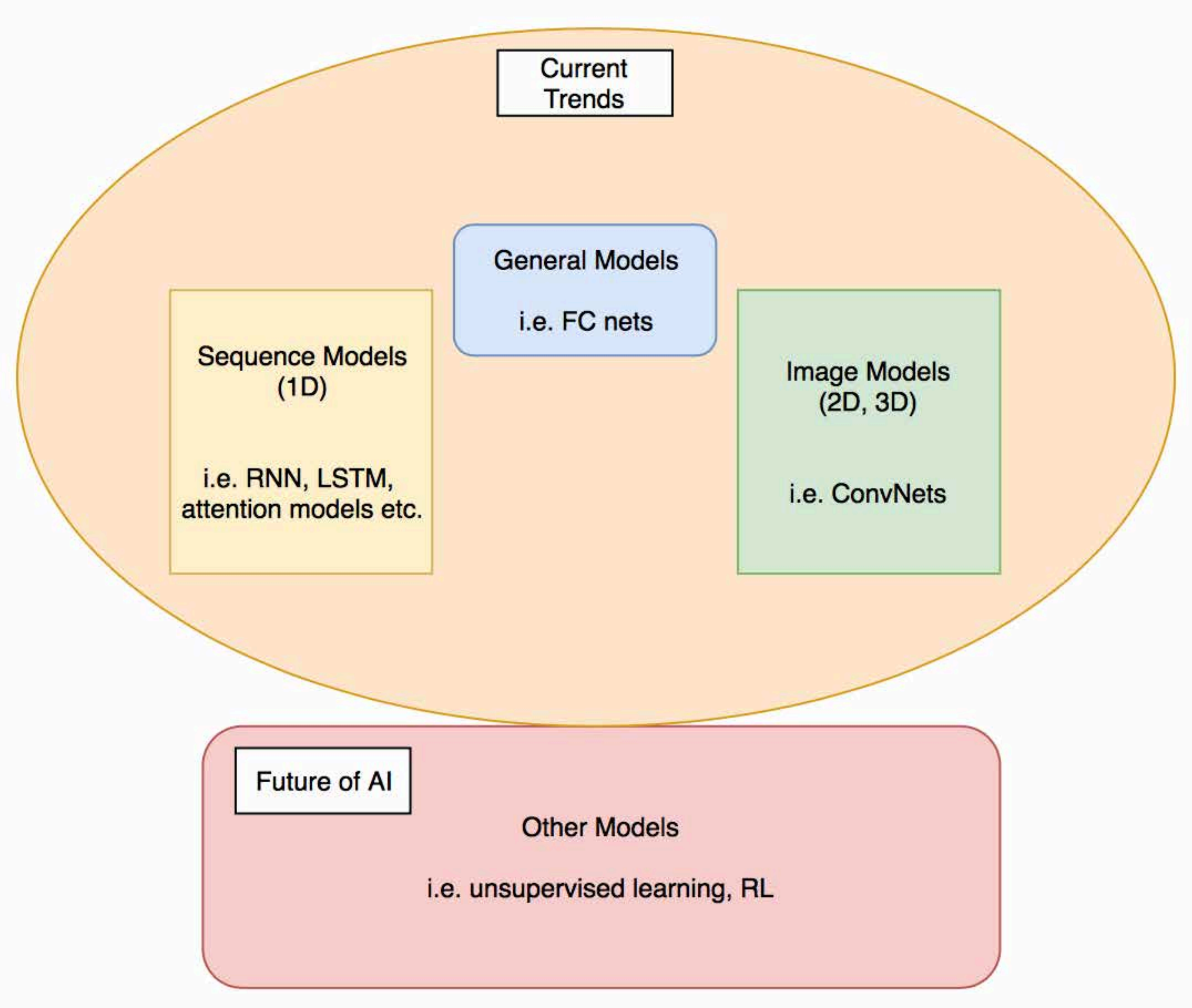
CS 466/566
Introduction to Deep Learning

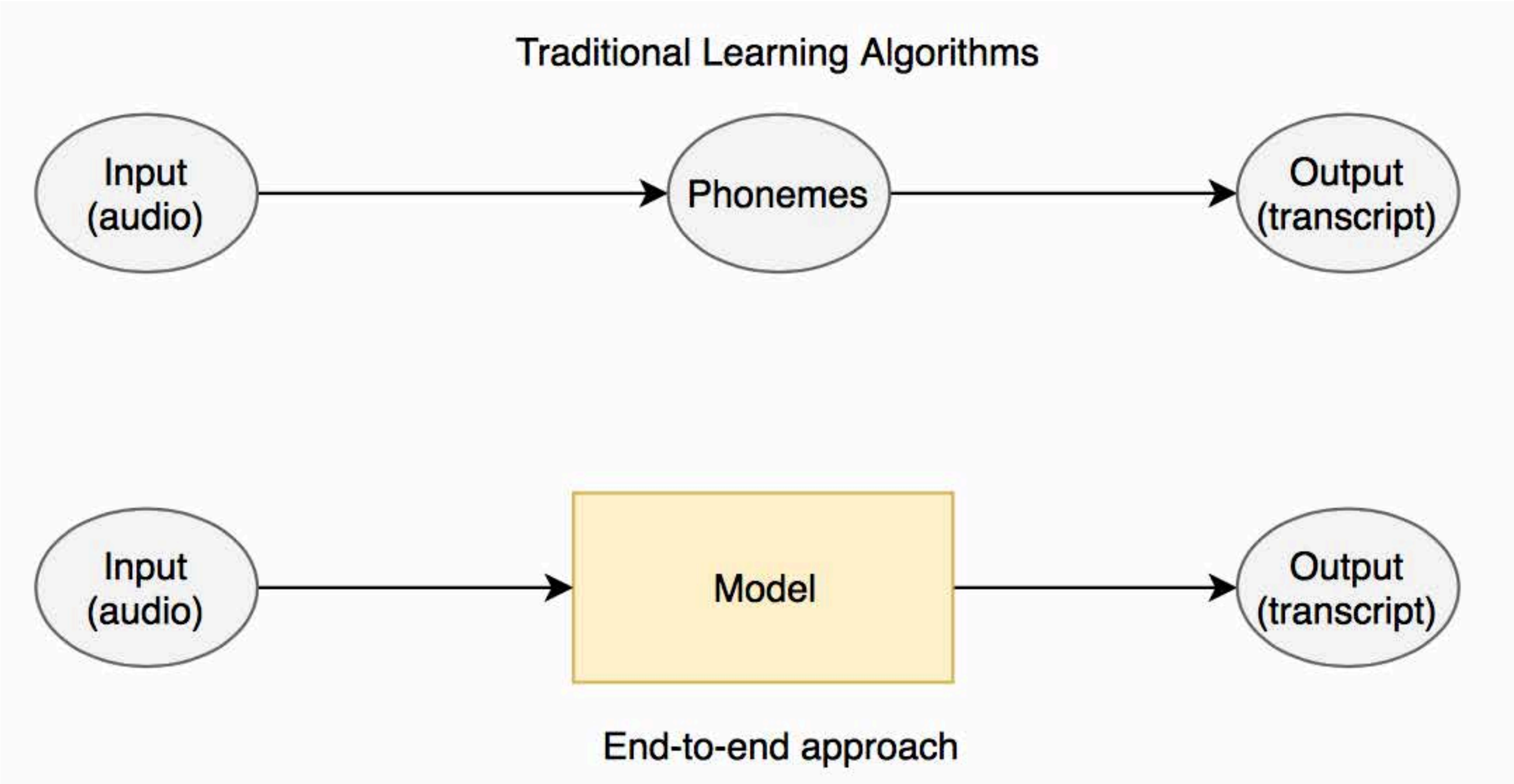
Lecture 10 – Fully Convolutional Networks

Compiled from various resources

What were we doing? Where are we?



End to End Training?



Teaser #1:

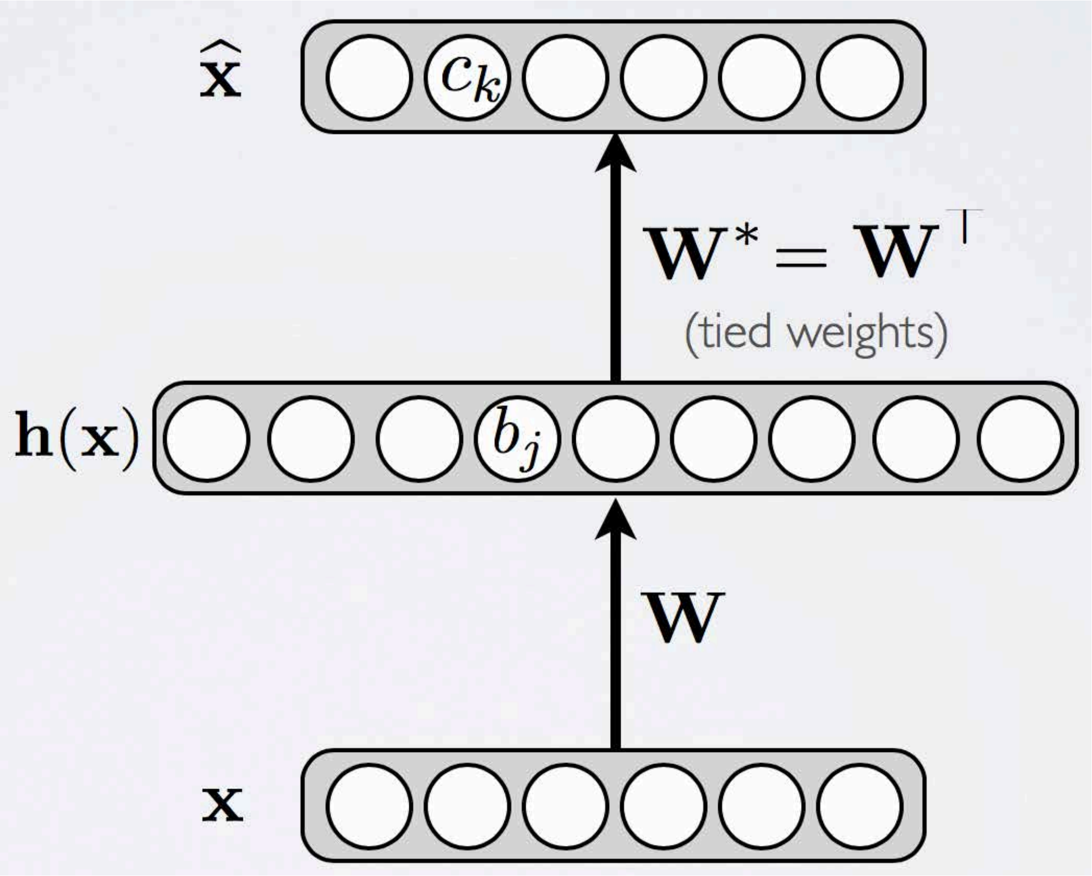
Sparse Auto-encoders

Family of Sparse Auto-encoders

- The simple Auto-Encoder targets to compress information of the given data as keeping the reconstruction cost lower as much as possible.
- However another use is to enlarge the given input's representation. In that case, you learn over-complete representation of the given data instead of compressing it.
- Most common implication is Sparse Auto-Encoder that learns over-complete representation but in a sparse (smart) manner.
- That means, for a given instance only informative set of units are activated, therefore you are able to capture more discriminative representation, especially if you use AE for pre-training of your deep neural network.

