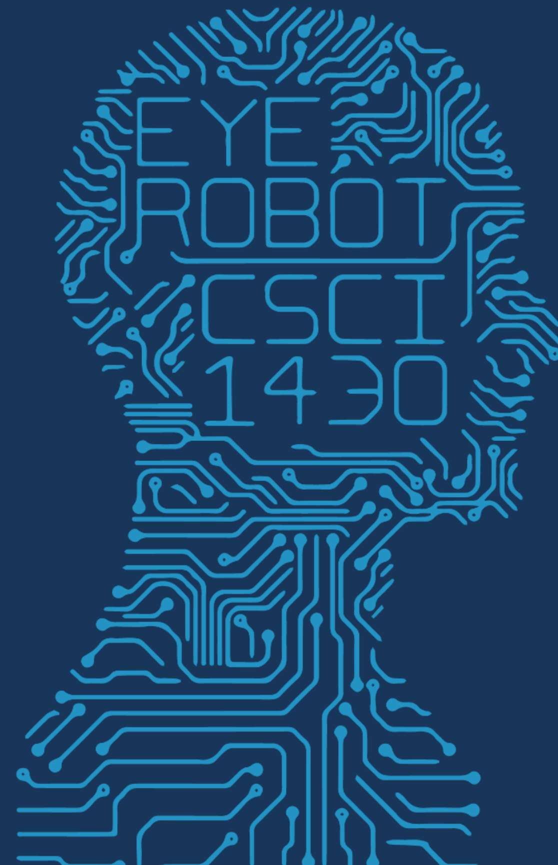
1950

Future Vision



2017 MWF 1pm 368

Computer Vision

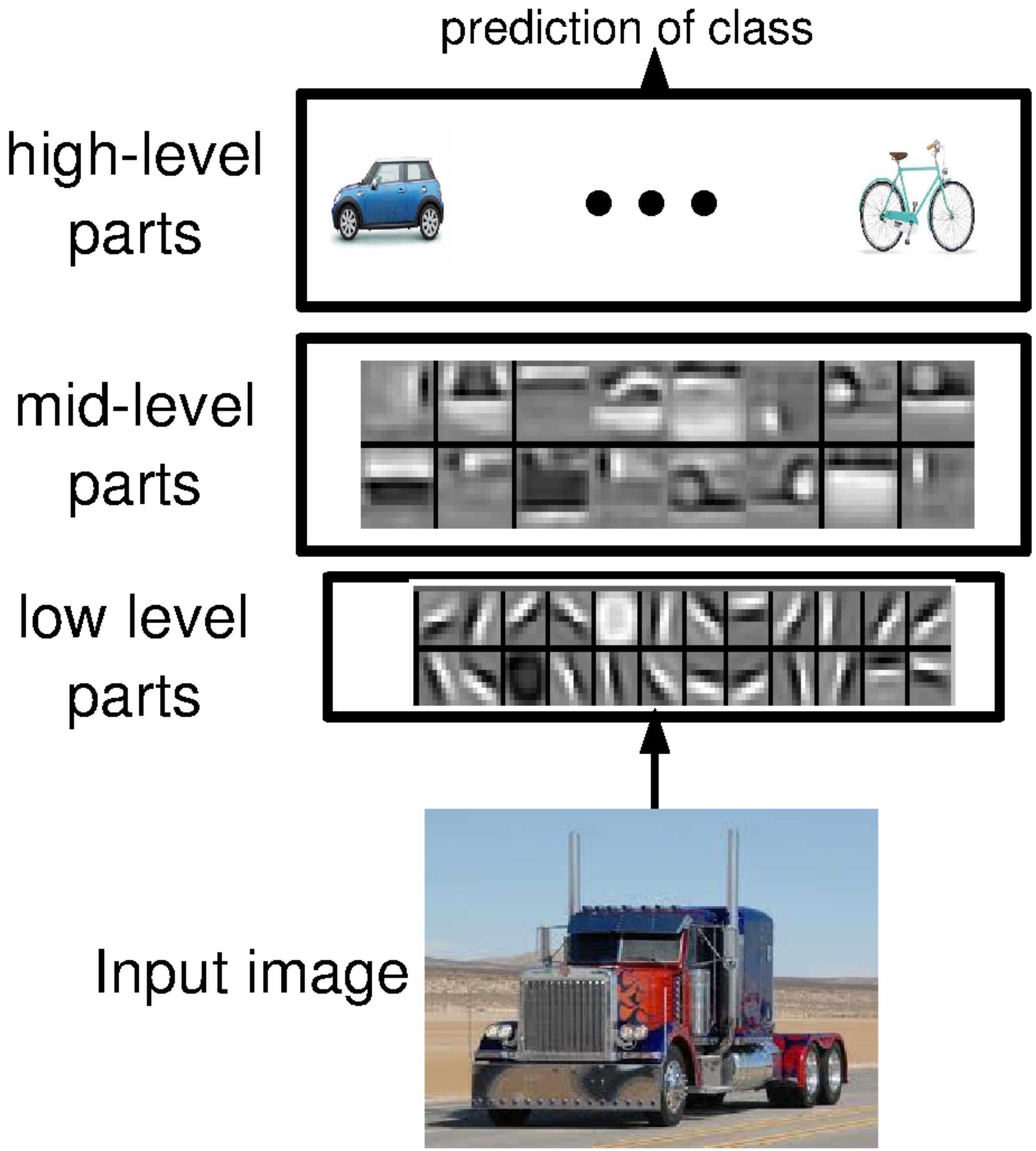
[Boston March for Science 2017 – photo Hendrik Strobelt]

[Boston March for Science 2017]

[Boston March for Science 2017]

[Boston March for Science 2017]

