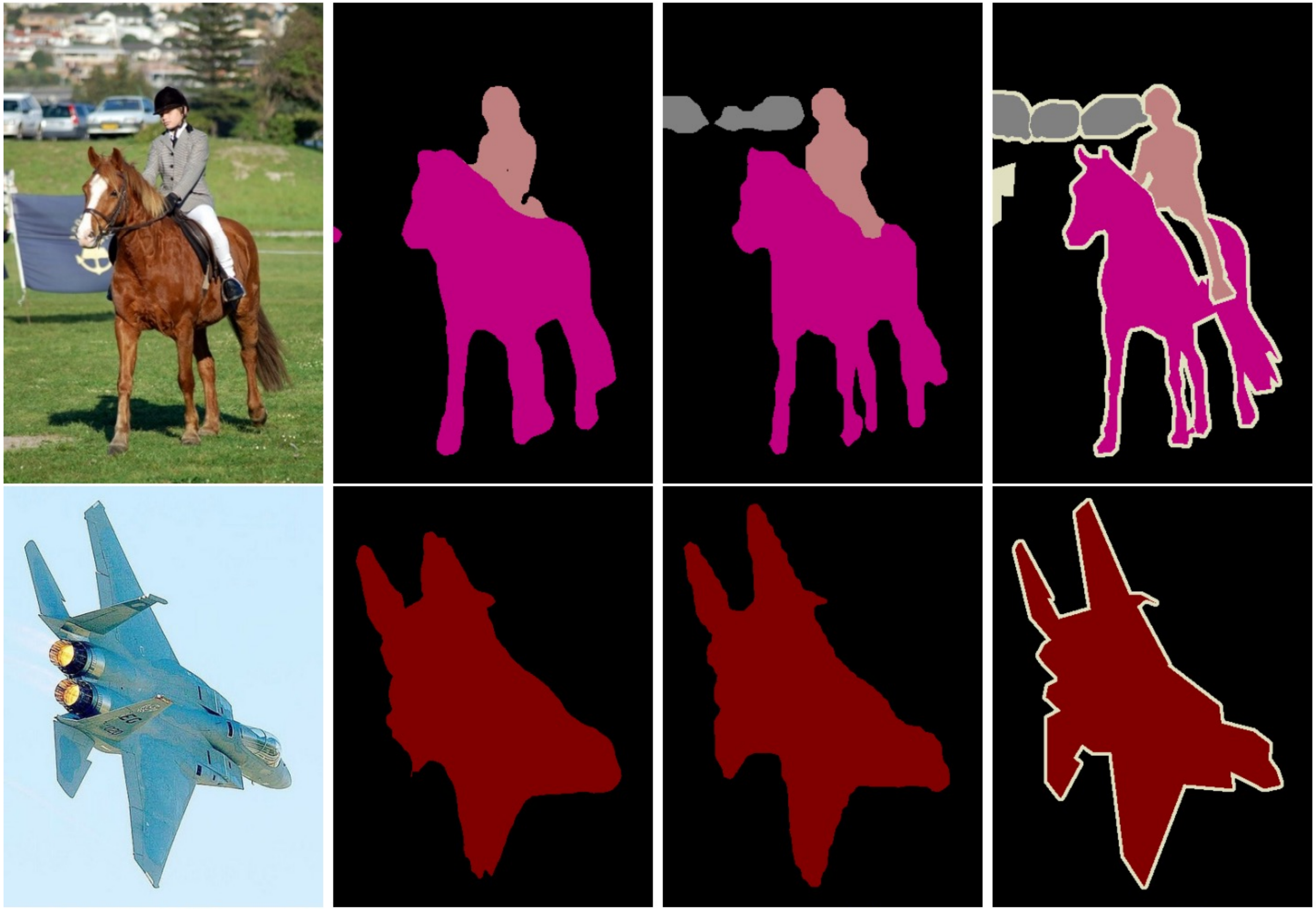
Image source

Fix 1: A trous Conv., Dilated Conv.

- Use in FCN to remove downsampling: change stride of max pooling layer from 2 to 1, dilate subsequent convolutions by factor of 2 (possibly without re-training any parameters)
- Instead of reducing spatial resolution of feature maps, use a large sparse filter

L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. Yuille, [DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs](), PAMI 2017

Fix 1: A trous Conv., Dilated Conv.

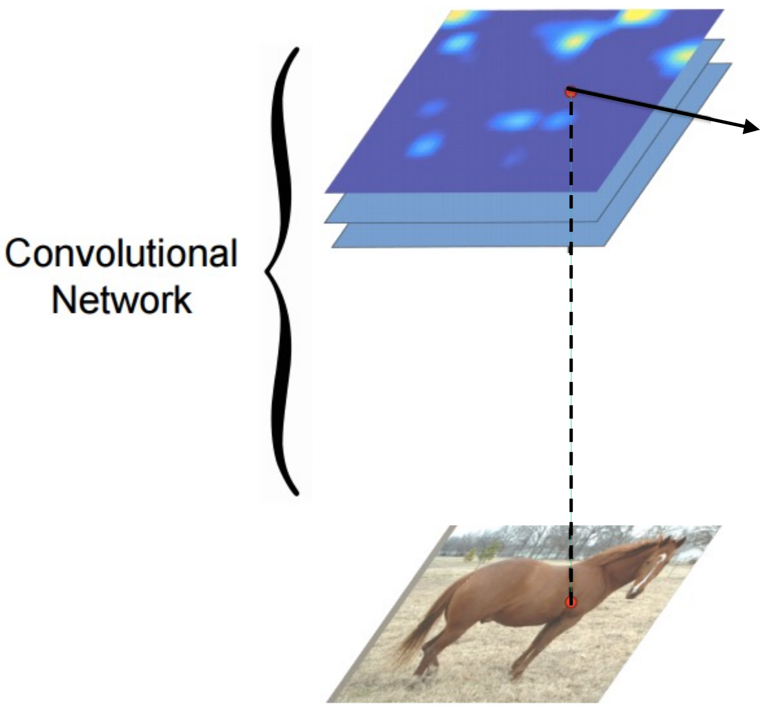


FCN Dilated Convolutions

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

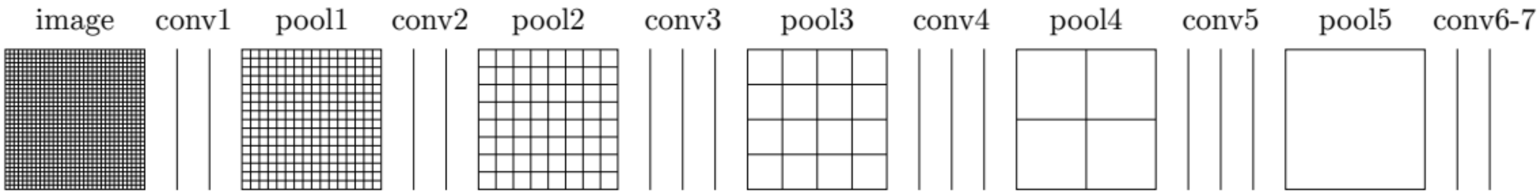
Fix 2: Hyper-columns/Skip Connections

- Even though with dilation we can predict each pixel, fine-grained information needs to be propagated through the network.
- Idea: Additionally use features from within the network.