Denoising Auto-encoder with Over-complete Hidden Layer

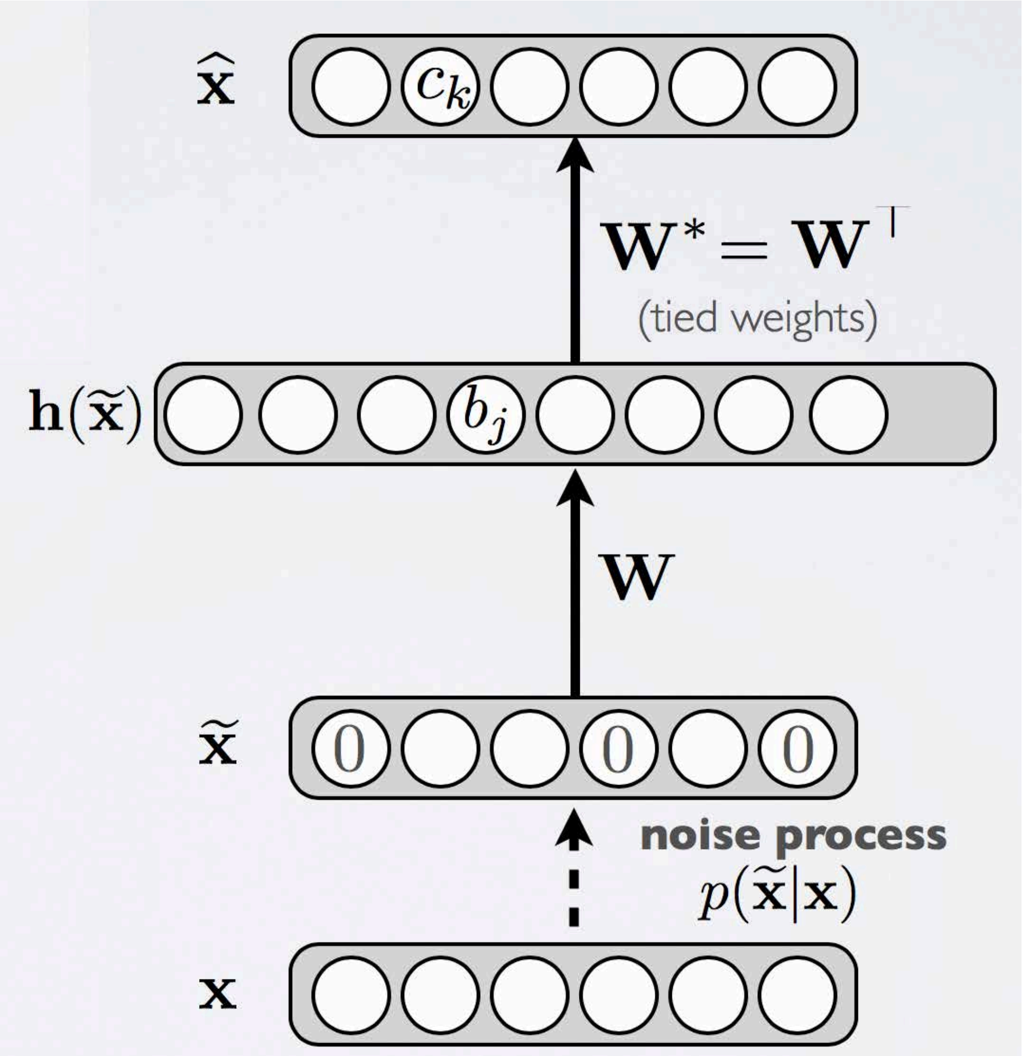
- Hidden layer is over-complete if greater than the input layer
  - No compression in the hidden layer
  - Each hidden unit could copy an input unit
- No guarantee that the hidden units will extract meaningful structure!
  - That's why we do corrupt input



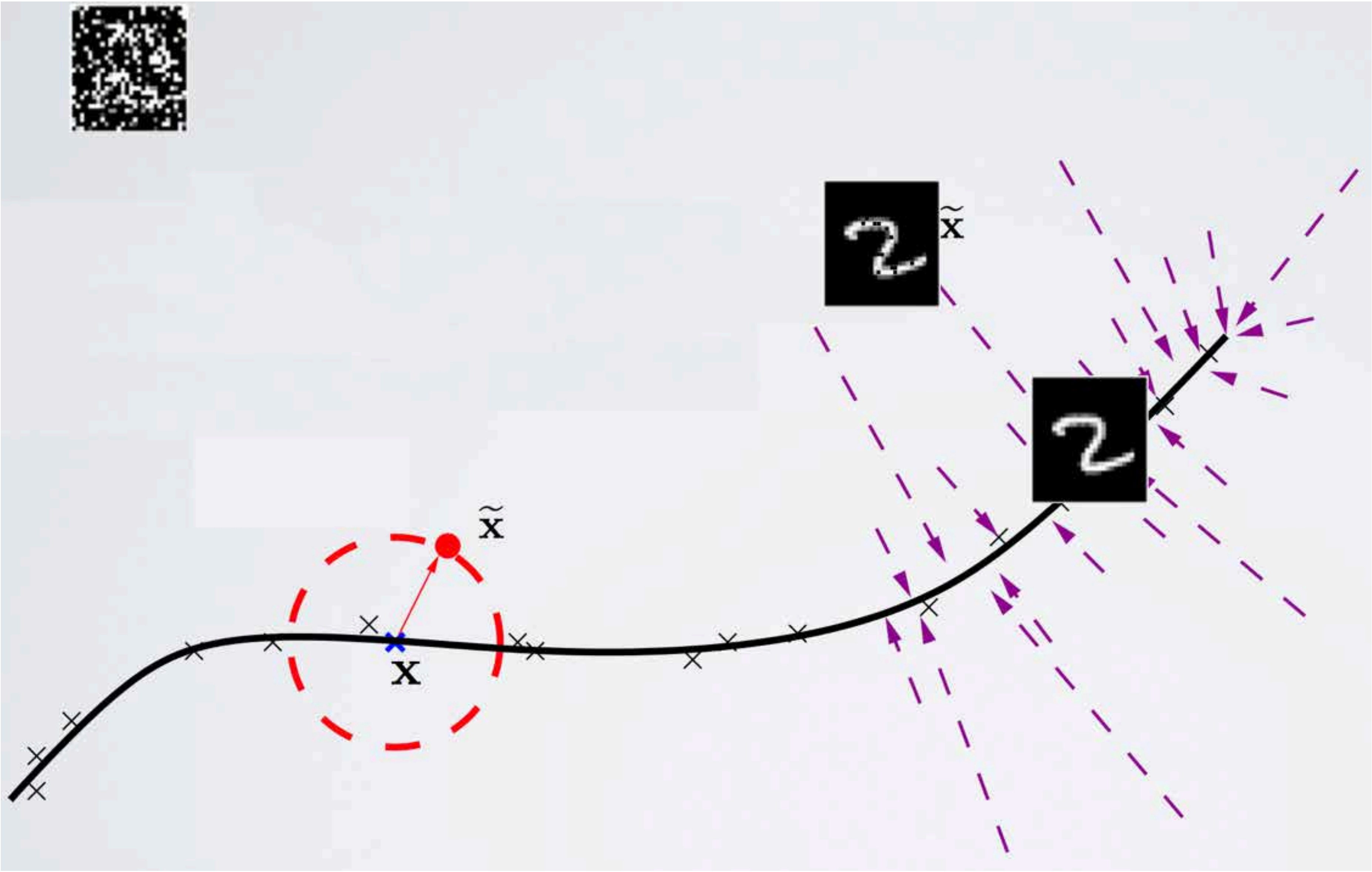
Denoising Auto-encoder with Over-complete Hidden Layer

- Loss function compares the output with the noiseless input vector.

- Can use
  - Random assignment of subset of inputs to 0, with a probability *v*.
  - Gaussian additive noise
  - etc…

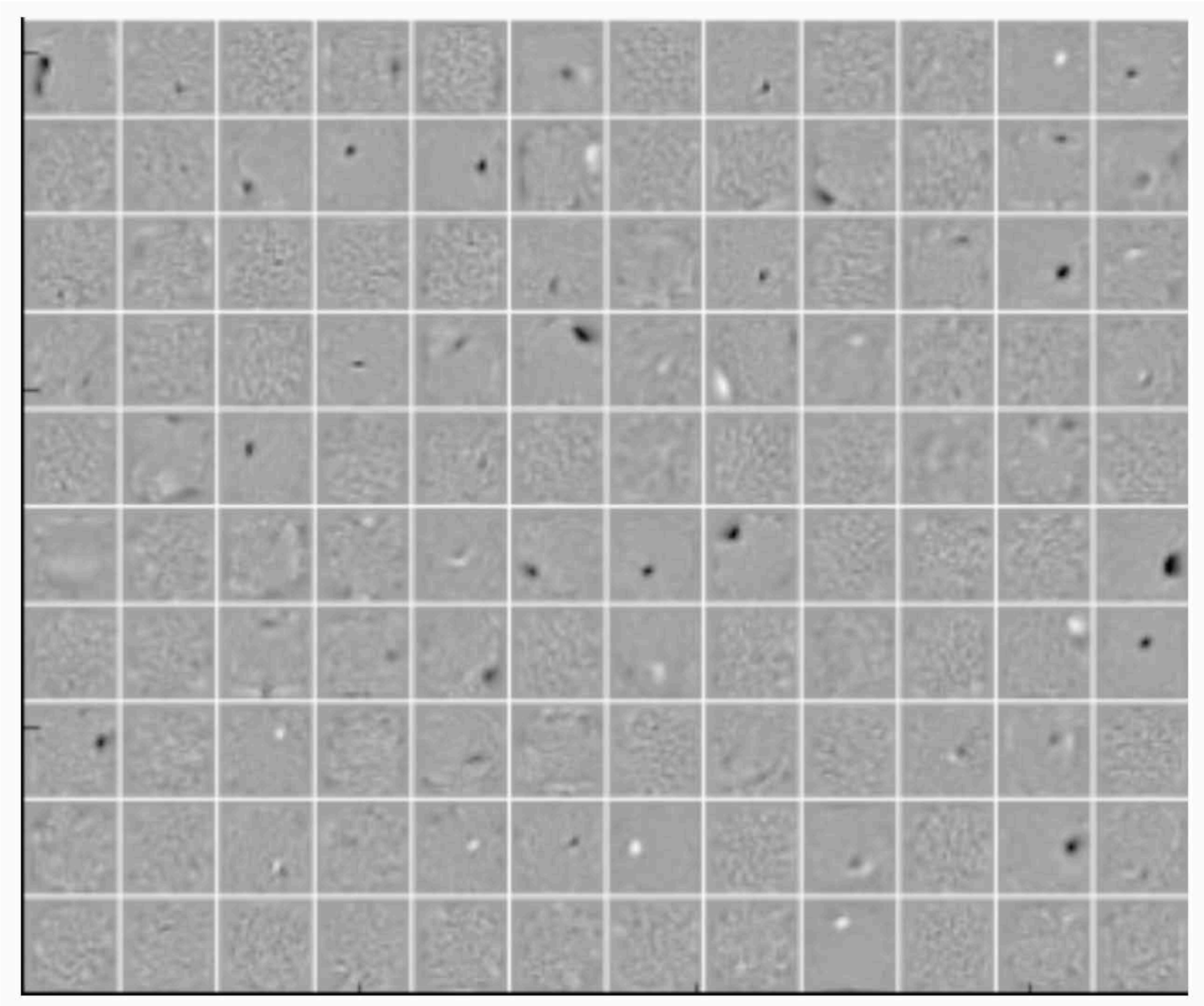


Denoising Auto-encoder with Over-complete Hidden Layer



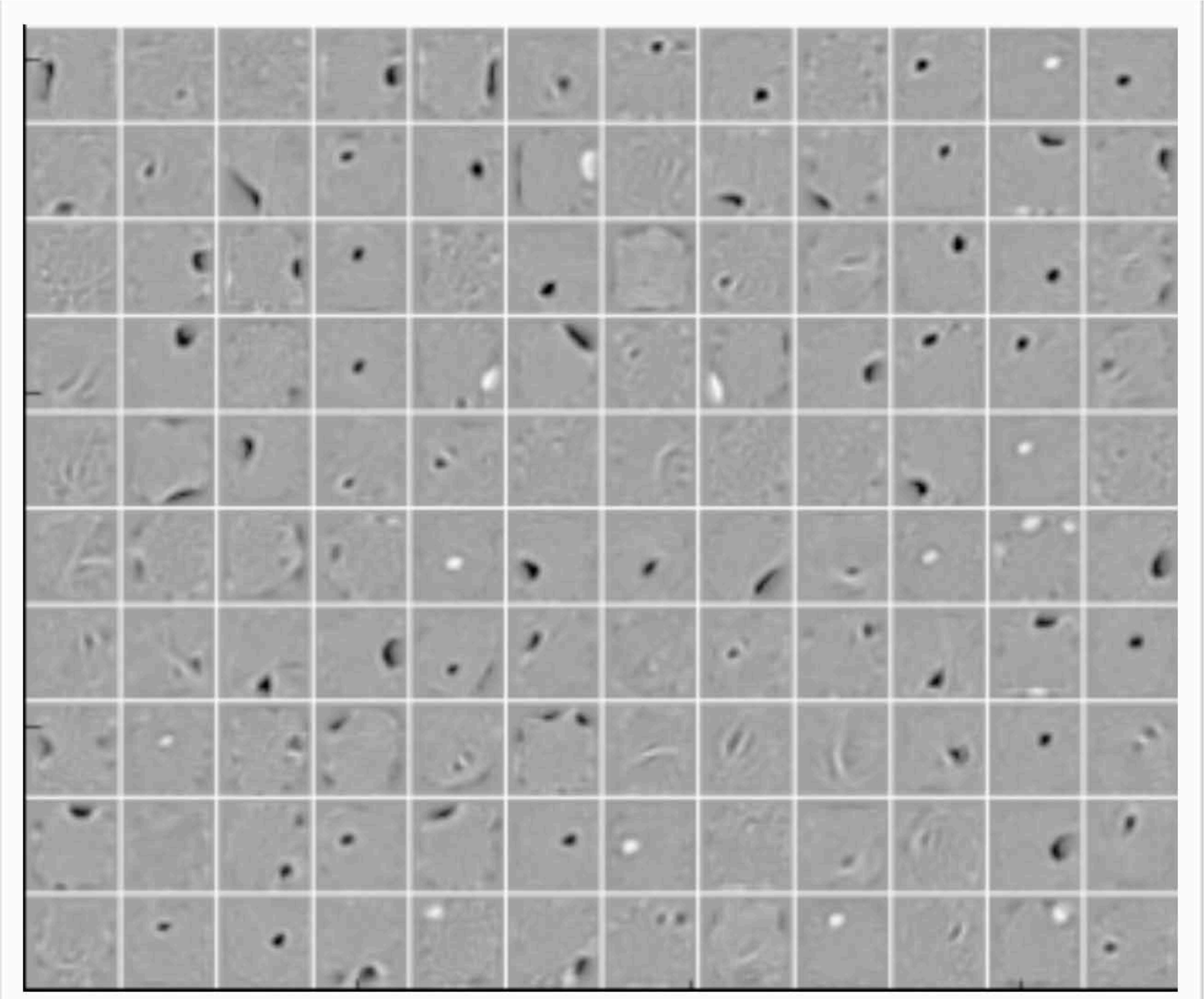
Denoising Auto-encoder with Over-complete Hidden Layer