Interpretation



- distributed representations
- feature sharing
- compositionality

Lee et al. "Convolutional DBN's ..." ICML 2009

Object Detectors Emerge in Deep Scene CNNs

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, AntonioTorralba

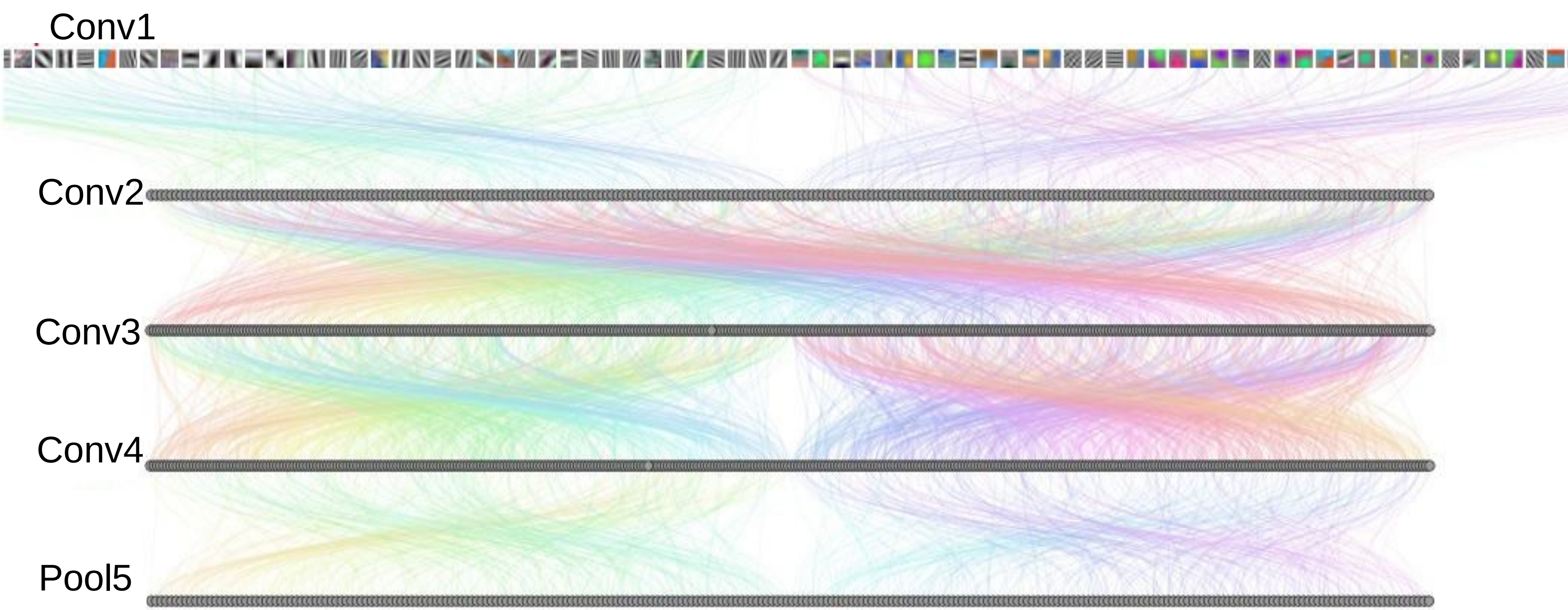



Massachusetts Institute of Technology

ICLR 2015

How Objects are Represented in CNN?

CNN uses **distributed code** to represent objects.

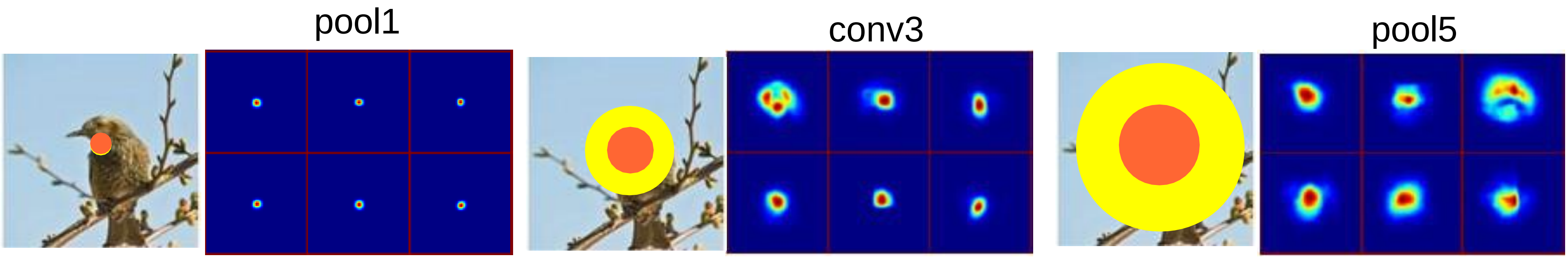


Agrawal, et al. Analyzing the performance of multilayer neural networks for object recognition. ECCV, 2014
Szegedy, et al. Intriguing properties of neural networks.arXiv preprint arXiv:1312.6199, 2013.
Zeiler, M. et al. Visualizing and Understanding Convolutional Networks, ECCV 2014.