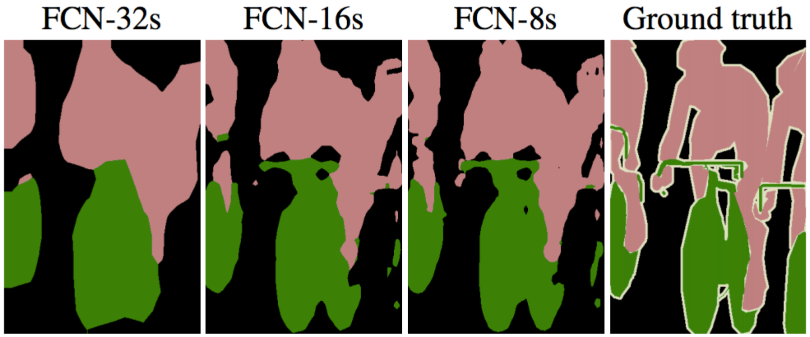
B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik, Hypercolumns for Object Segmentation and Fine-grained Localization, CVPR 2015
J. Long, et al., Fully Convolutional Networks for Semantic Segmentation, CVPR 2015

Fix 2: Hyper-columns/Skip Connections



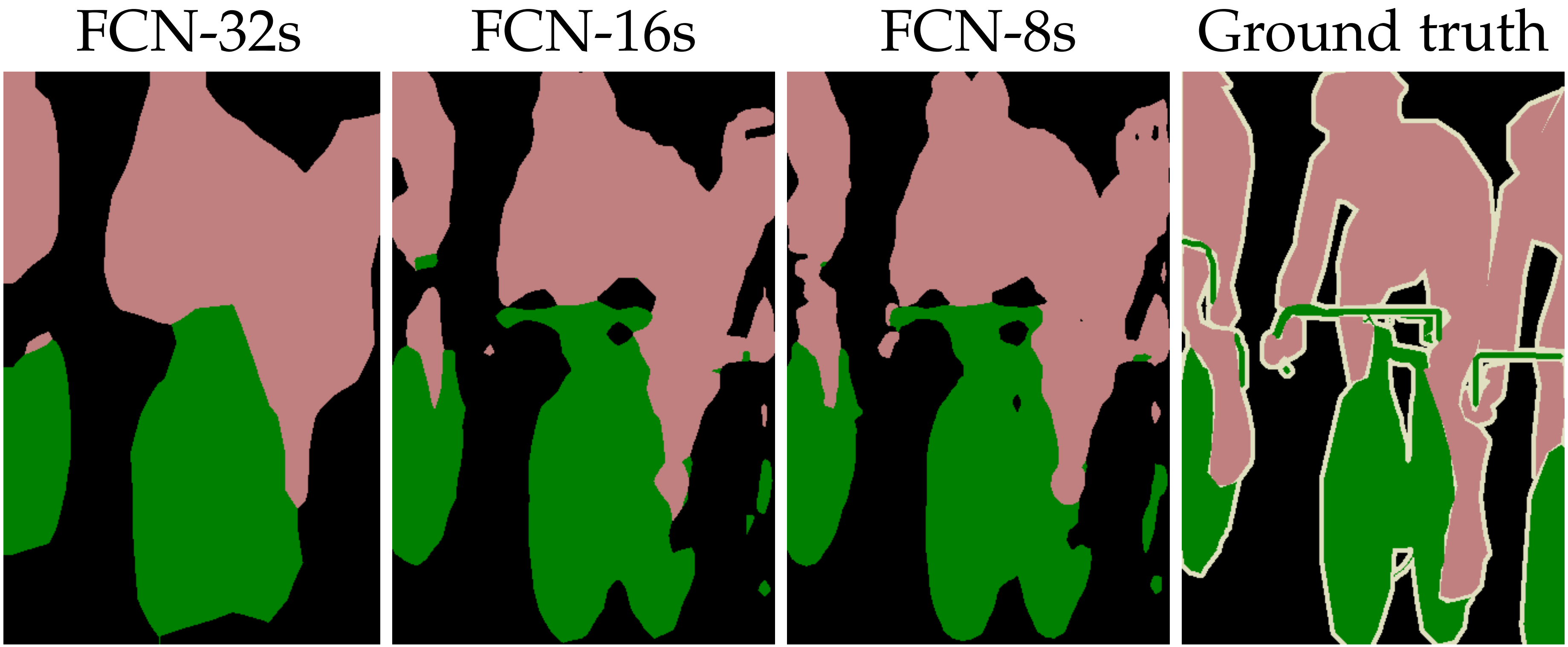
- Predictions by 1x1 conv layers, bilinear upsampling

- Predictions by 1x1 conv layers, *learned* 2x upsampling, fusion by summing



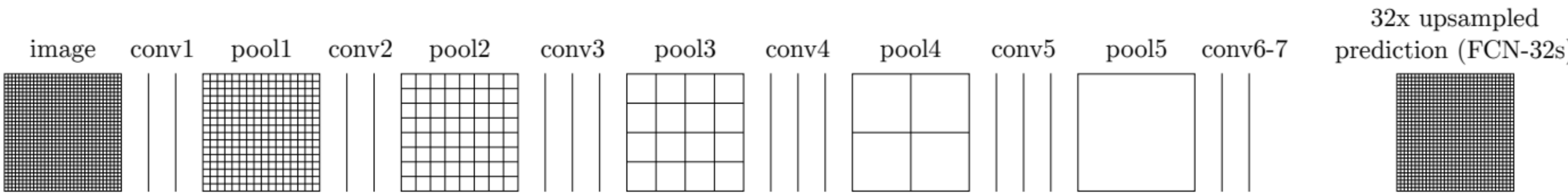
J. Long, E. Shelhamer, and T. Darrell, [Fully Convolutional Networks for Semantic Segmentation](), CVPR 2015

Fix 2: Hyper-columns/Skip Connections



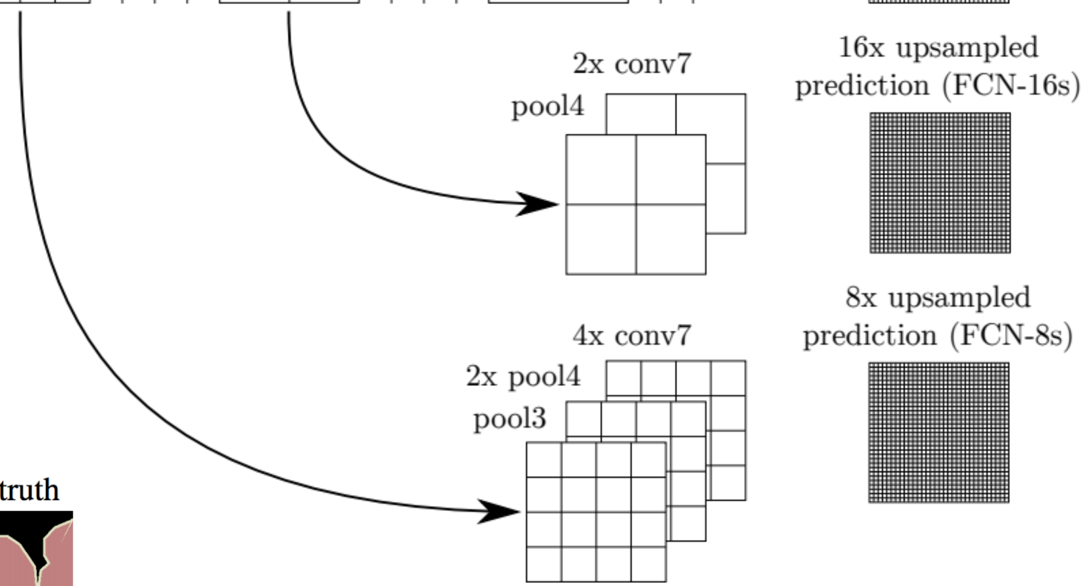
J. Long, et al., [Fully Convolutional Networks for Semantic Segmentation](), CVPR 2015