No corruption



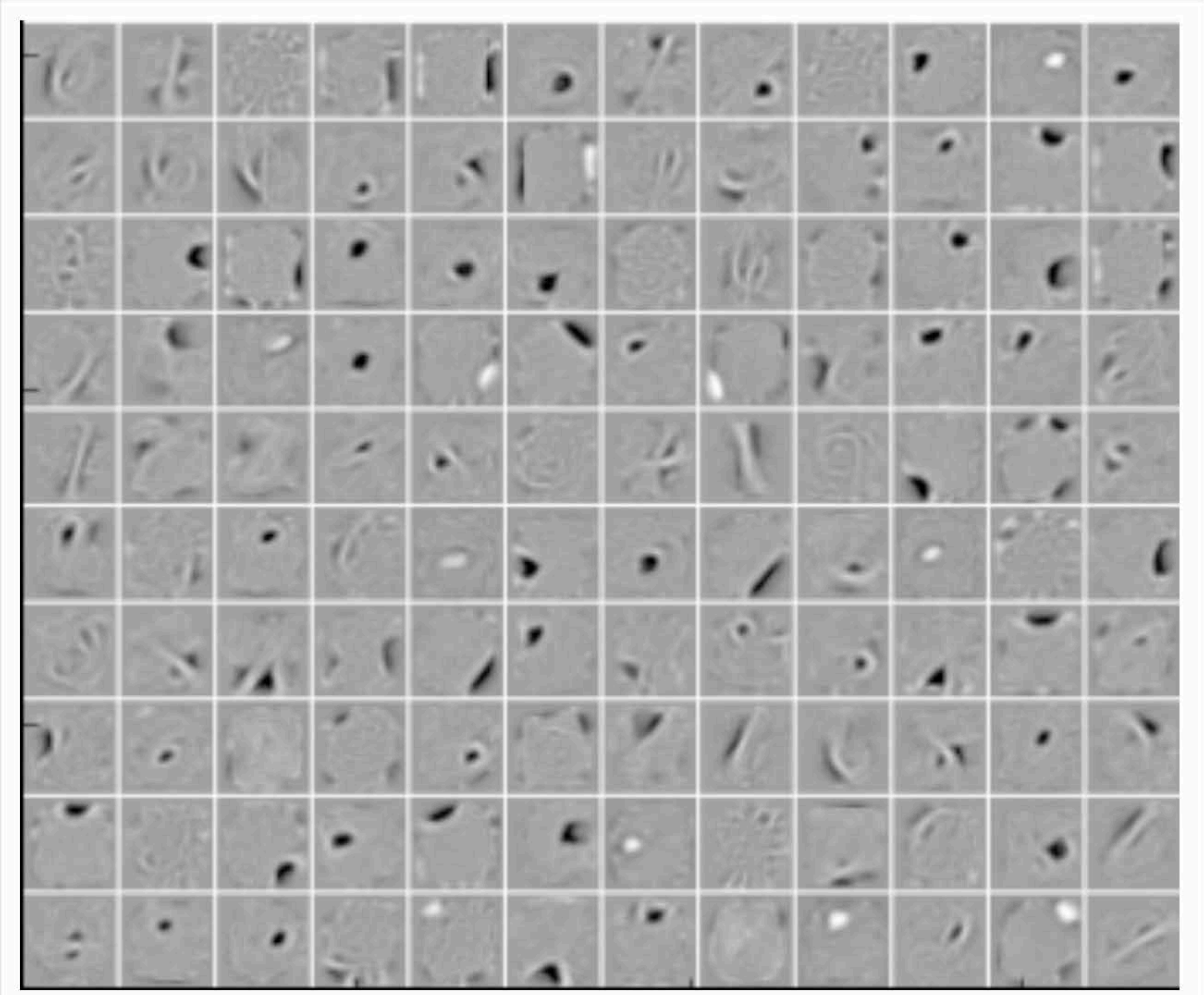
Denoising Auto-encoder with Over-complete Hidden Layer

25% corruption



Denoising Auto-encoder with Over-complete Hidden Layer

50% corruption



Semantic Segmentation

Google Maps: Road finder. How to do?

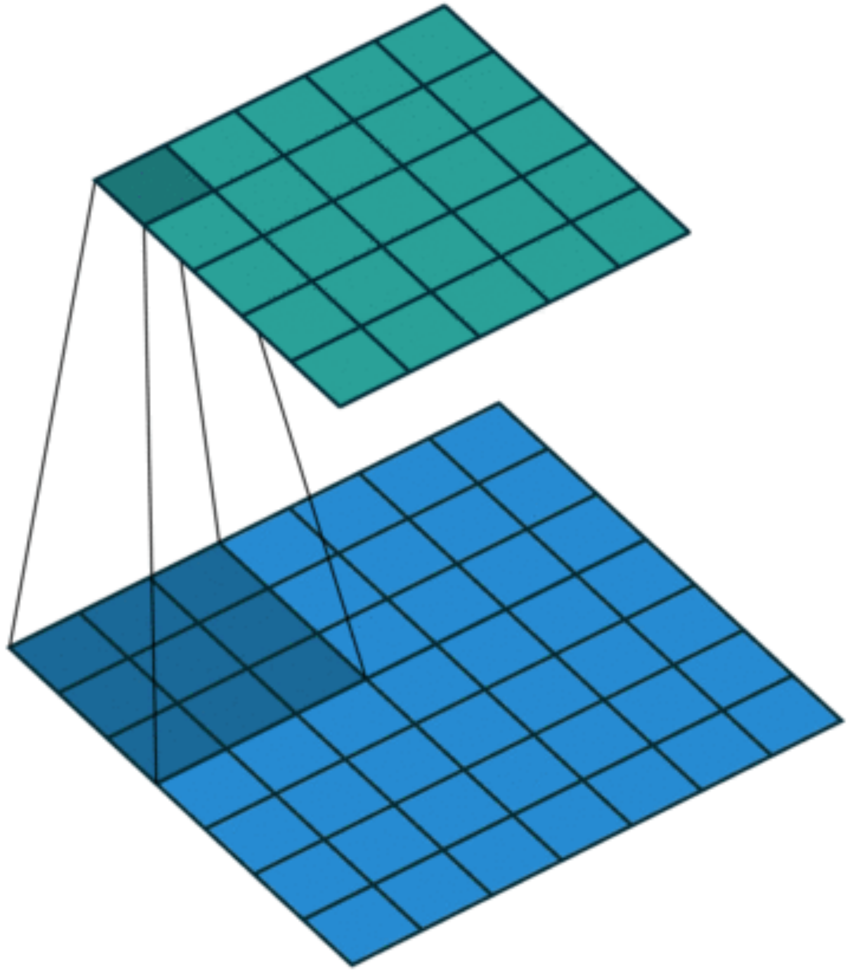
Finding roads in high-resolution images

- Vlad Mnih (ICML 2012) used a non-convolutional net with local fields and multiple layers of rectified linear units to find roads in cluttered aerial images.
  - It takes a large image patch and predicts a binary road label for the central 16x16 pixels.
  - There is lots of labeled training data available for this task.
- The task is hard for many reasons:
  - Occlusion by buildings, trees and cars.
  - Shadows, Lighting changes
  - Minor viewpoint changes
- The worst problems are incorrect labels:
  - Badly registered maps
  - Arbitrary decisions about what counts as a road.
- Big neural nets trained on big image patches with millions of examples were the only hope.

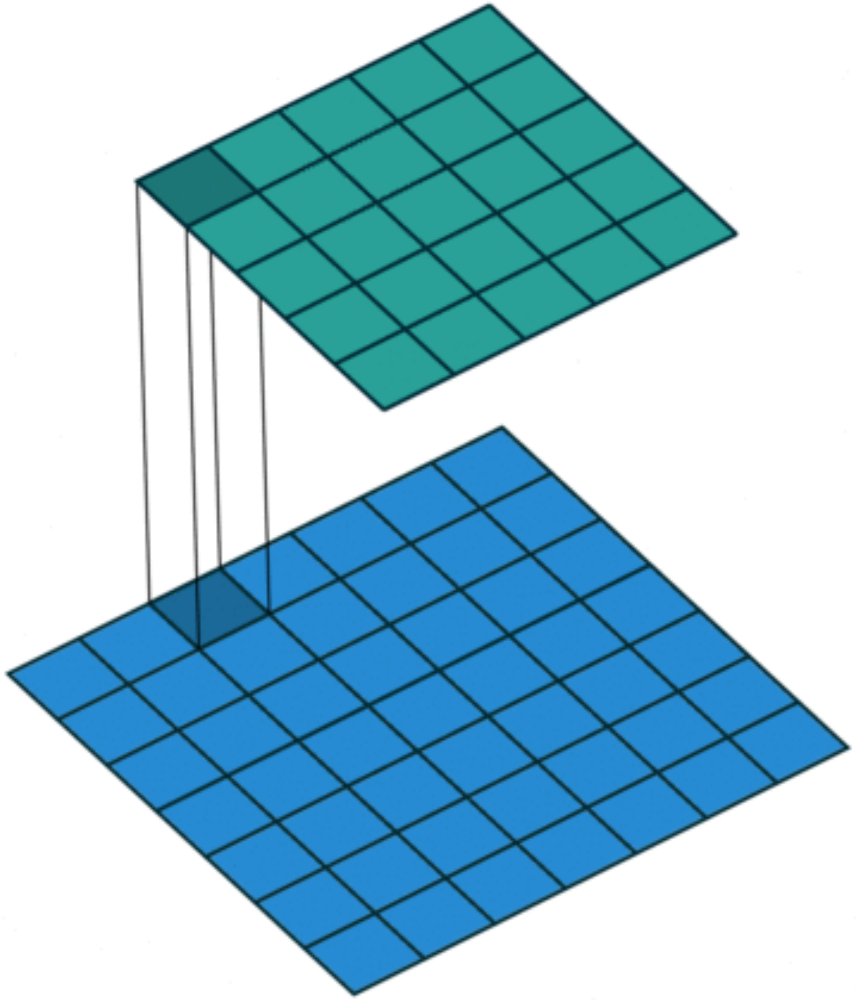
Road finder, how?