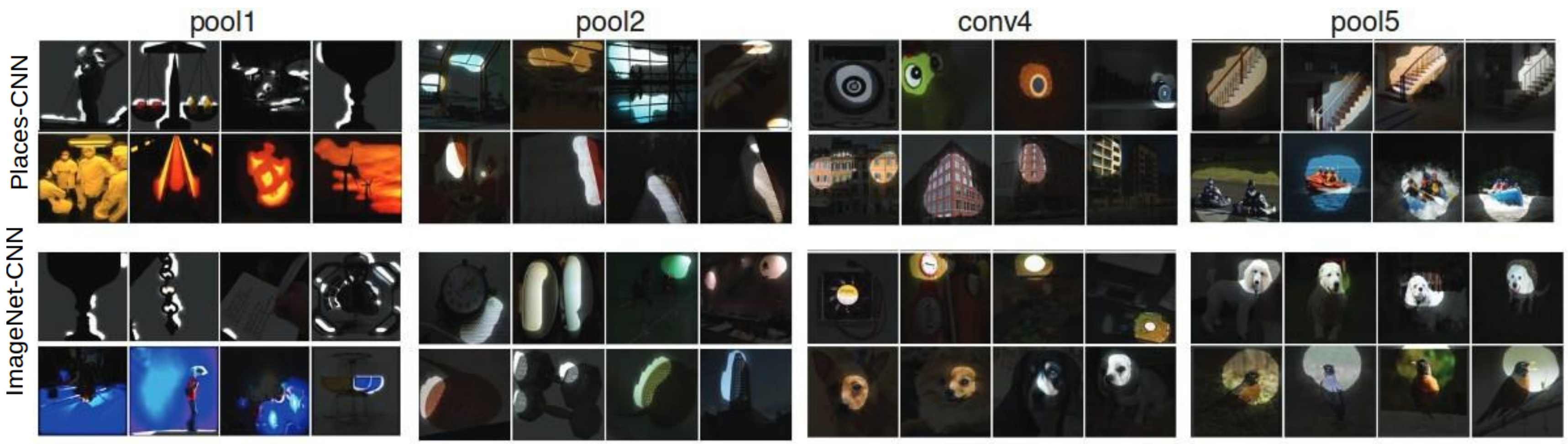
Estimating the Receptive Fields

Estimated receptive fields

Actual size of RF is much smaller than the theoretic size



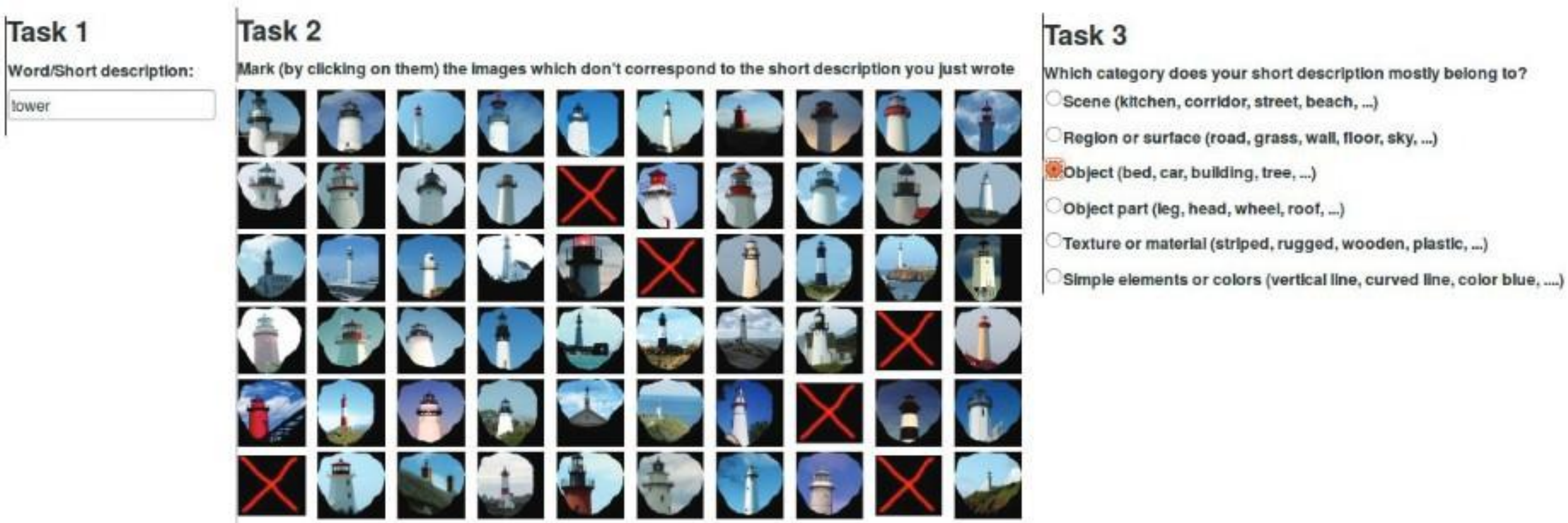