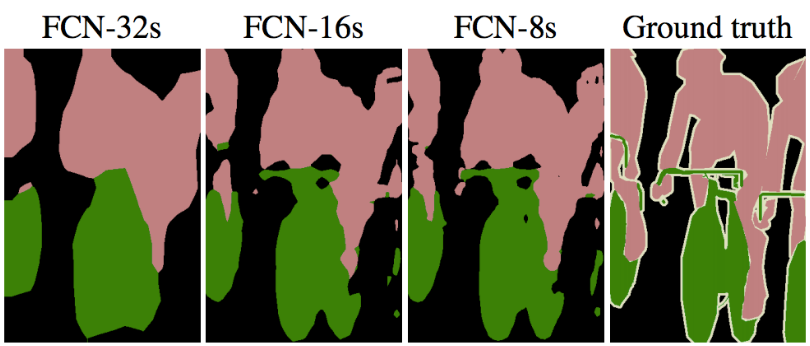
Fix 2b: Learned Upsampling



- Predictions by 1x1 conv layers, bilinear upsampling

- Predictions by 1x1 conv layers, *learned* 2x upsampling, fusion by summing

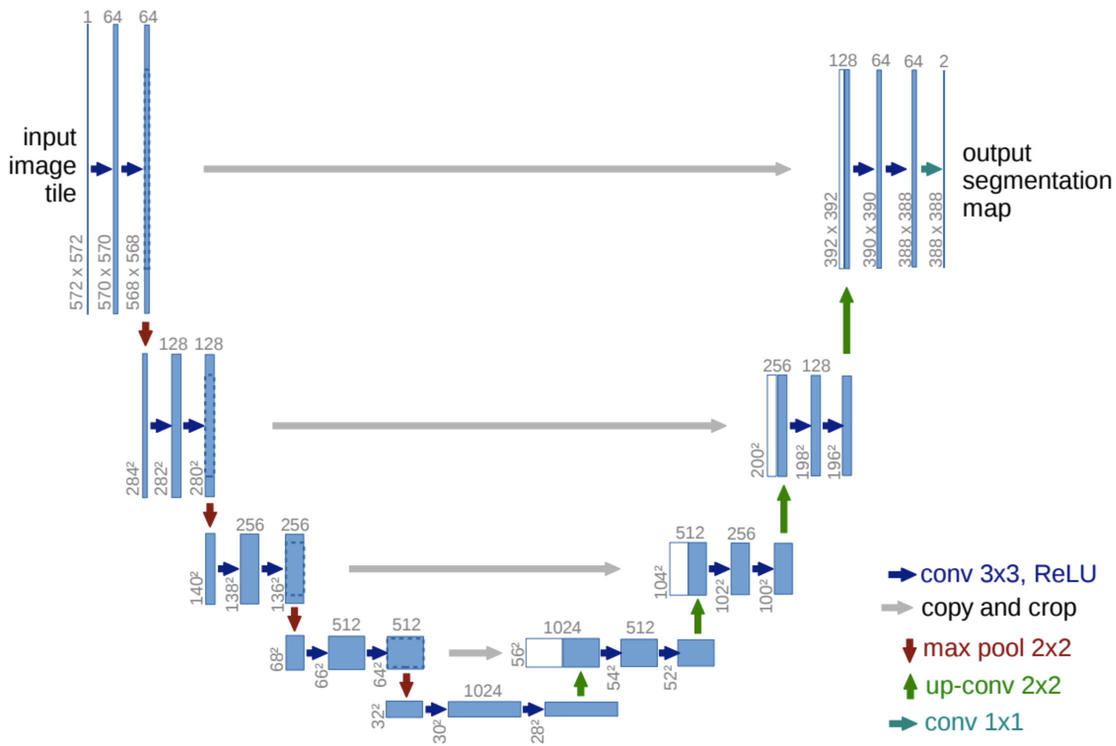




J. Long, E. Shelhamer, and T. Darrell, [Fully Convolutional Networks for Semantic Segmentation](), CVPR 2015

U-Net

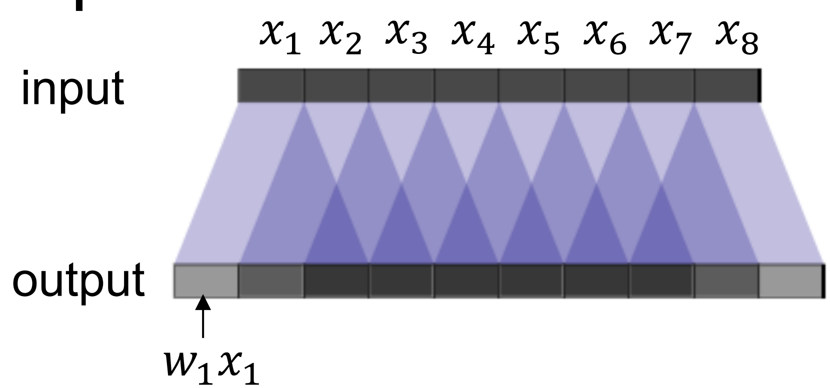
- Like FCN, fuse upsampled higher-level feature maps with higher-res, lower-level feature maps
- Unlike FCN, fuse by concatenation, predict at the end



O. Ronneberger, P. Fischer, T. Brox [U-Net: Convolutional Networks for Biomedical Image Segmentation](), MICCAI 2015

Learned Upsampling (Transposed convolution)

- Use the filter to "paint" in the output: place copies of the filter on the output, multiply by corresponding value in the input, sum where copies of the filter overlap
- 1D example:

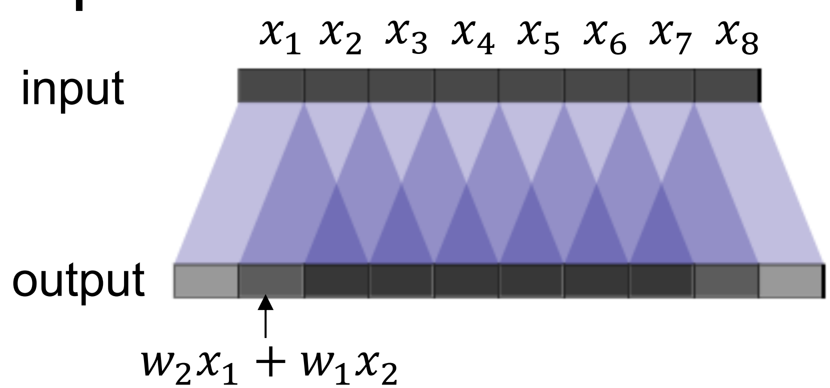


filter: $[w_1 \quad w_2 \quad w_3]$

Animation: https://distill.pub/2016/deconv-checkerboard/

Transposed convolution

- Use the filter to “paint” in the output: place copies of the filter on the output, multiply by corresponding value in the input, sum where copies of the filter overlap
- 1D example:

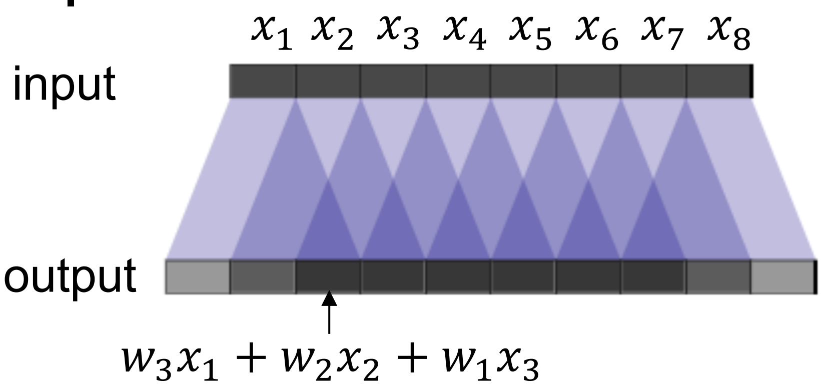


filter: $[w_1 \quad w_2 \quad w_3]$

Animation: https://distill.pub/2016/deconv-checkerboard/

Transposed convolution

- Use the filter to "paint" in the output: place copies of the filter on the output, multiply by corresponding value in the input, sum where copies of the filter overlap
- 1D example:



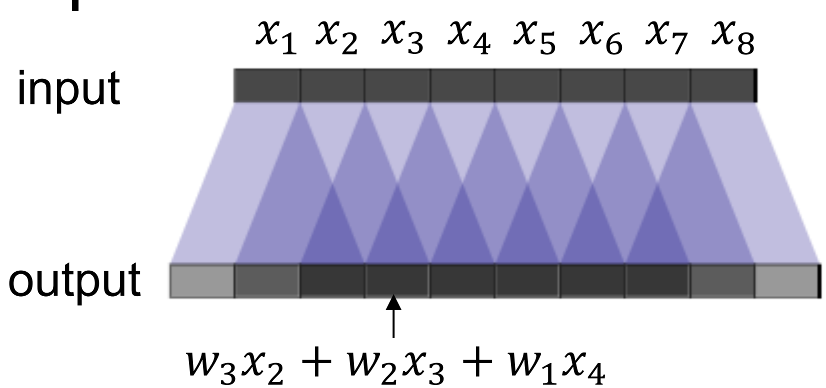
filter: $[w_1 \quad w_2 \quad w_3]$

Same as convolution with a flipped filter!