One by One [1 x 1] Convolution:

http://iamaaditya.github.io/2016/03/one-by-one-convolution/

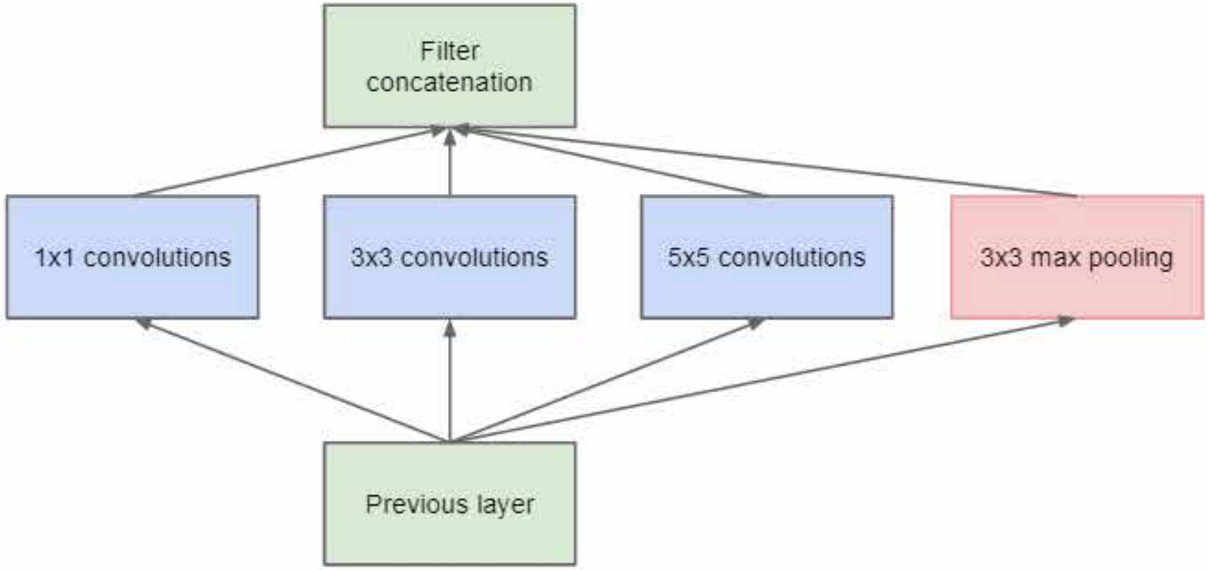


3x3 convolution

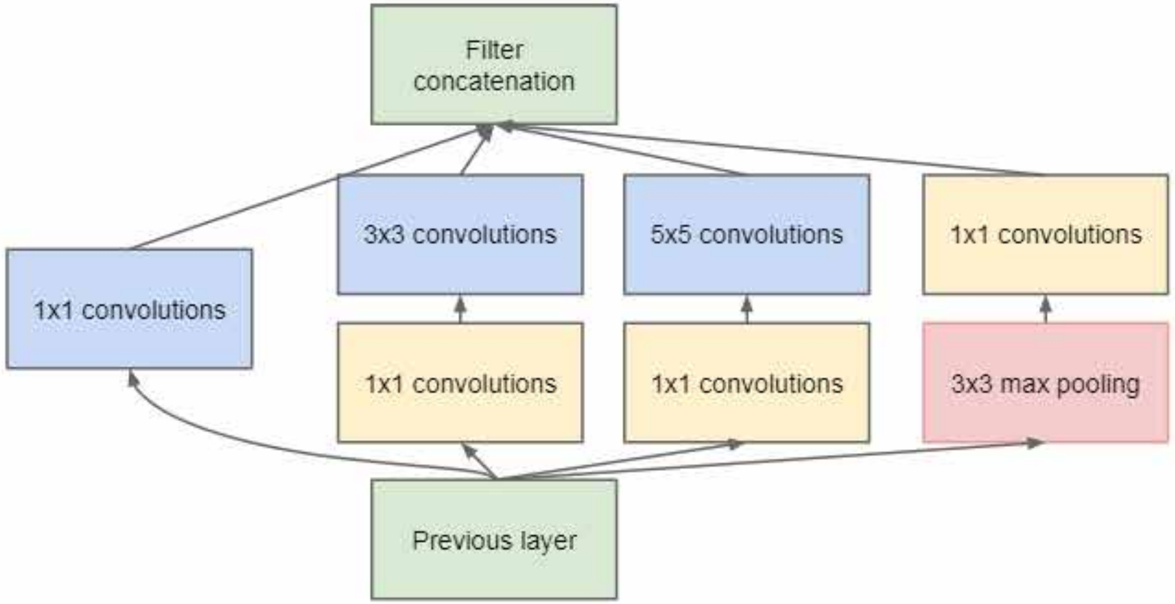


1x1 convolution

Recall: Inception Module

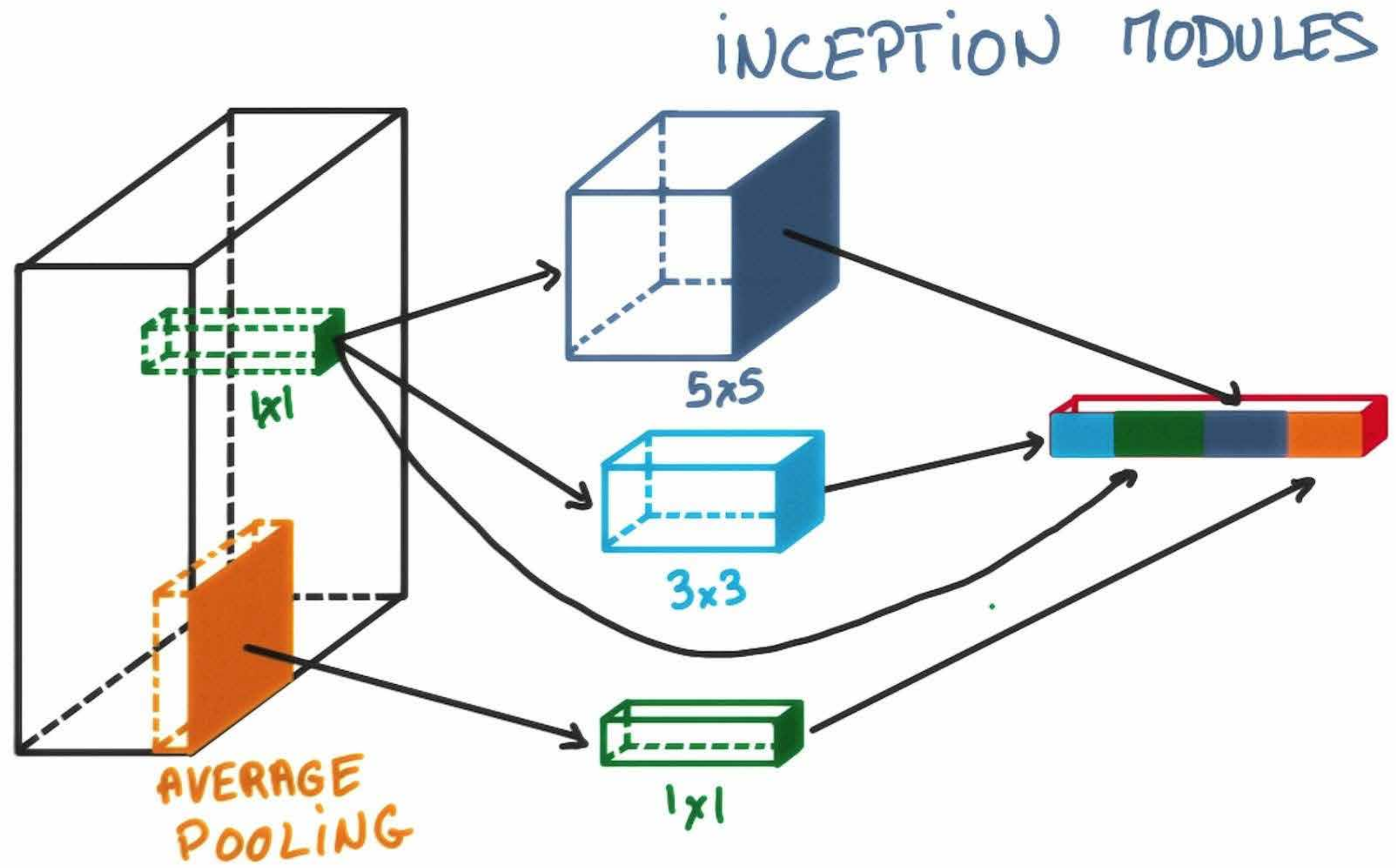


(a) Inception module, naïve version

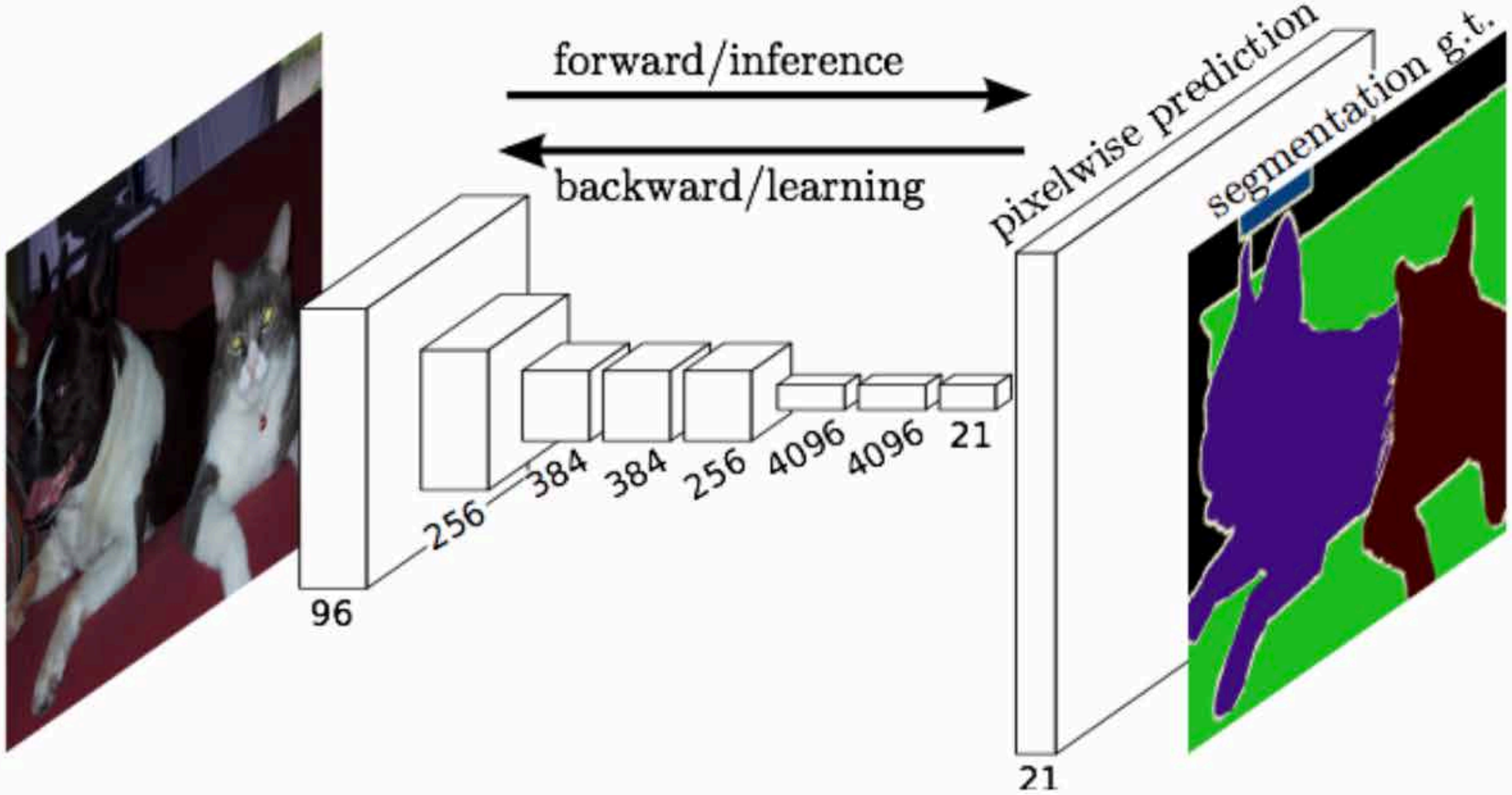


(b) Inception module with dimension reductions

Recall: Inception Module



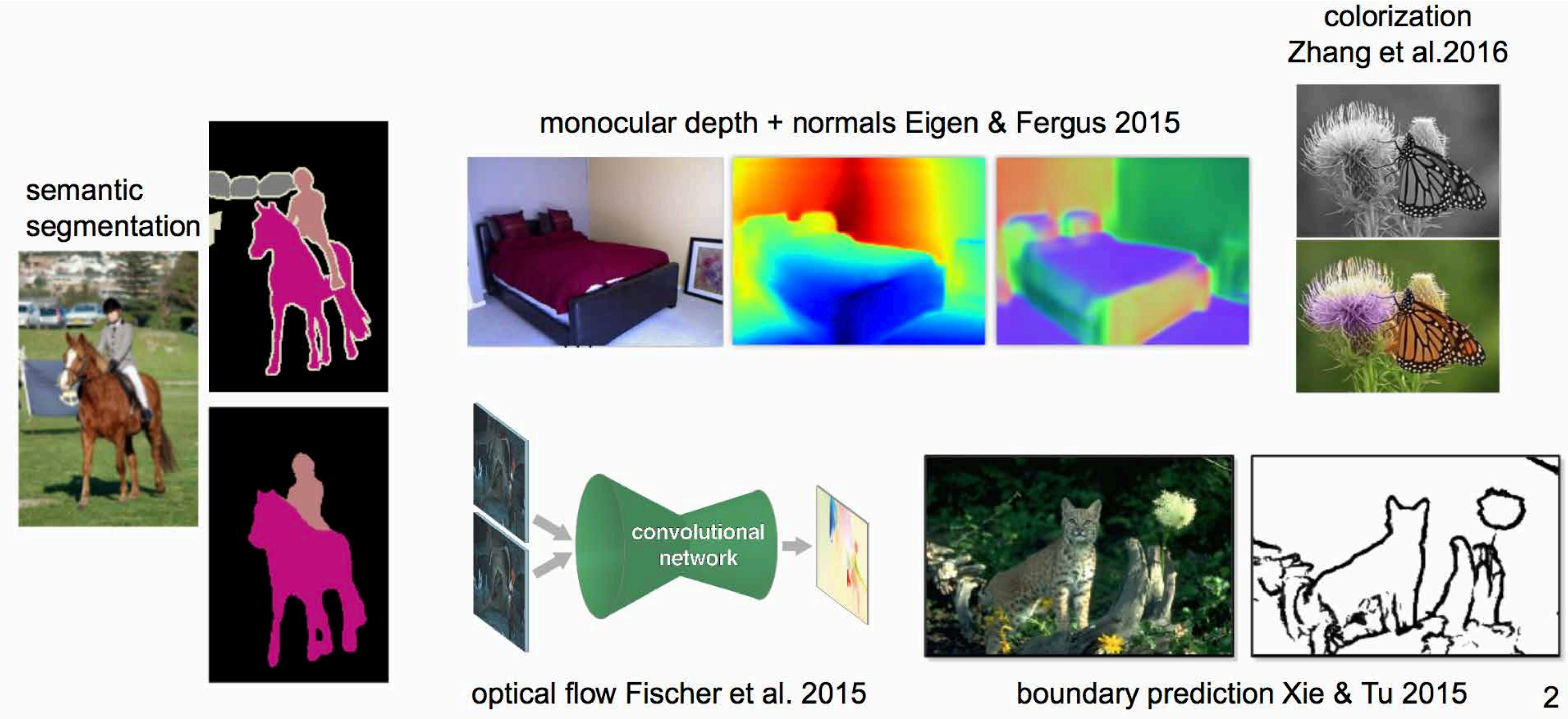
Fully Convolutional Networks (FCN)