Segmentation using the RF of Units



More semantically meaningful

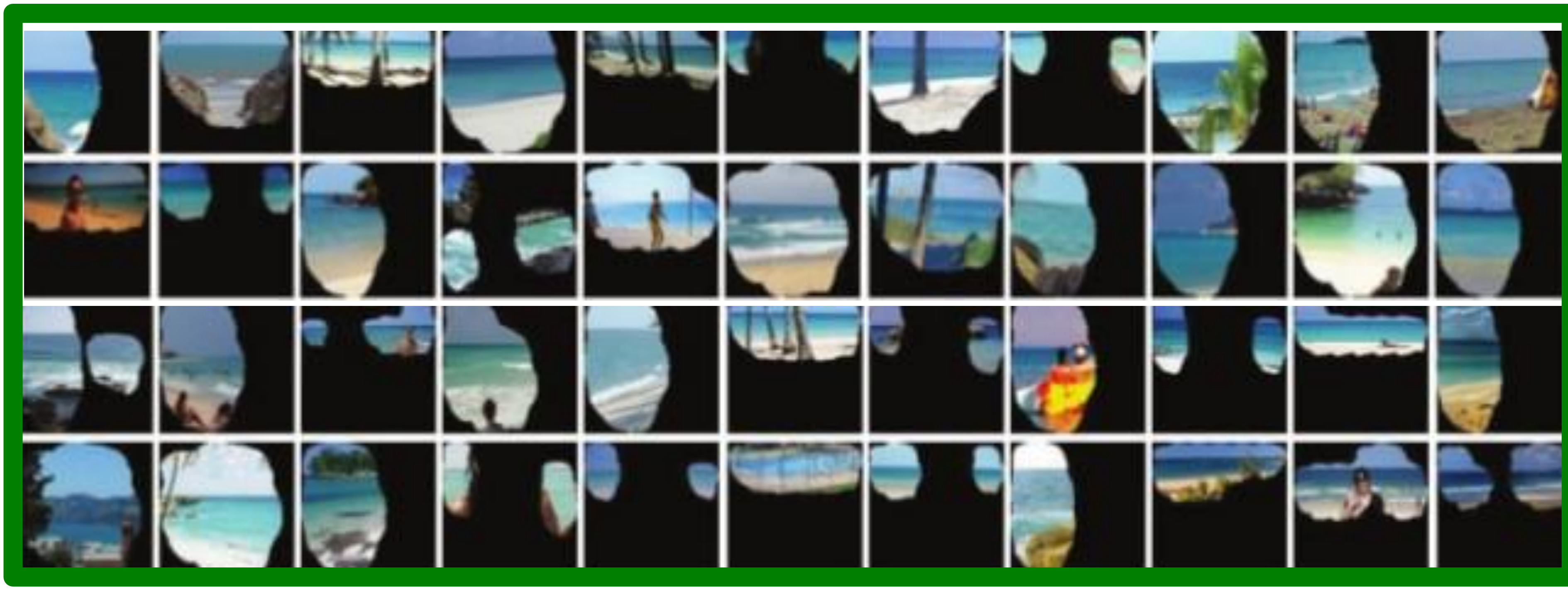
Annotating the Semantics of Units

Top ranked segmented images are cropped and sent to Amazon Turk for annotation.