Animation: https://distill.pub/2016/deconv-checkerboard/

Transposed convolution

- Use the filter to "paint" in the output: place copies of the filter on the output, multiply by corresponding value in the input, sum where copies of the filter overlap
- 1D example:



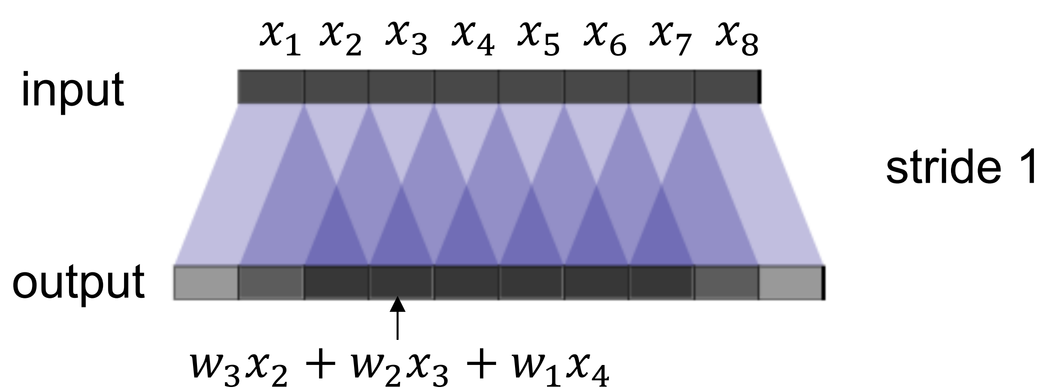
filter: $[w_1 \quad w_2 \quad w_3]$

Same as convolution with a flipped filter!

Animation: https://distill.pub/2016/deconv-checkerboard/

Upsampling by transposed convolution

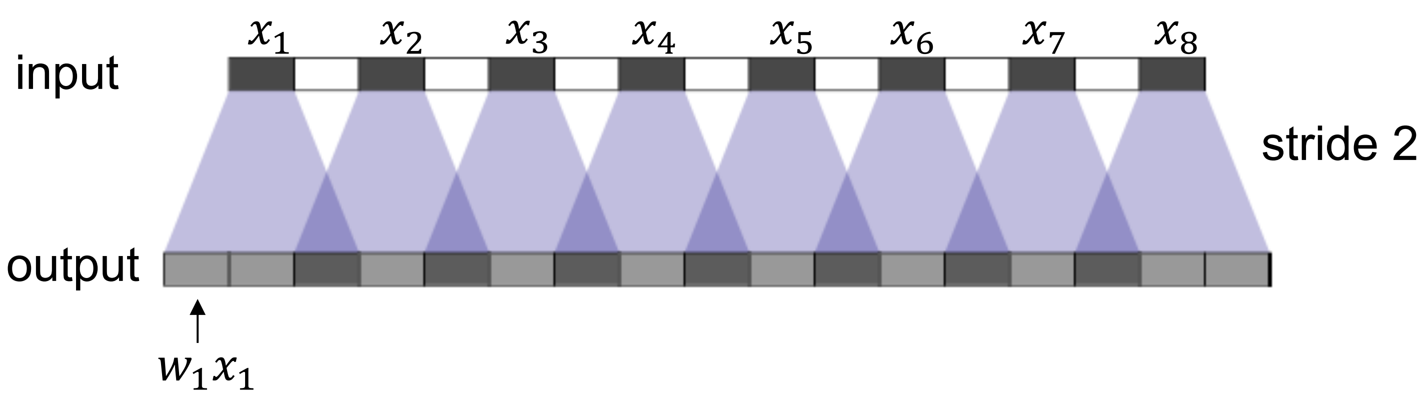
- *Backwards-strided convolution*: to increase resolution, use *output stride* > 1



Animation: https://distill.pub/2016/deconv-checkerboard/

Upsampling by transposed convolution

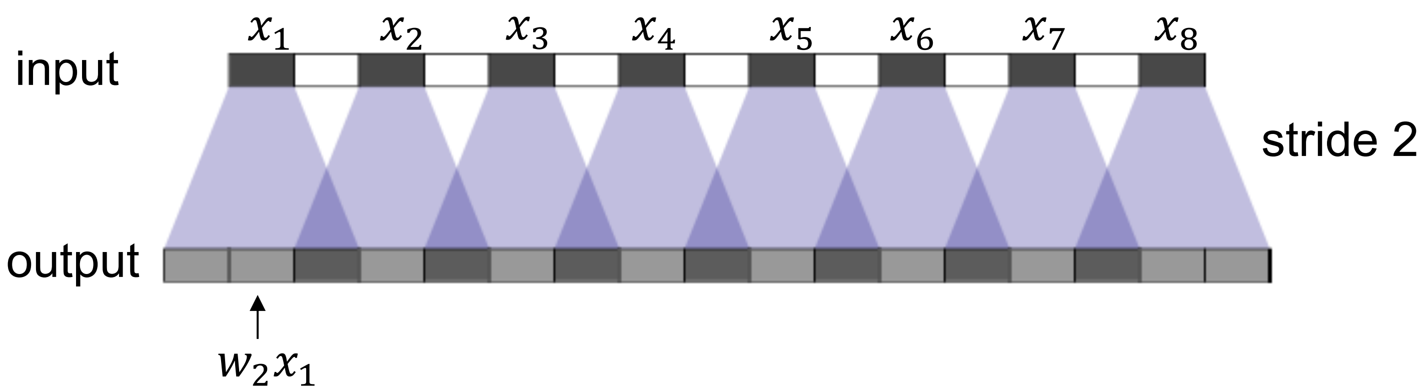
- *Backwards-strided convolution*: to increase resolution, use *output stride* > 1



Animation: https://distill.pub/2016/deconv-checkerboard/

Upsampling by transposed convolution

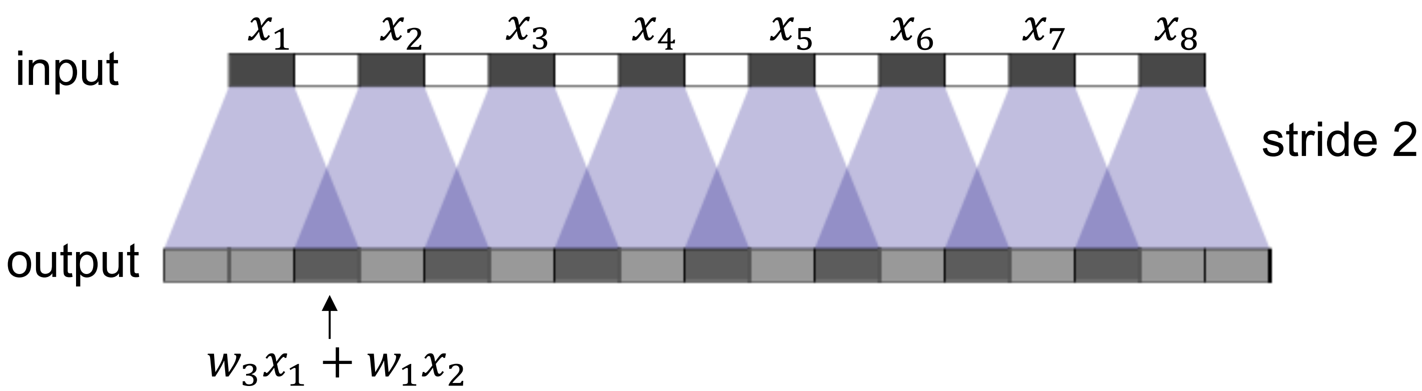
- *Backwards-strided convolution*: to increase resolution, use *output stride* > 1