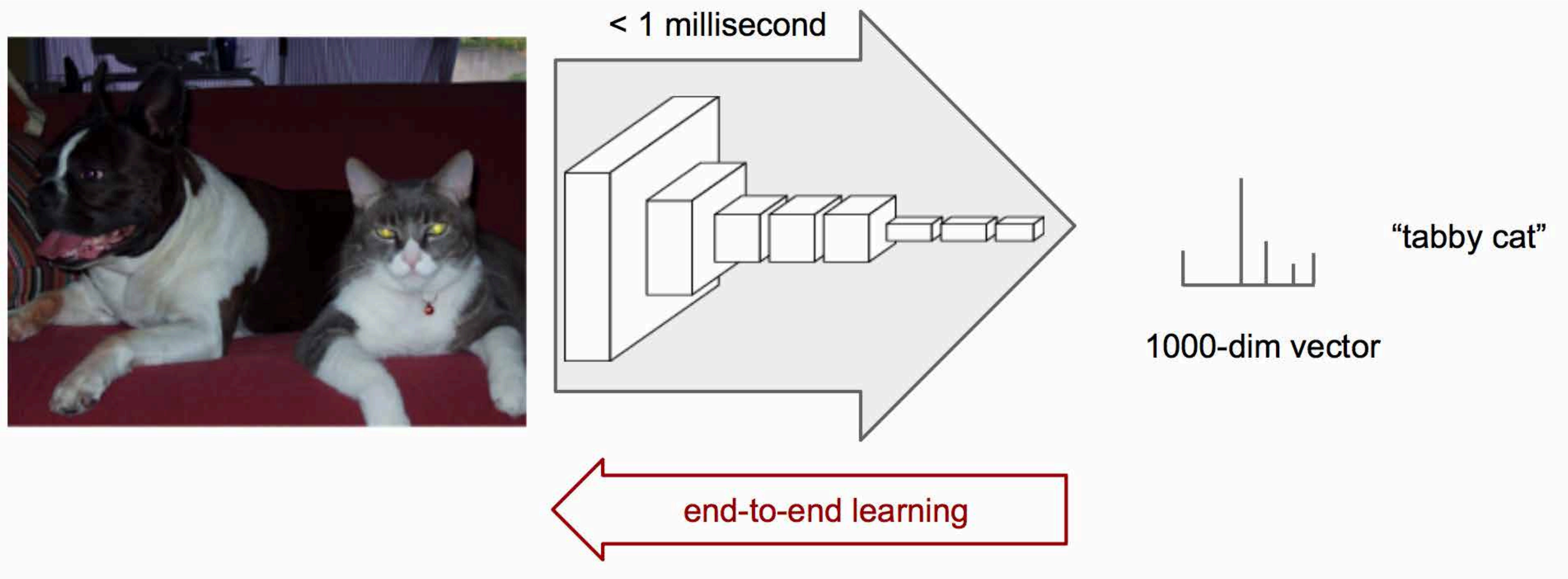
Evan Shelhamer* Jonathan Long* Trevor Darrell

UC Berkeley in CVPR'15, PAMI'16

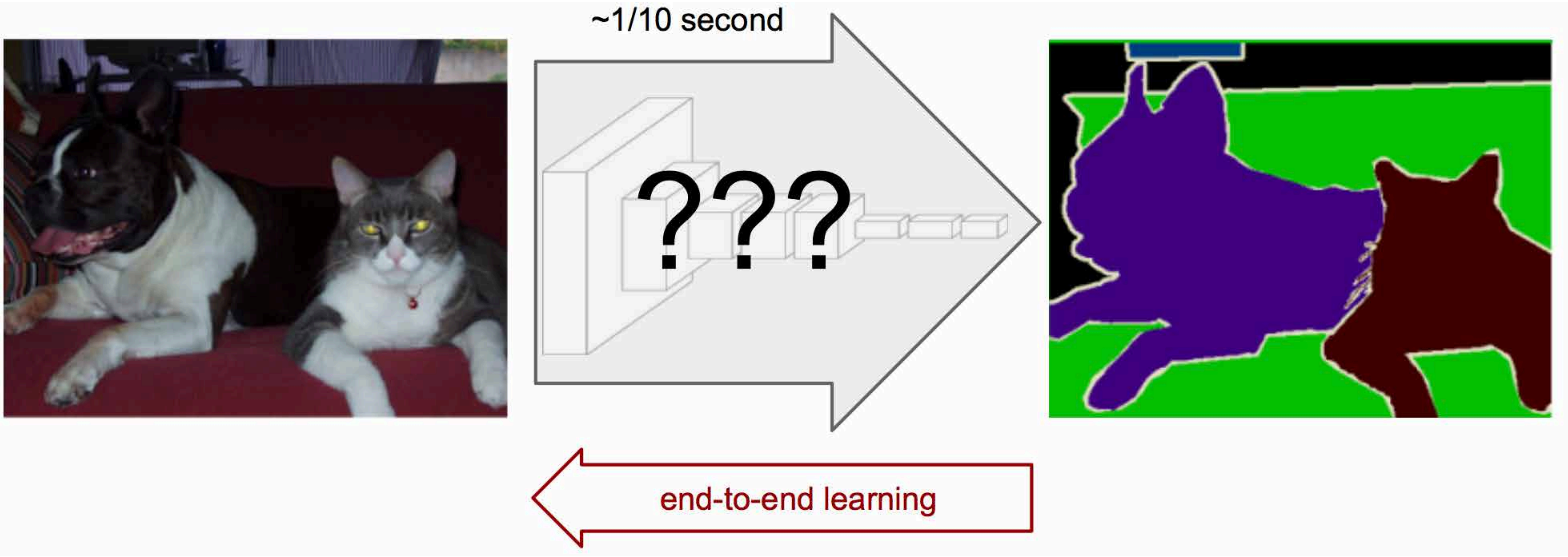
FCN: Pixels in, pixels out



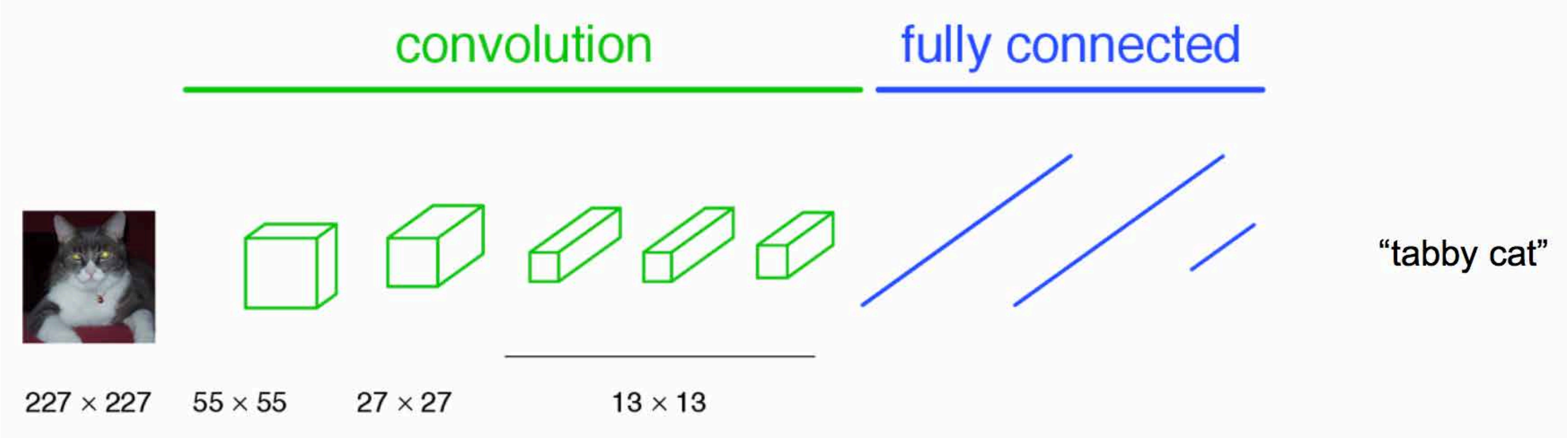
CNNs can do Classification



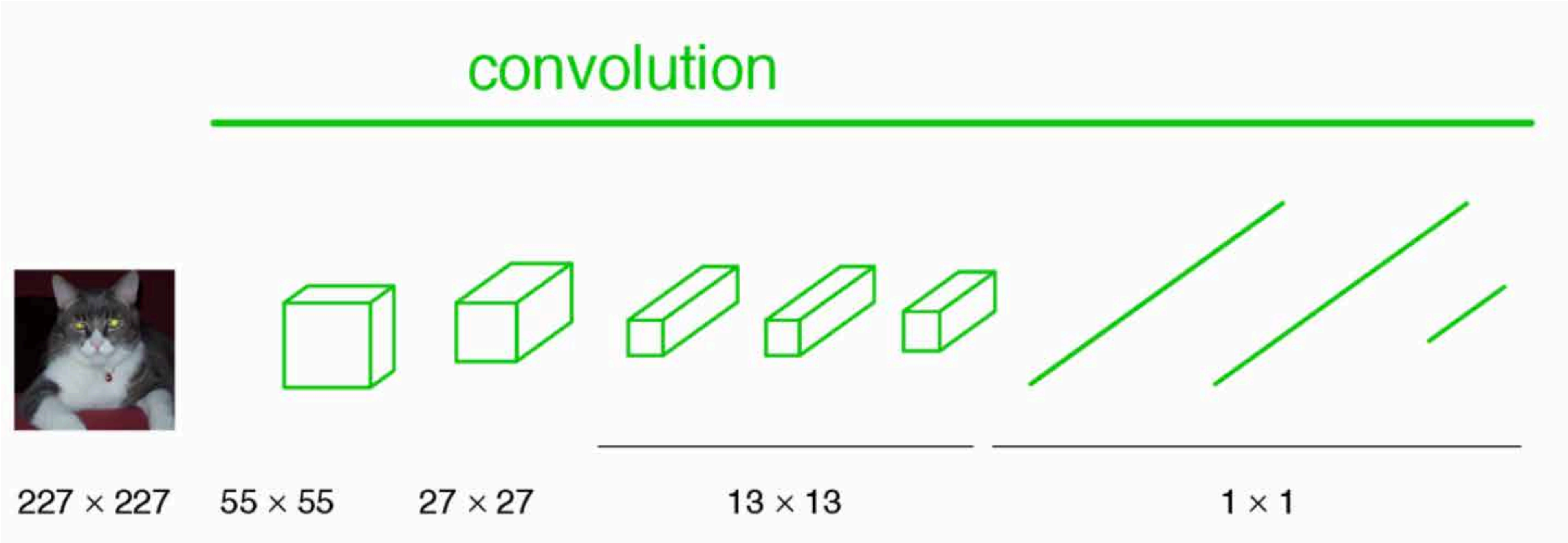
What about per-pixel classification?