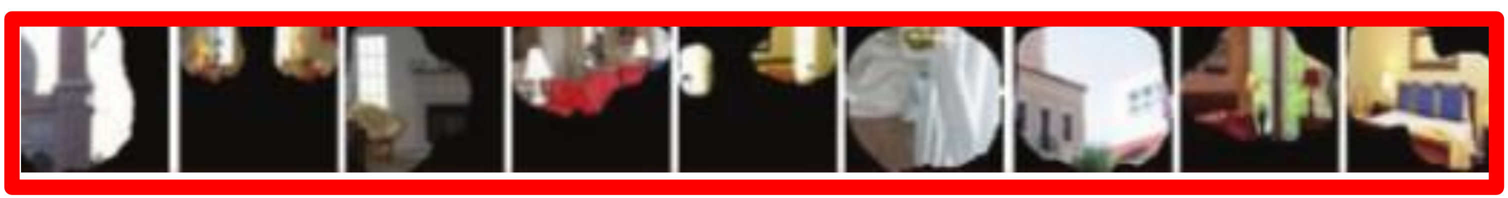
Annotating the Semantics of Units

Pool5, unit 76; Label: ocean; Type: scene; Precision: 93%



Annotating the Semantics of Units

Pool5, unit 13; Label: Lamps; Type: object; Precision: 84%

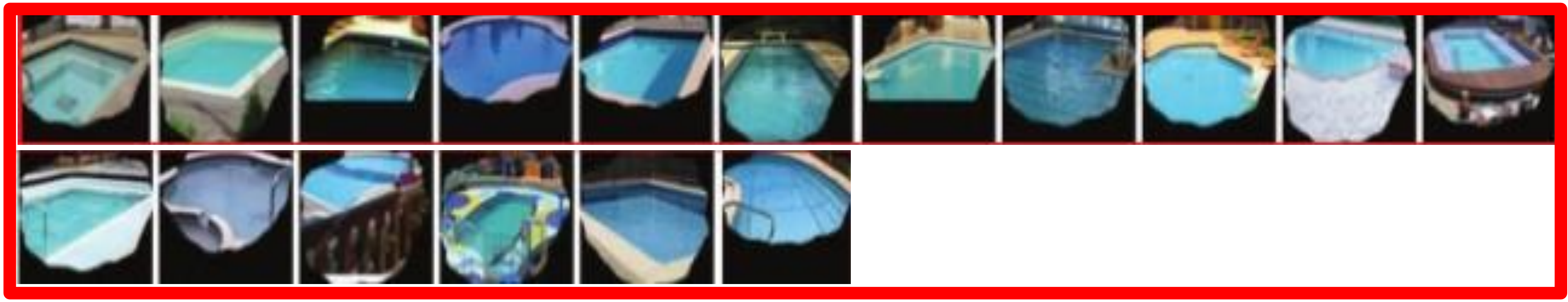


Annotating the Semantics of Units

Pool5, unit 77; Label:legs; Type: object part; Precision: 96%

Annotating the Semantics of Units

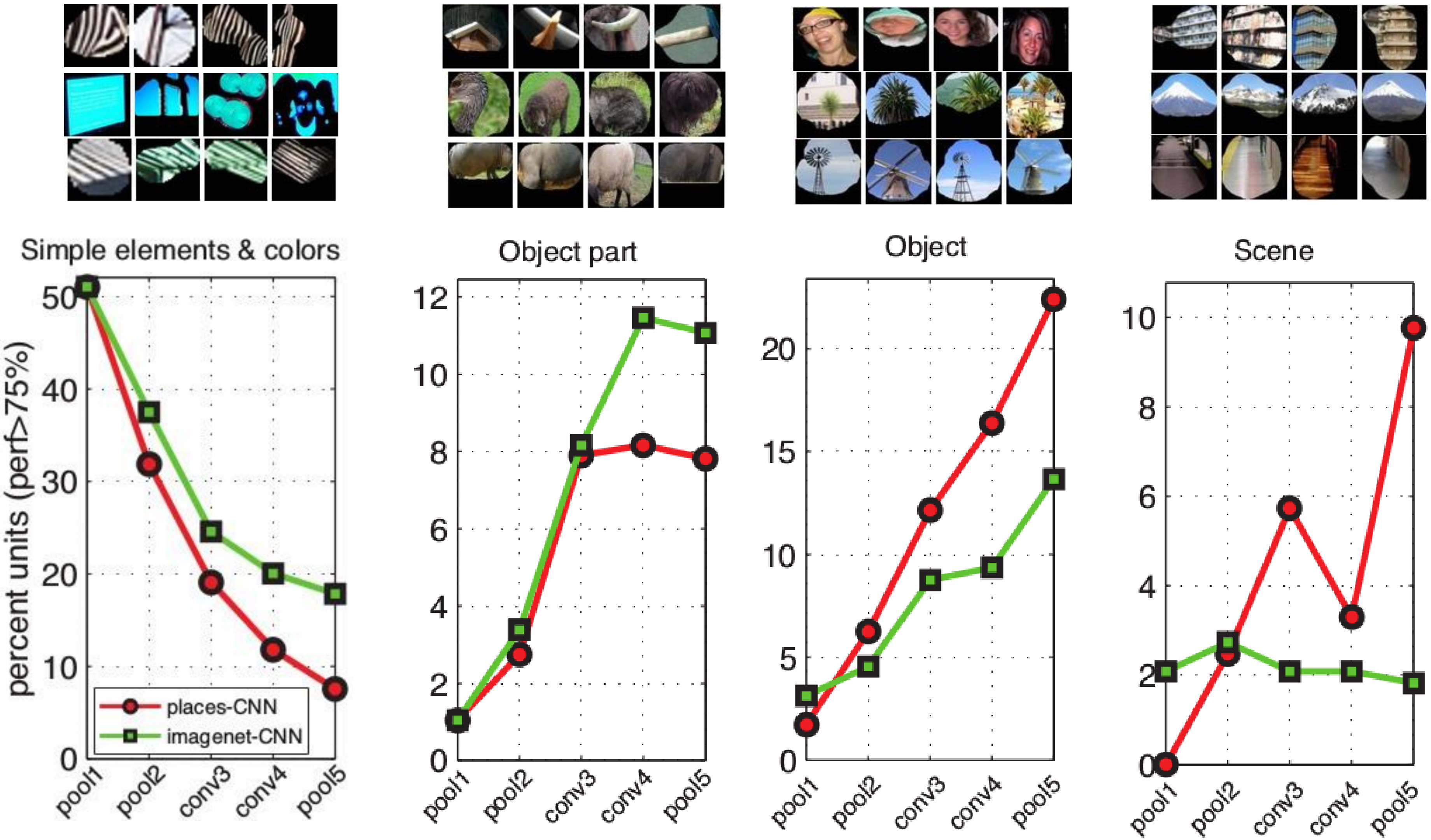
Pool5, unit 112; Label: pool table; Type: object; Precision: 70%