Animation: https://distill.pub/2016/deconv-checkerboard/

Upsampling by transposed convolution

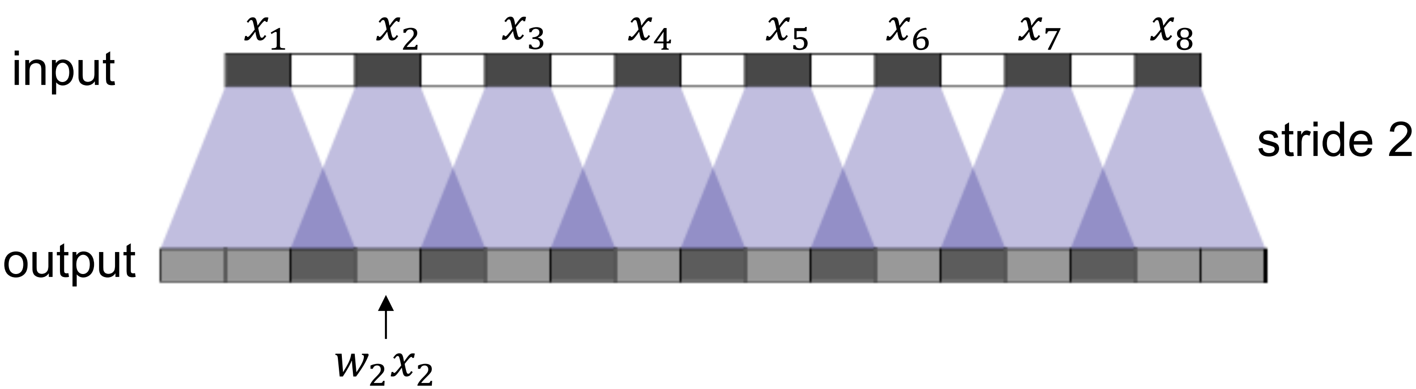
- *Backwards-strided convolution*: to increase resolution, use *output stride* > 1



Animation: https://distill.pub/2016/deconv-checkerboard/

Upsampling by transposed convolution

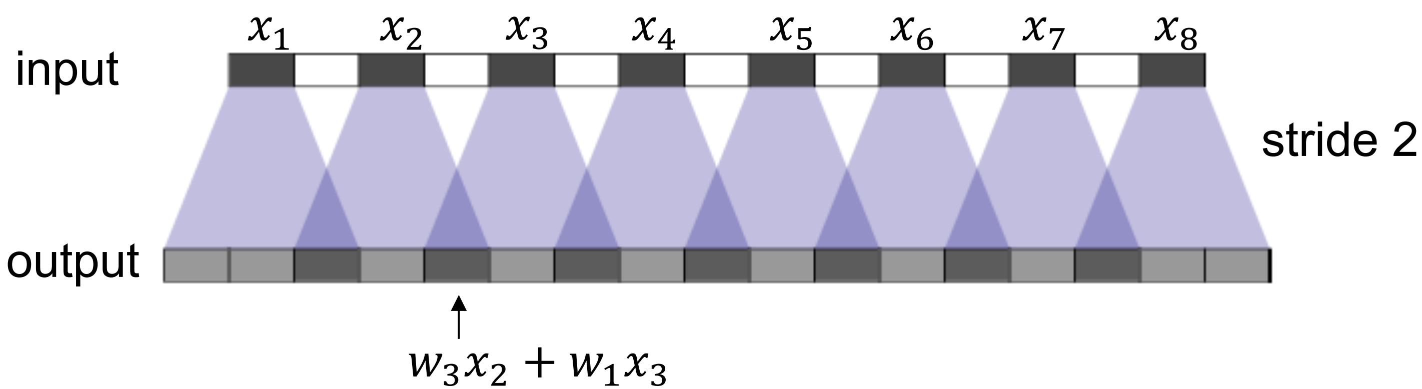
- *Backwards-strided convolution*: to increase resolution, use *output stride* > 1



Animation: https://distill.pub/2016/deconv-checkerboard/

Upsampling by transposed convolution

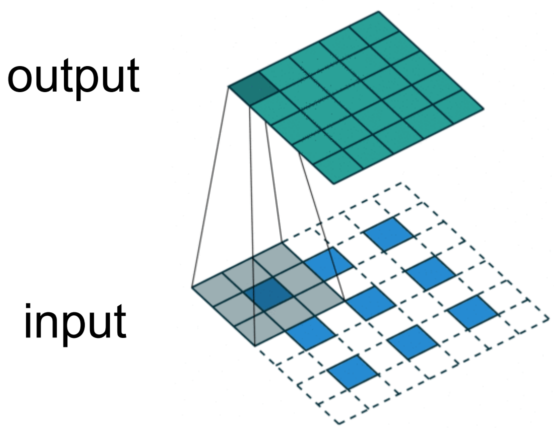
- *Backwards-strided convolution*: to increase resolution, use *output stride* > 1



Animation: https://distill.pub/2016/deconv-checkerboard/

Upsampling by transposed convolution

- *Backwards-strided convolution*: to increase resolution, use *output stride* > 1
  - For stride 2, dilate the input by inserting rows and columns of zeros between adjacent entries, convolve with flipped filter
  - Sometimes called convolution with *fractional input stride* 1/2



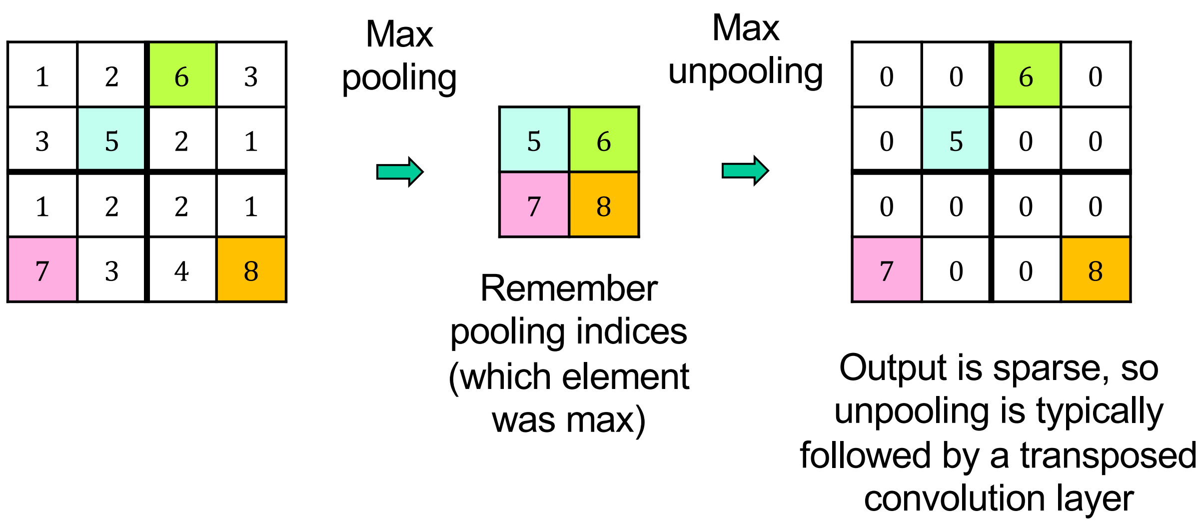
Q: What 3x3 filter would correspond to bilinear upsampling?