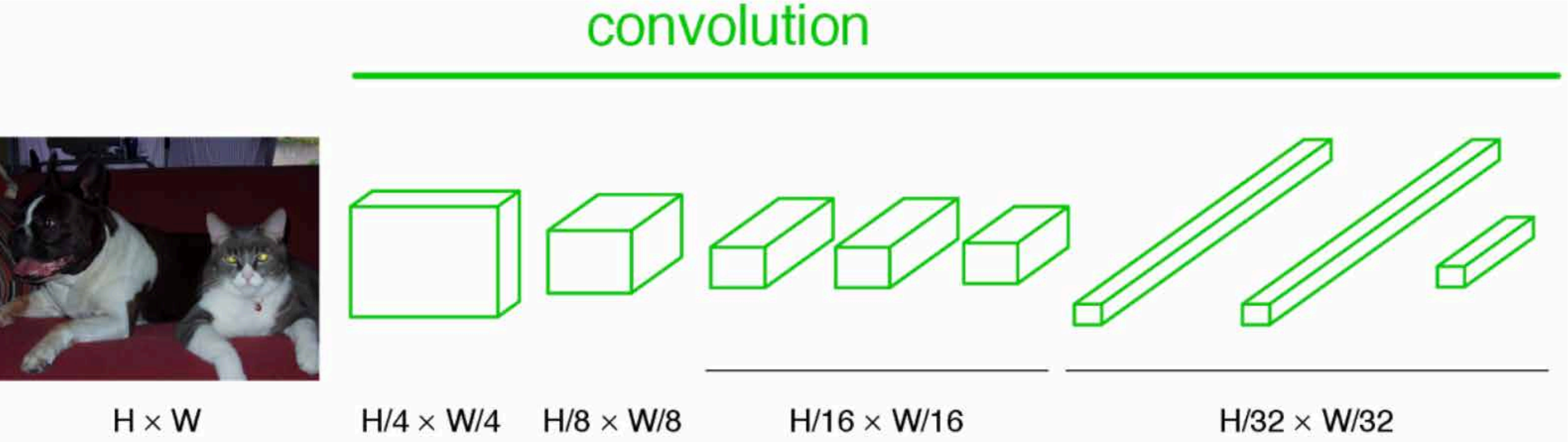
A classification ConvNet



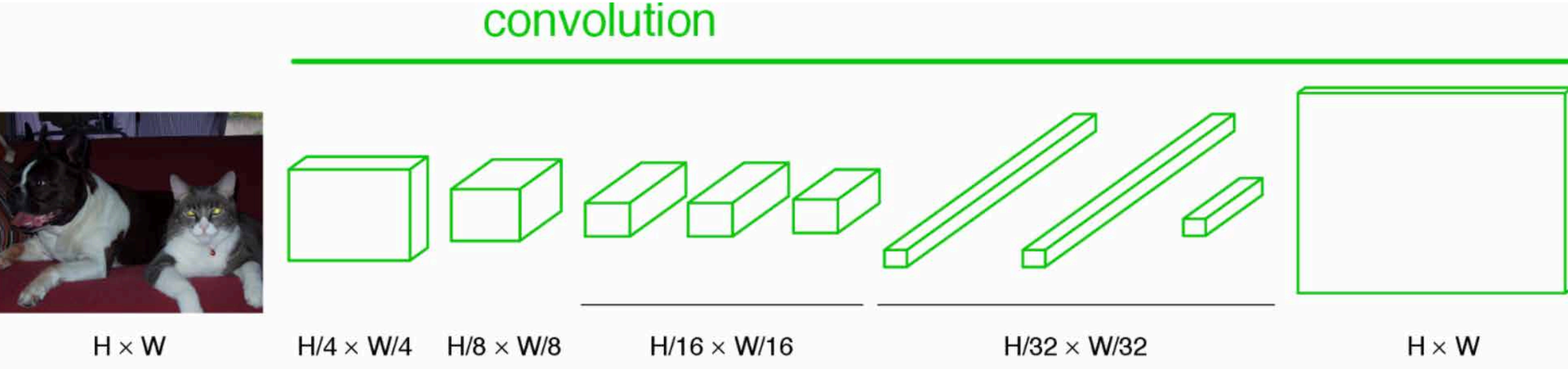
Becoming Fully Convolutional



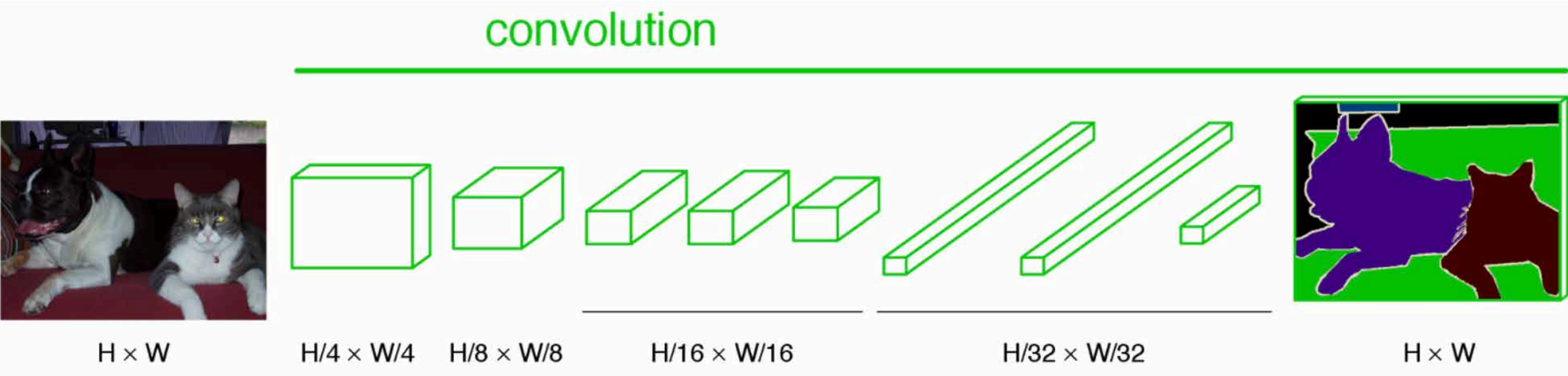
Becoming Fully Convolutional (arbitrary input)



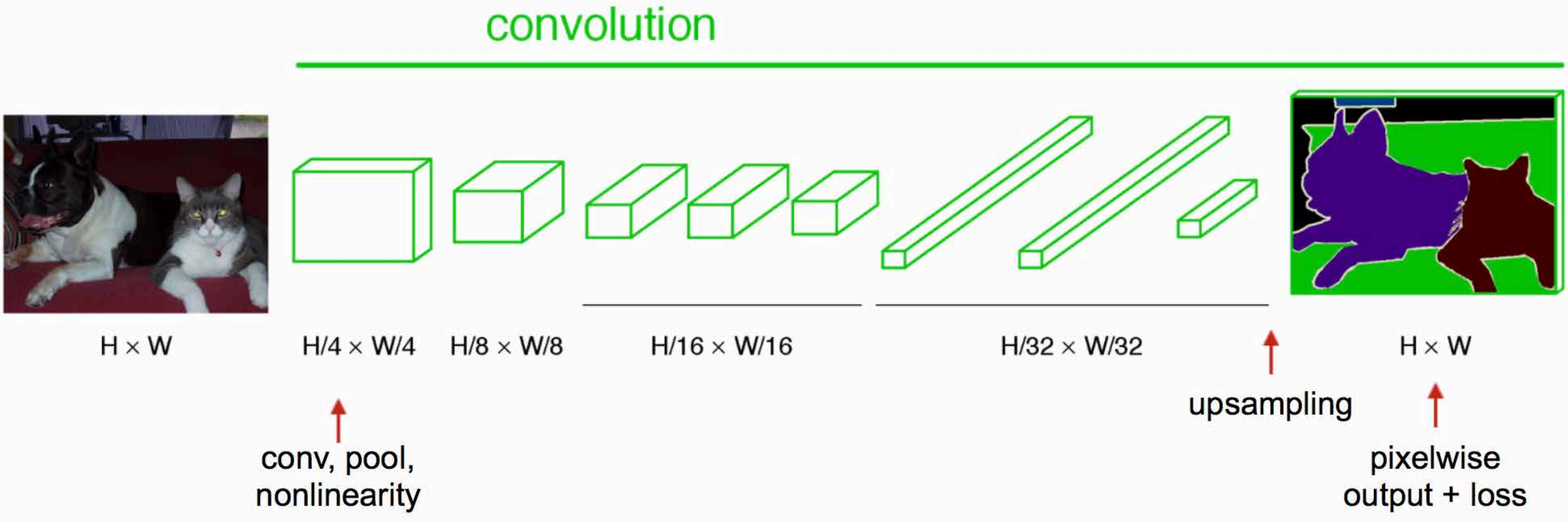
Upsampling Output



End-to-end, pixels-to-pixels network

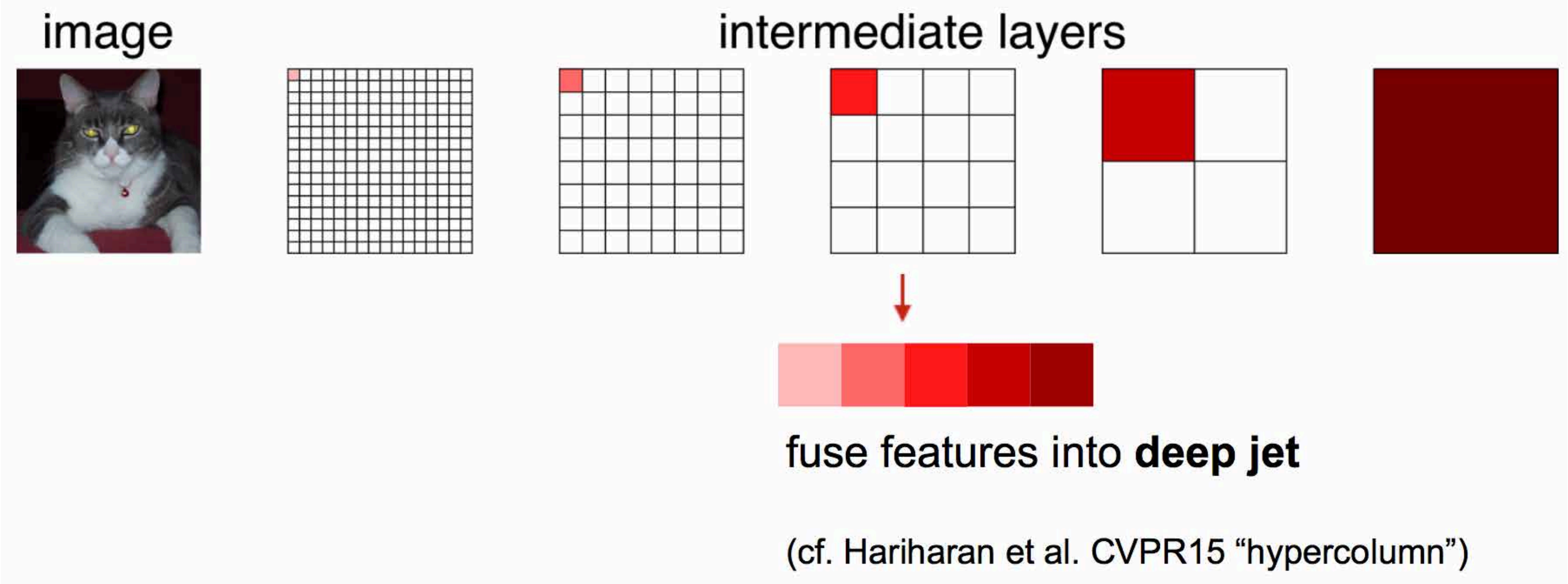


End-to-end, pixels-to-pixels network



Spectrum of Deep Features

combine *where* (local, shallow) with *what* (global, deep)