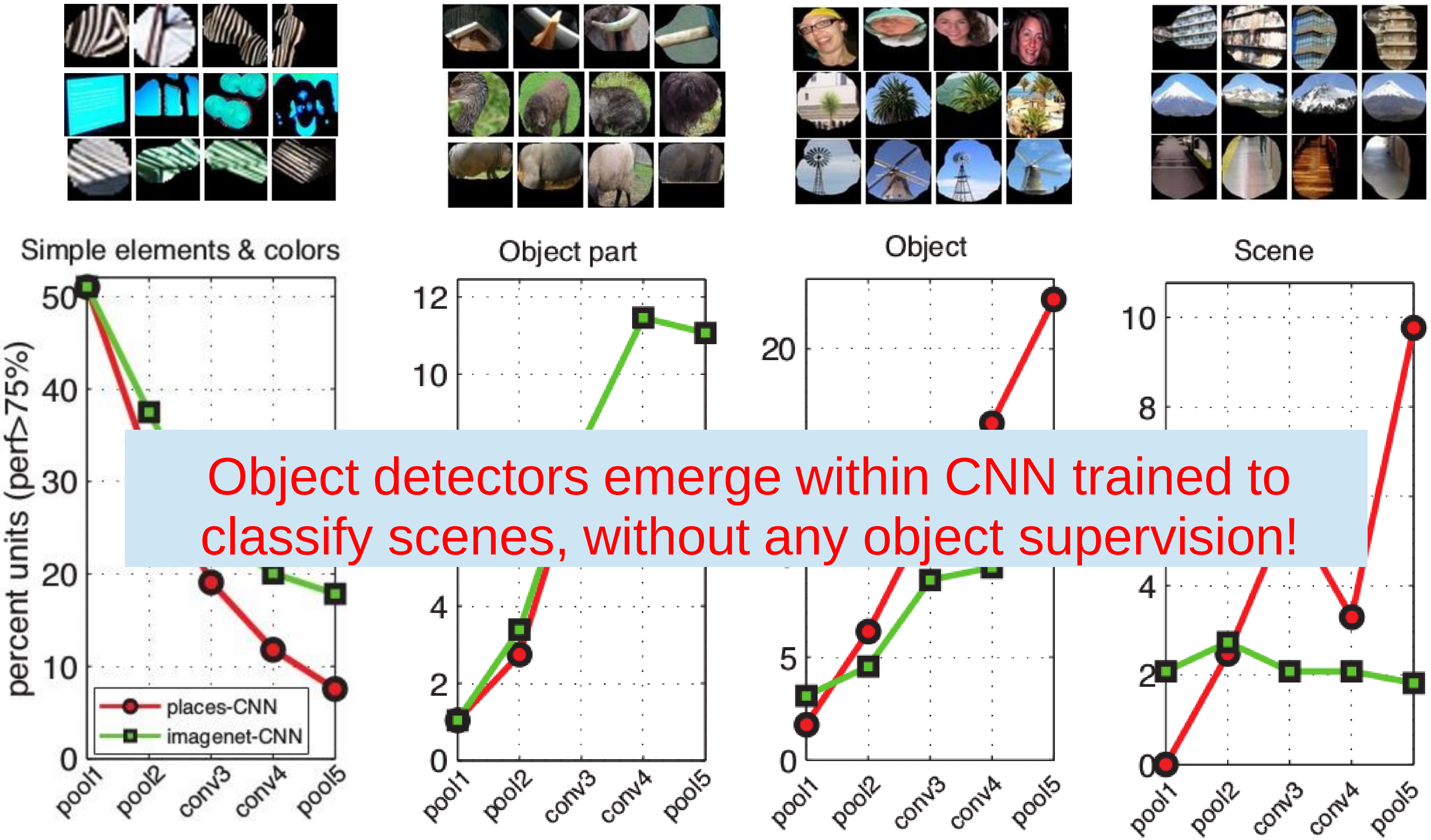
Annotating the Semantics of Units

Pool5, unit 22; Label: dinner table; Type: scene; Precision: 60%

Distribution of Semantic Types at Each Layer

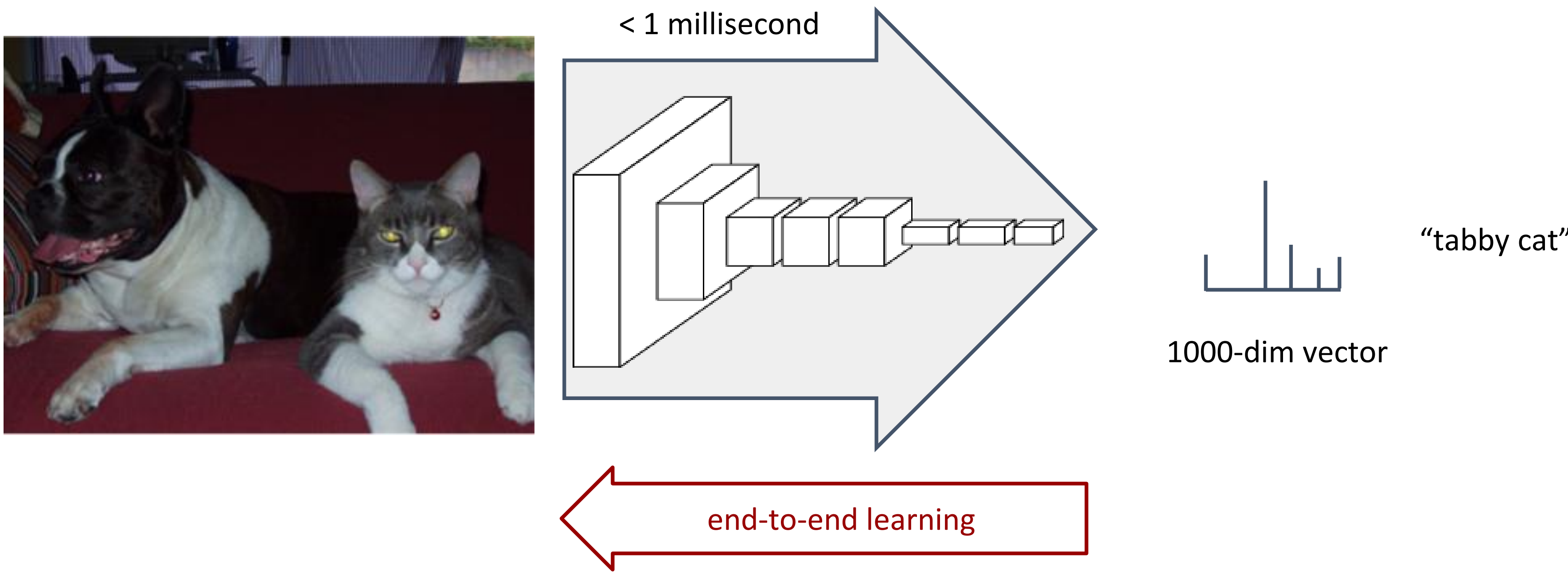


Distribution of Semantic Types at Each Layer



Object detectors emerge within CNN trained to classify scenes, without any object supervision!