$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$
$\frac{1}{2}$	1	$\frac{1}{2}$
$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$

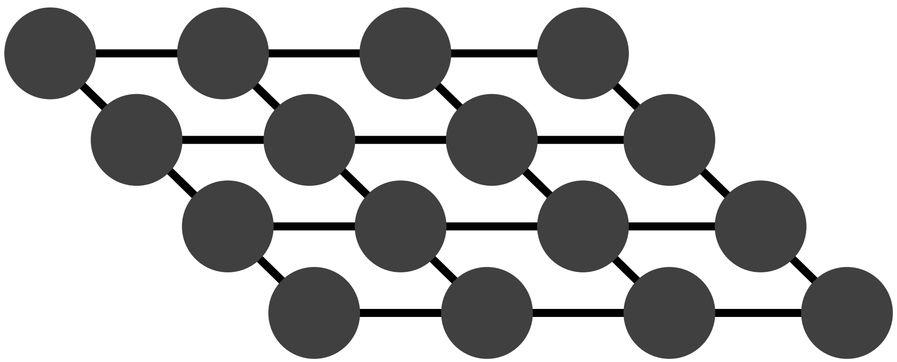
V. Dumoulin and F. Visin, [A guide to convolution arithmetic for deep learning](), arXiv 2018

Upsampling by unpooling

- Alternative to transposed convolution: max unpooling



Fix 3: Use local edge information (CRFs)



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

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$$

$$= \arg\min_{\mathbf{y}} E(\mathbf{y},\mathbf{x})$$

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

Source: B. Hariharan

Fix 3: Use local edge information (CRFs)

Idea: take convolutional network prediction and sharpen using classic techniques

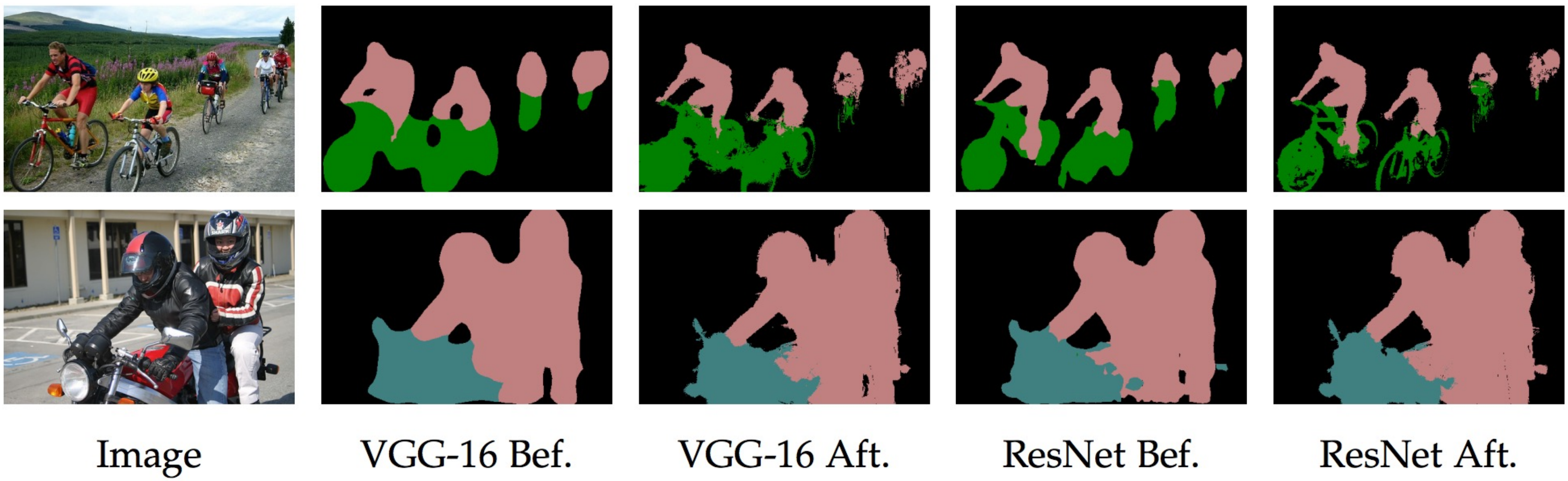
Conditional Random Field

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

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

Source: B. Hariharan

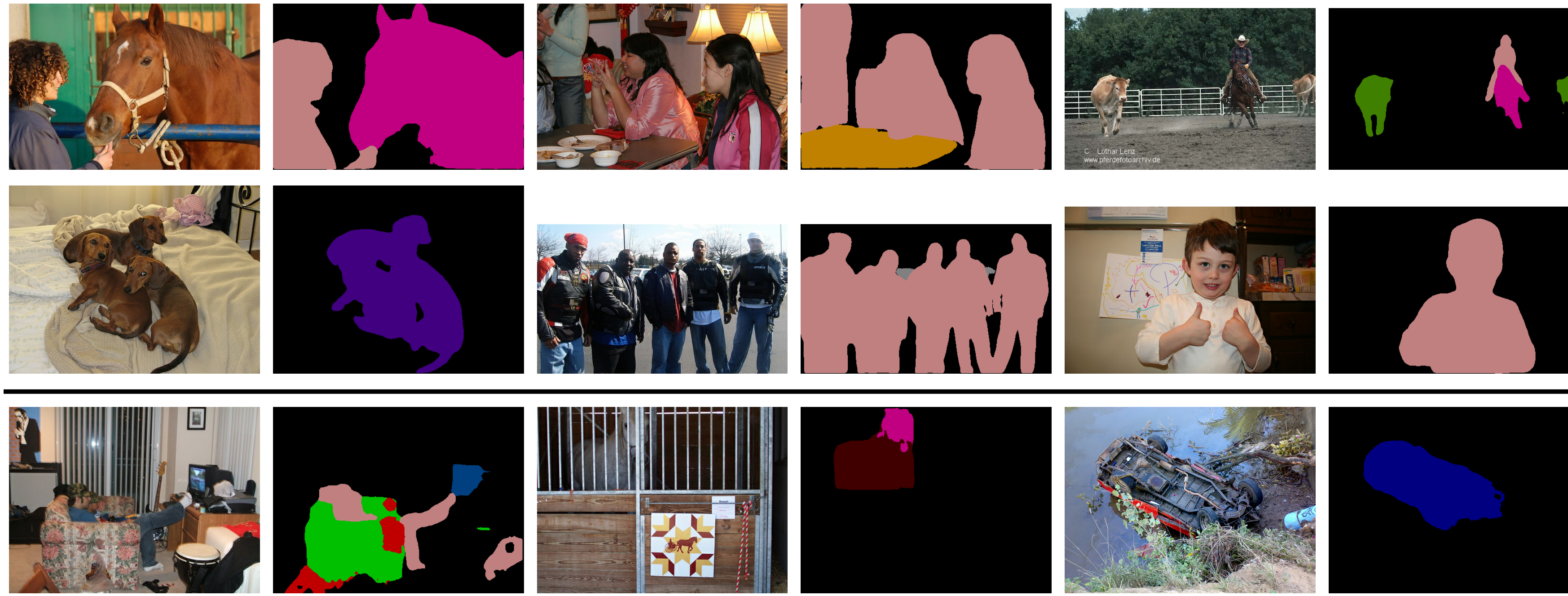
Fix 3: Use local edge information (CRFs)



Largely unnecessary given modern networks

Source: B. Hariharan

Semantic Segmentation Results



Method	mIOU
Deep Layer Cascade (LC) [82]	82.7
TuSimple [77]	83.1
Large_Kernel_Matters [60]	83.6
Multipath-RefineNet [58]	84.2
ResNet-38_MS_COCO [83]	84.9
PSPNet [24]	85.4
IDW-CNN [84]	86.3
CASIA_IVA_SDN [63]	86.6
DIS [85]	86.8
DeepLabv3 [23]	85.7
DeepLabv3-JFT [23]	86.9
DeepLabv3+ (Xception)	87.8
DeepLabv3+ (Xception-JFT)	89.0

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, Hartwig Adam, DeepLabv3+: Encoder-Decoder with Atrous Separable Convolution, ECCV 2018