Skip Layers

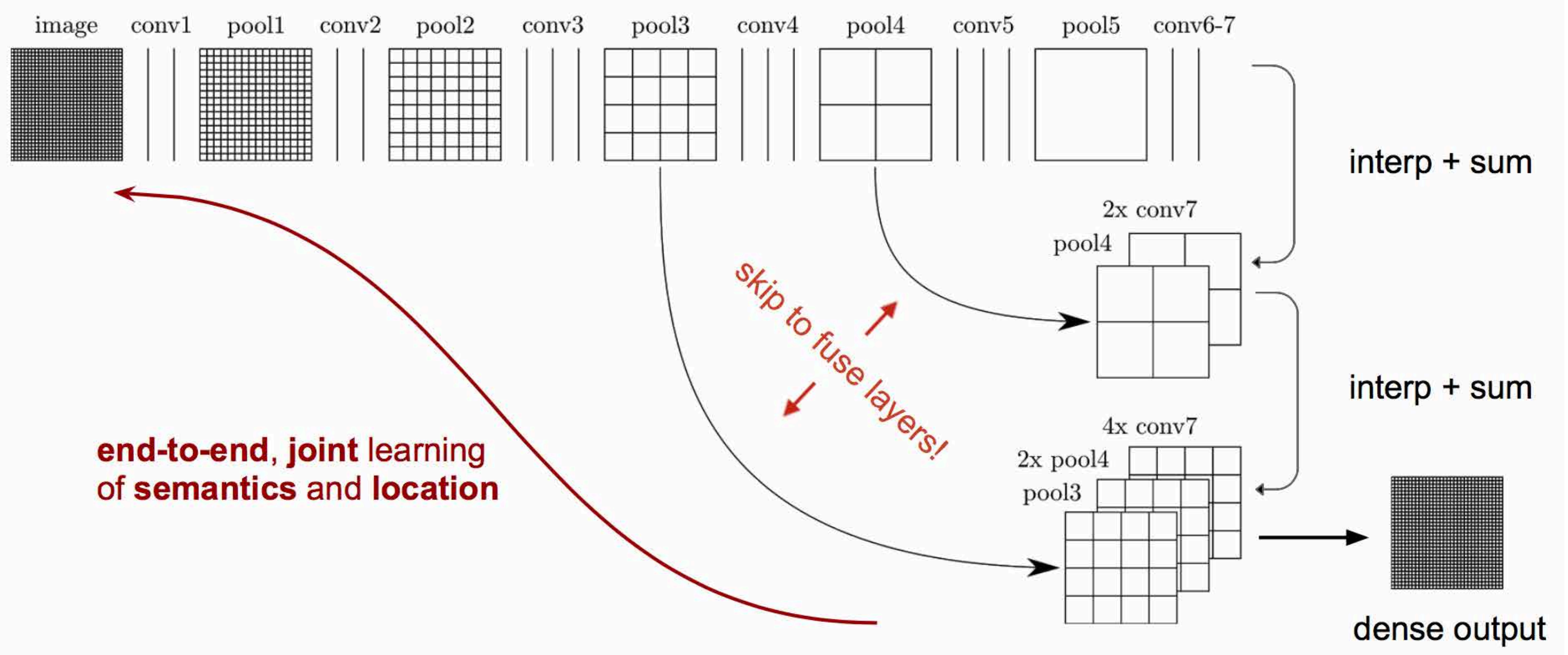
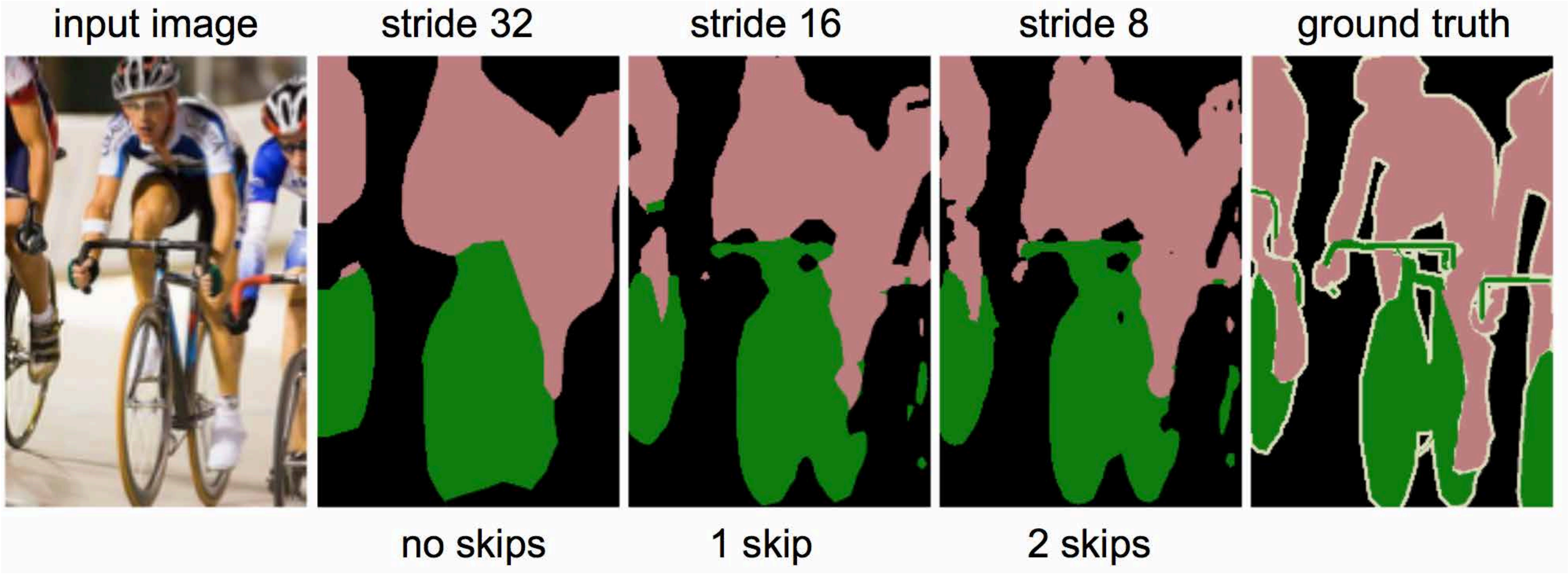
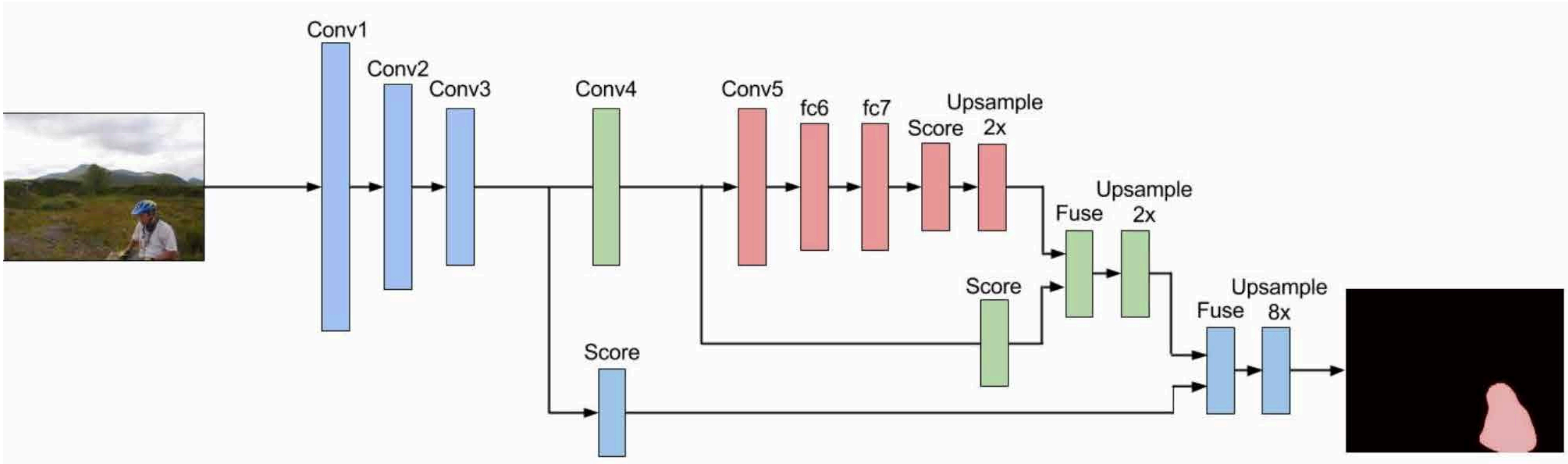


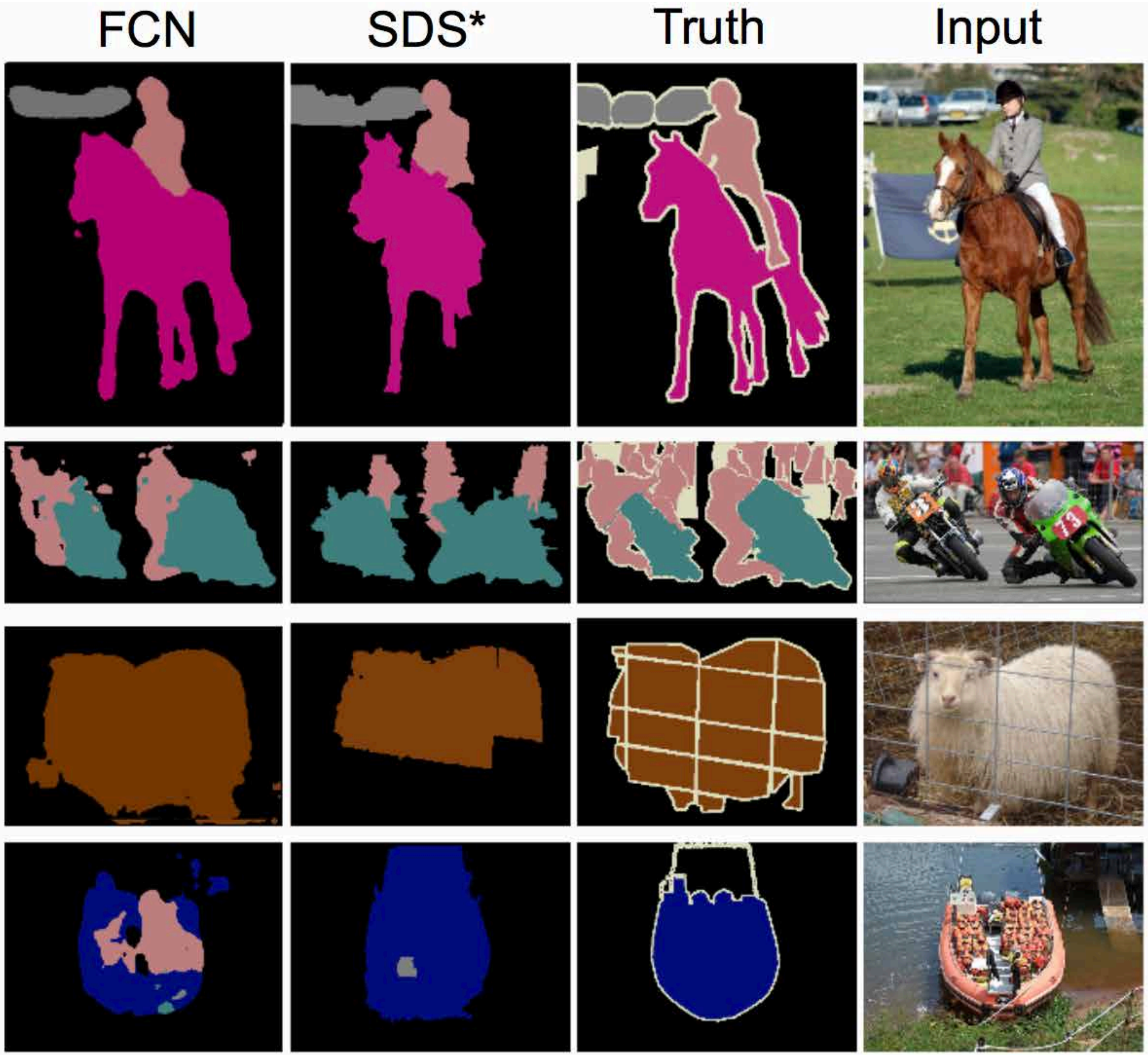
image
conv1
pool1
conv2
pool2
conv3
pool3
conv4
pool4
conv5
pool5
conv6-7
interp + sum
2x conv7
pool4
skip to fuse layers!
interp + sum
4x conv7
2x pool4
pool3
end-to-end, joint learning
of semantics and location
dense output