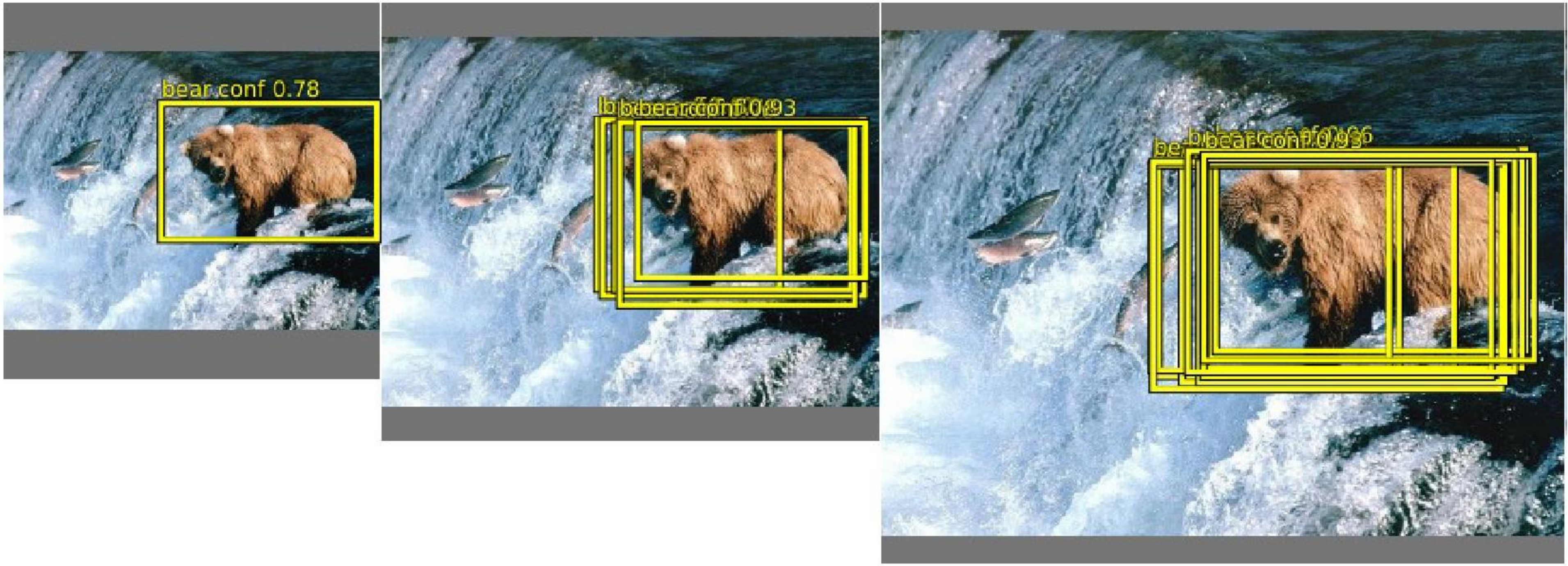
ConvNets perform classification



[Slides from Long, Shelhamer, and Darrell]

CONV NETS: EXAMPLES

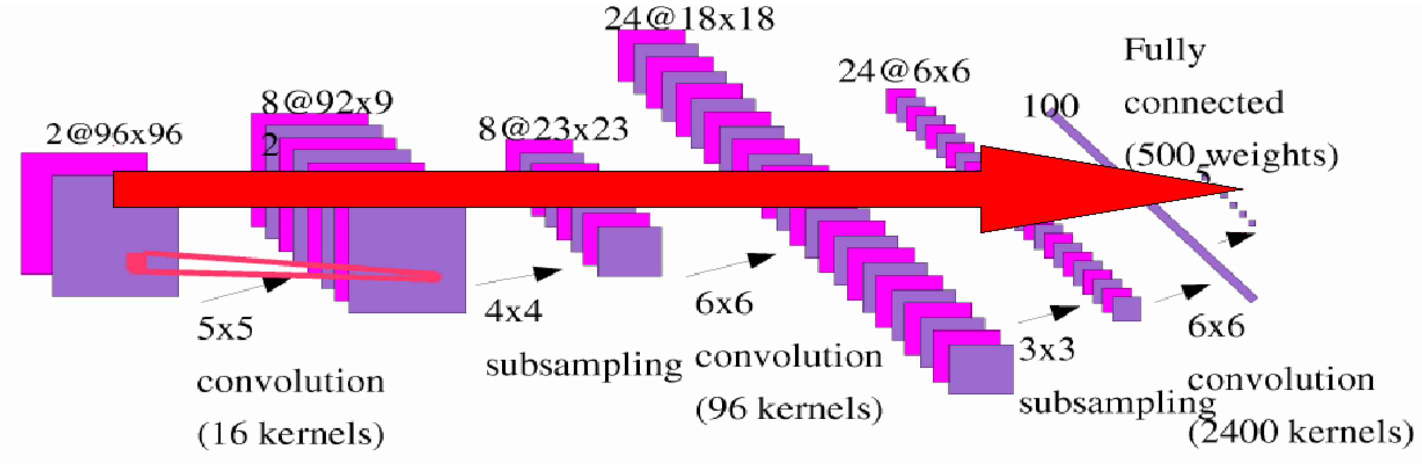
- **Object detection**



Sermanet et al. "OverFeat: Integrated recognition, localization, ..." arxiv 2013
Girshick et al. "Rich feature hierarchies for accurate object detection..." arxiv 2013
Szegedy et al. "DNN for object detection" NIPS 2013

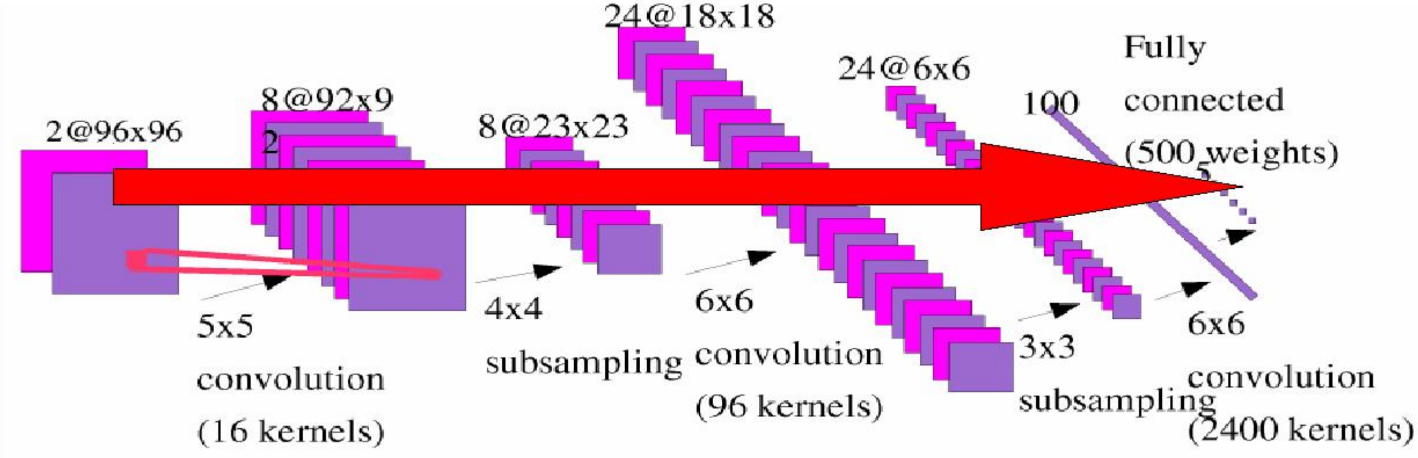
ConvNets: Test

At test time, run only is forward mode (FPROP).