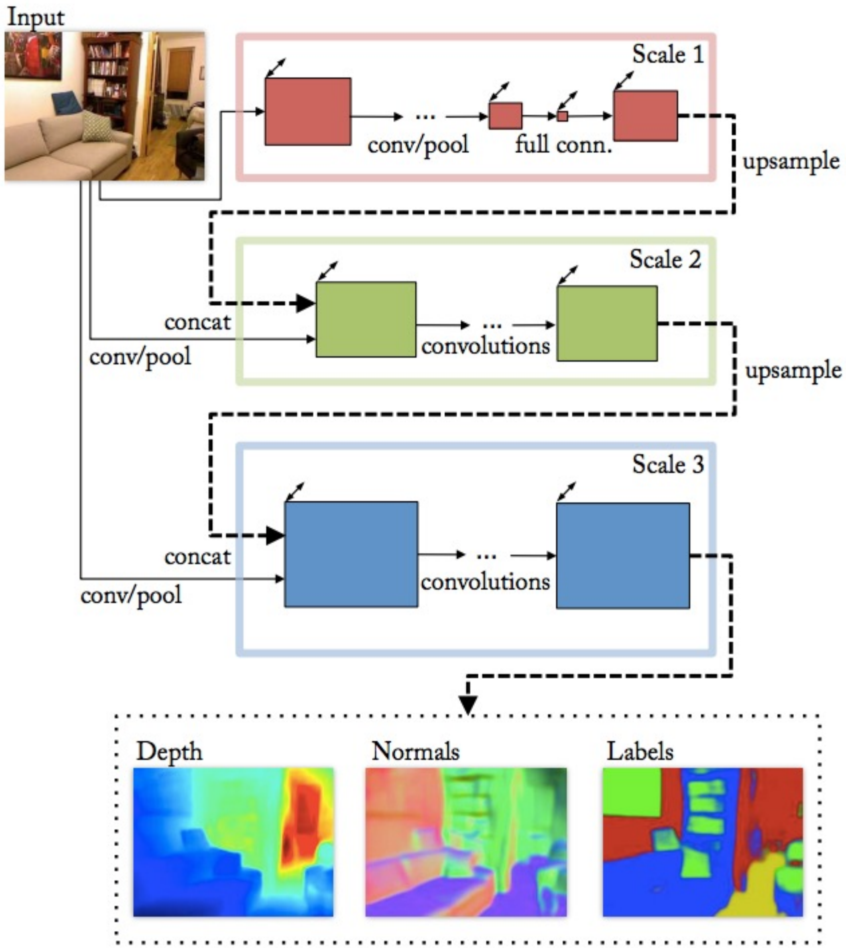
Outline

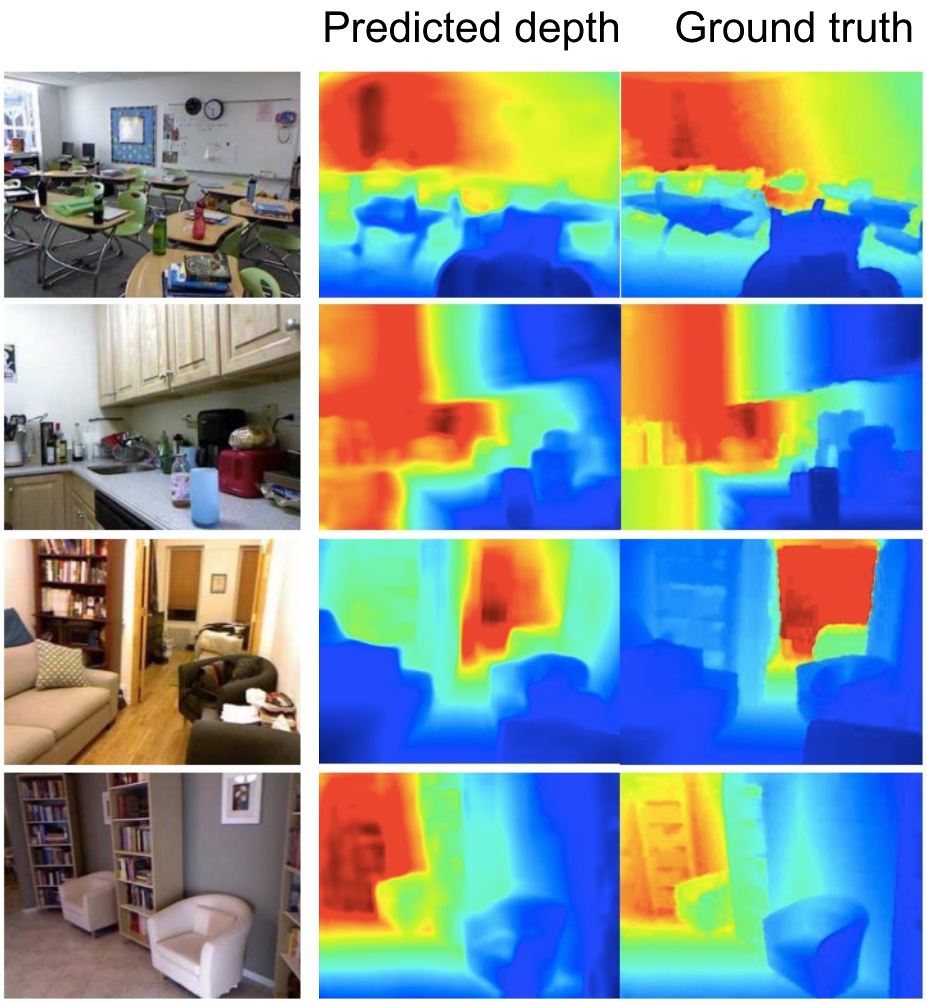
- Semantic segmentation
  - Architectures
    - “Convolutionalization”
    - Dilated convolutions
    - Hyper-columns / skip-connections
    - Learned up-sampling architectures
- Other dense prediction problems

Other dense prediction tasks

- Depth estimation
- Surface normal estimation
- Colorization
- ….