Skip Layer Refinement



Skip-FCN Computation



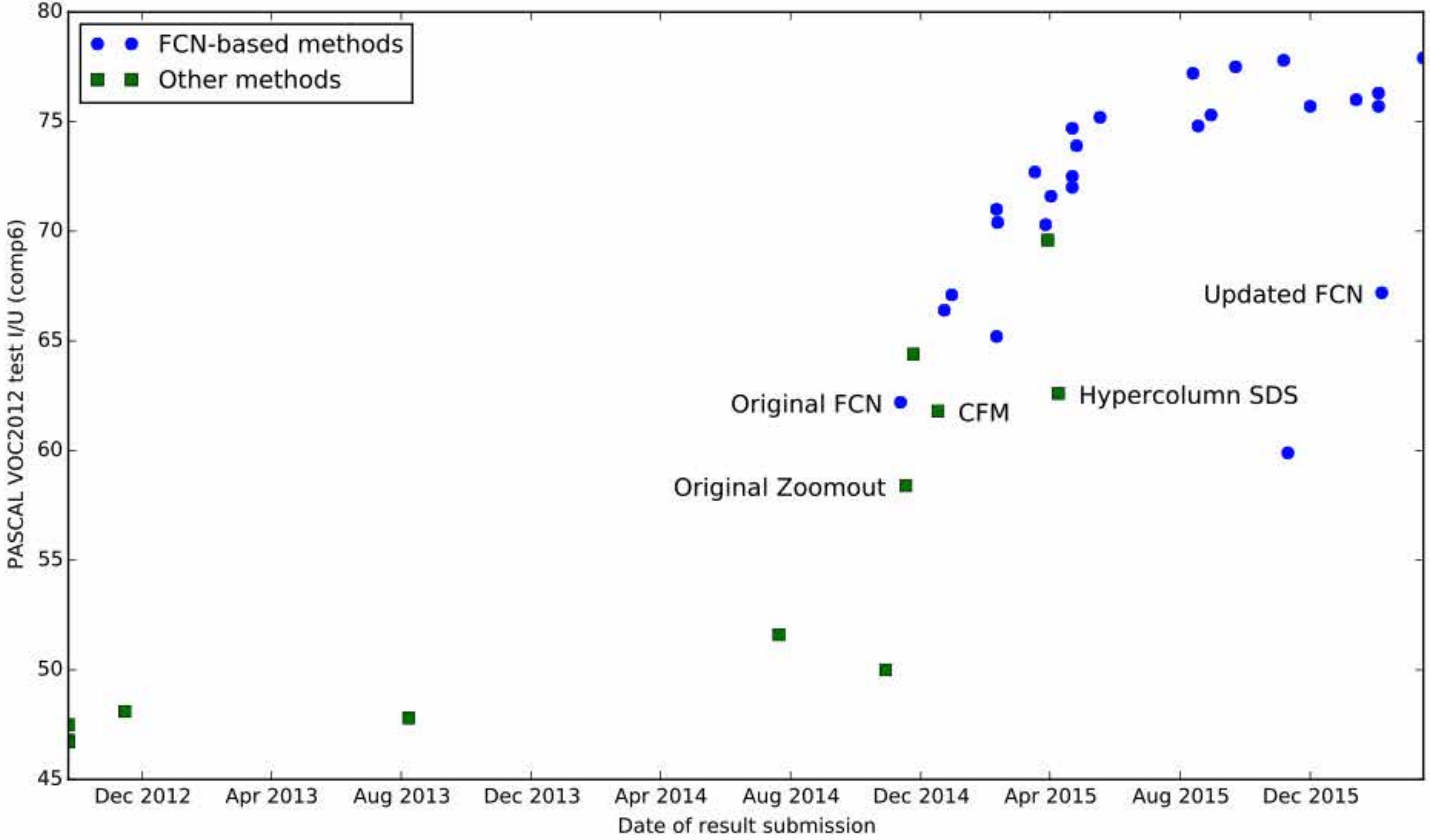


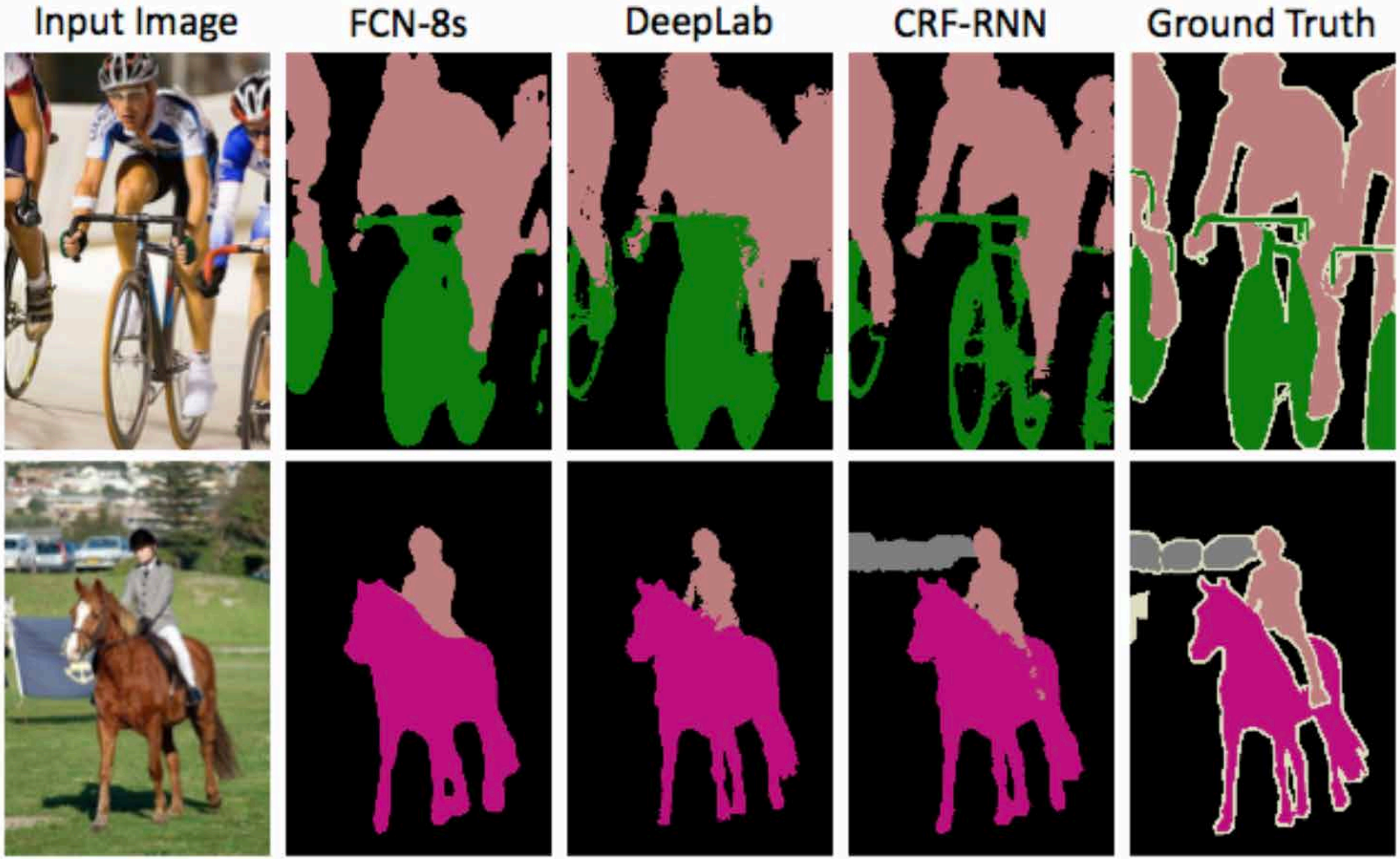
Relative to prior state-of-the-art SDS:

30% relative improvement for mean IoU

286× faster

*Simultaneous Detection and Segmentation Hariharan et al. ECCV14


FCN-based methods
Other methods
Original FCN
CFM
Hypercolumn SDS
Original Zoomout
Updated FCN
PASCAL VOC2012 test I/U (comp6)
Date of result submission
80
75
70
65
60
55
50
45
Dec 2012
Apr 2013
Aug 2013
Dec 2013
Apr 2014
Aug 2014
Dec 2014
Apr 2015
Aug 2015
Dec 2015



[comparison credit: CRF as RNN, Zheng* & Jayasumana* et al. ICCV 2015]

DeepLab: Chen* & Papandreou* et al. ICLR 2015. **CRF-RNN**: Zheng* & Jayasumana* et al. ICCV 2015

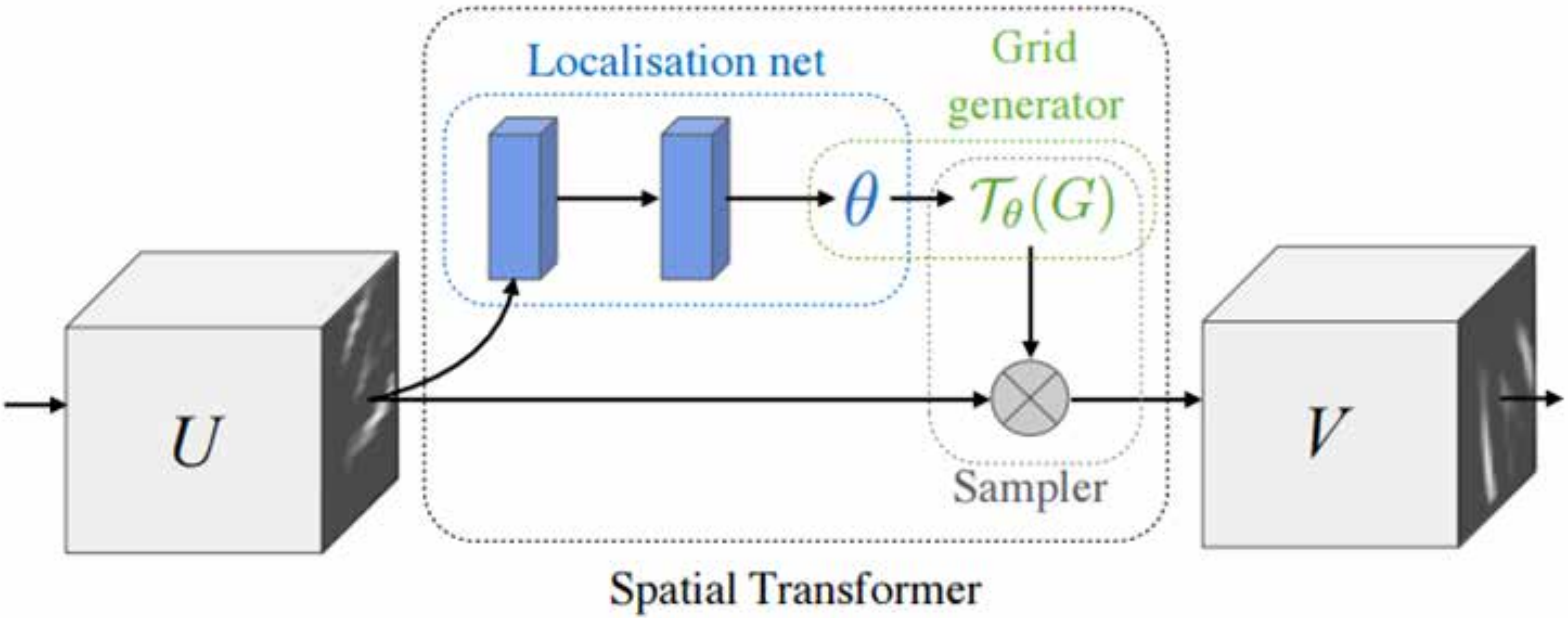
Teaser #2

Spatial Transformer Networks (2015)

By

A group in Google DeepMind

Spatial Transformer Networks (2015)



A Spatial Tranfomer module

Spatial Transformer Networks (2015)

- Spatial Transformer Module:
  - transforms the input image in a way so that the subsequent layers have an easier time making a classification.
- Instead of making changes to the main CNN architecture itself, the authors worry about making changes to the image *before* it is fed into the specific conv layer.
- The 2 things that this module hopes to correct are:
  - pose normalization (scenarios where the object is tilted or scaled)
  - spatial attention (bringing attention to the correct object in a crowded image).

Spatial Transformer Networks (2015)

- Drop this module into a CNN at any point and help the network learn how to transform feature maps in a way that minimizes the cost function during training.

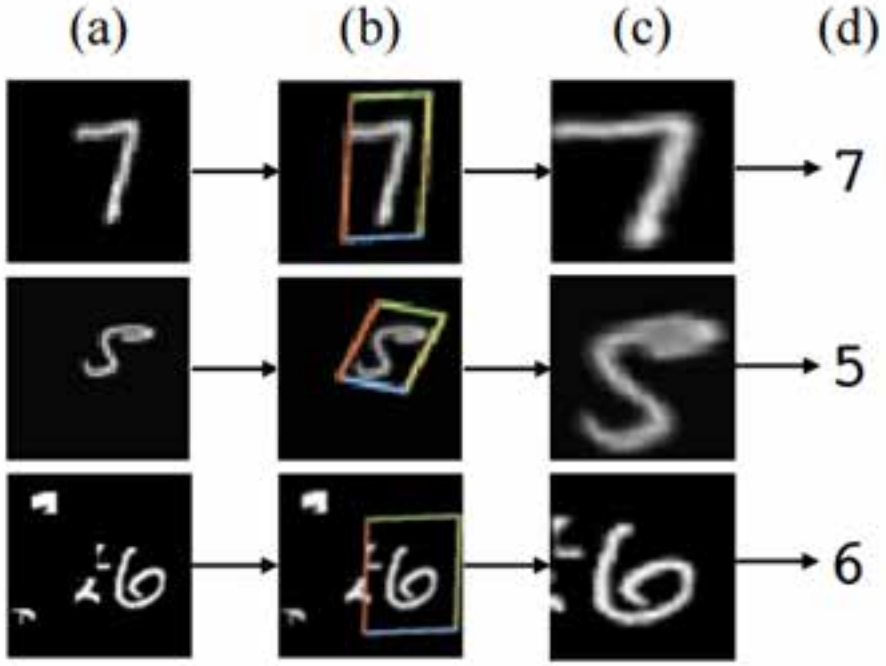


Figure 1: The result of using a spatial transformer as the first layer of a fully-connected network trained for distorted MNIST digit classification. (a) The input to the spatial transformer network is an image of an MNIST digit that is distorted with random translation, scale, rotation, and clutter. (b) The localisation network of the spatial transformer predicts a transformation to apply to the input image. (c) The output of the spatial transformer, after applying the transformation. (d) The classification prediction produced by the subsequent fully-connected network on the output of the spatial transformer. The spatial transformer network (a CNN including a spatial transformer module) is trained end-to-end with only class labels – no knowledge of the groundtruth transformations is given to the system.

Great overview of the function of a Spatial Tranfomer module

Spatial Transformer Networks (2015)