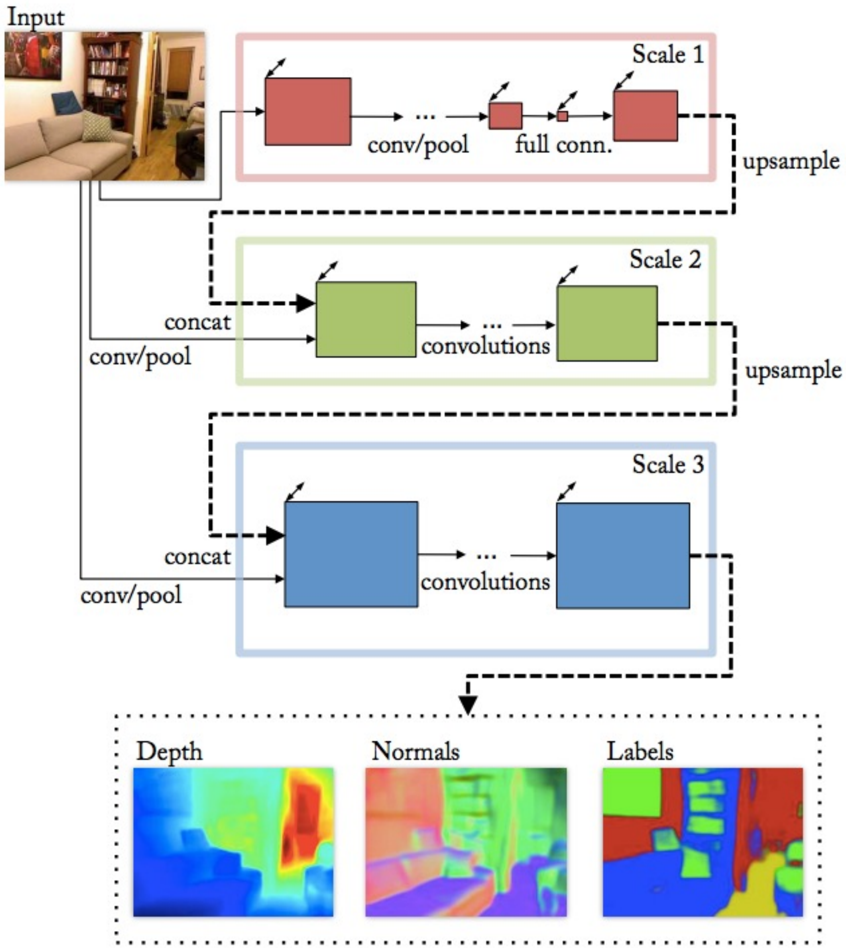
Depth and normal estimation



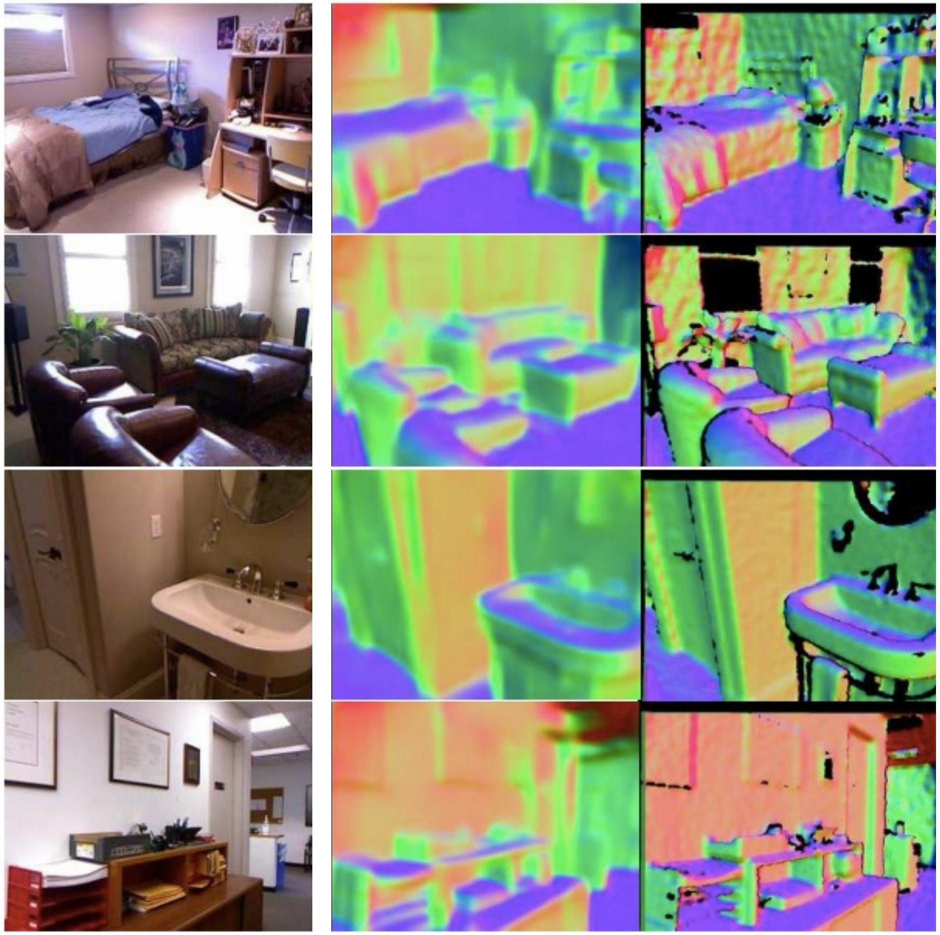


D. Eigen and R. Fergus, [Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture](), ICCV 2015

Depth and normal estimation

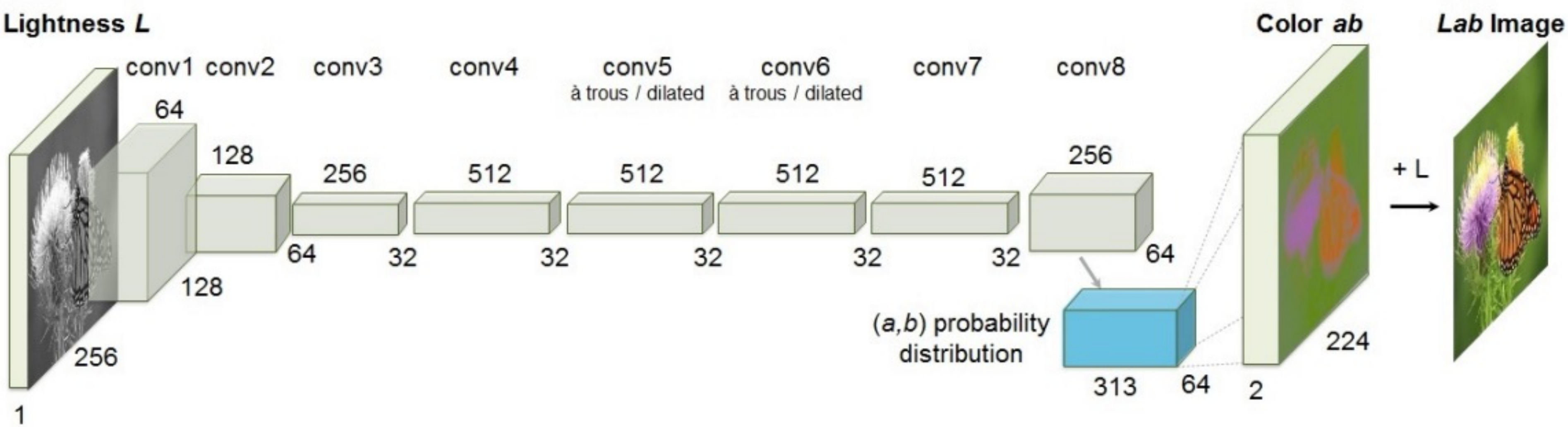


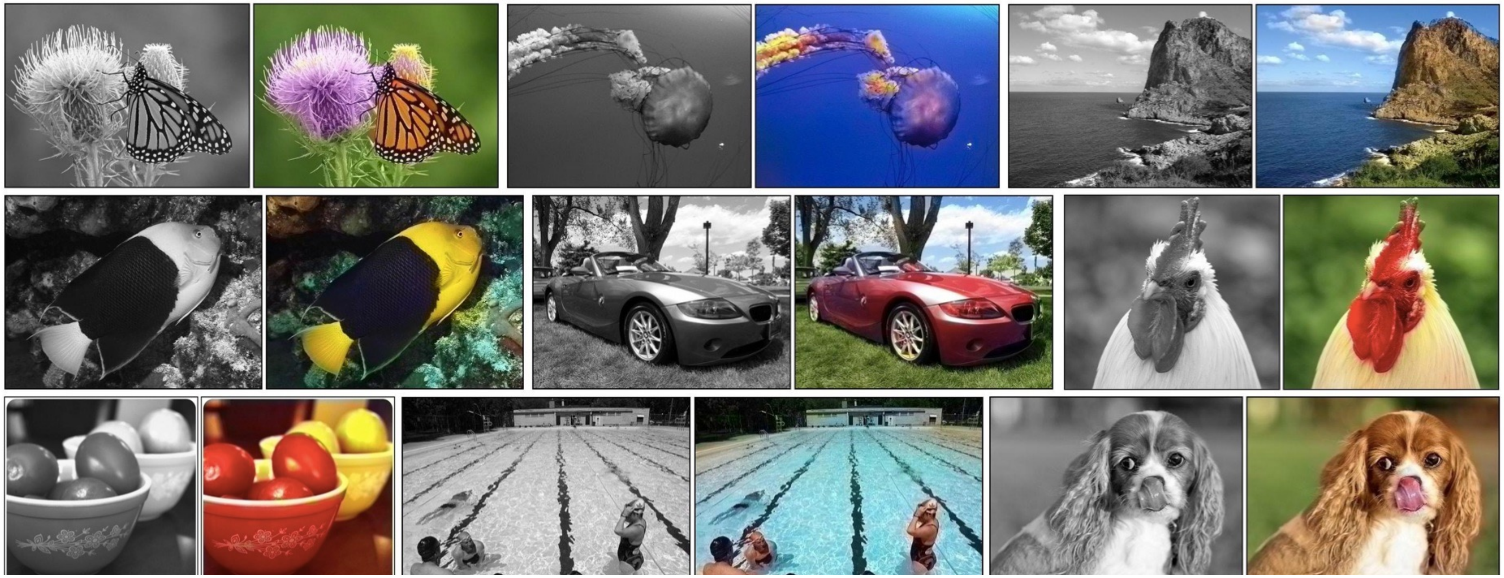
Predicted normals Ground truth



D. Eigen and R. Fergus, Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture, ICCV 2015

Colorization





R. Zhang, P. Isola, and A. Efros, [Colorful Image Colorization](), ECCV 2016