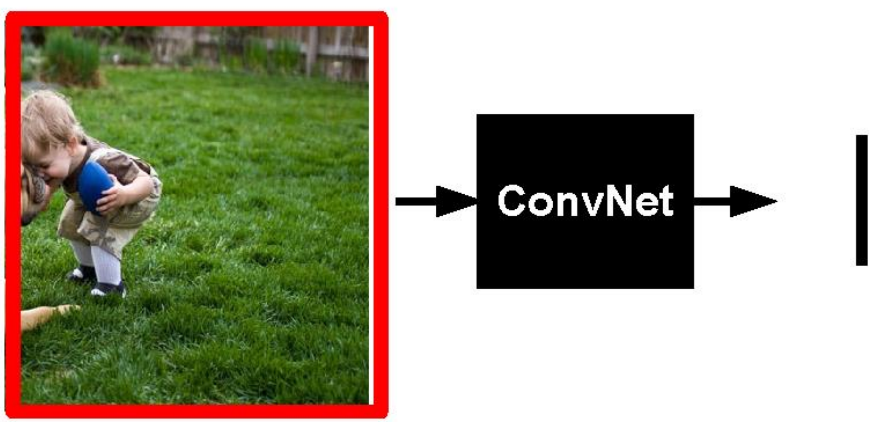


ConvNets: Test

At test time, run only is forward mode (FPROP).



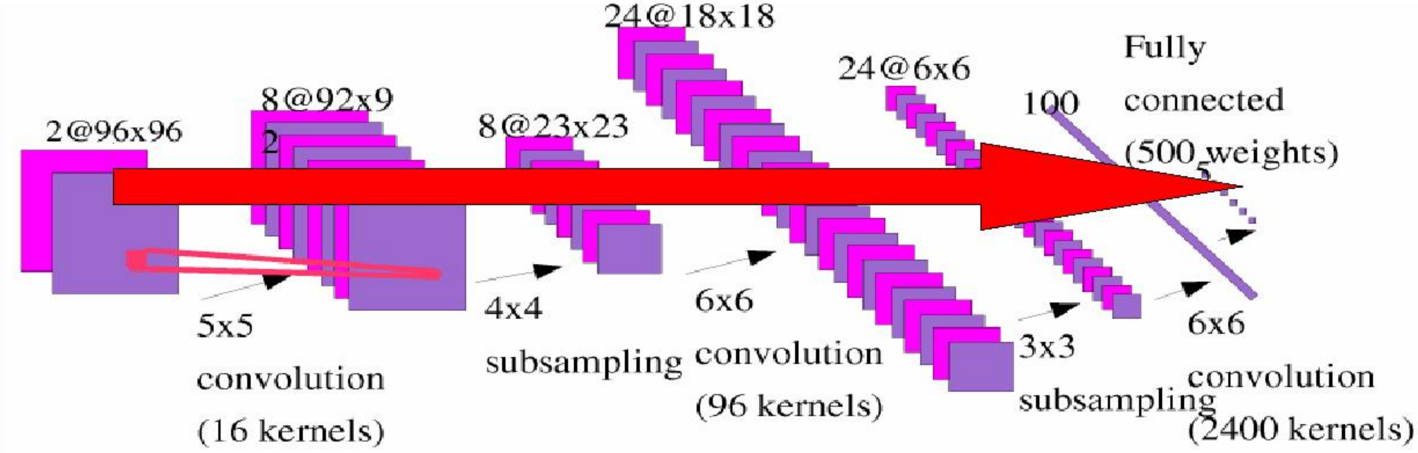
Naturally, convnet can process larger images



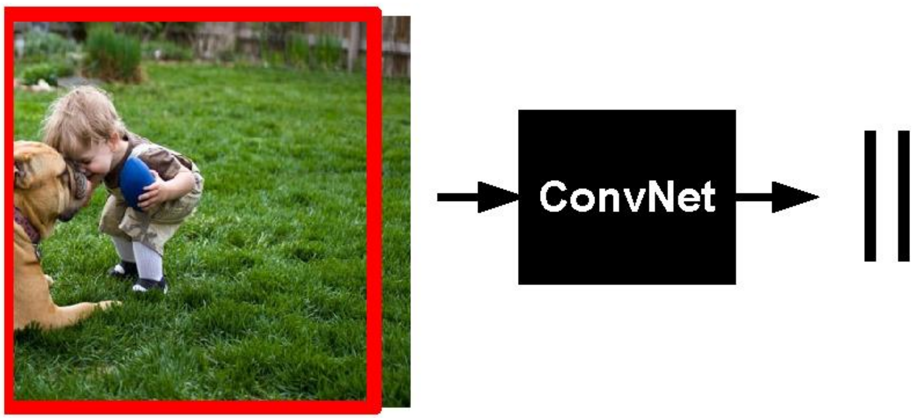
Traditional methods use inefficient sliding windows.

ConvNets: Test

At test time, run only is forward mode (FPROP).



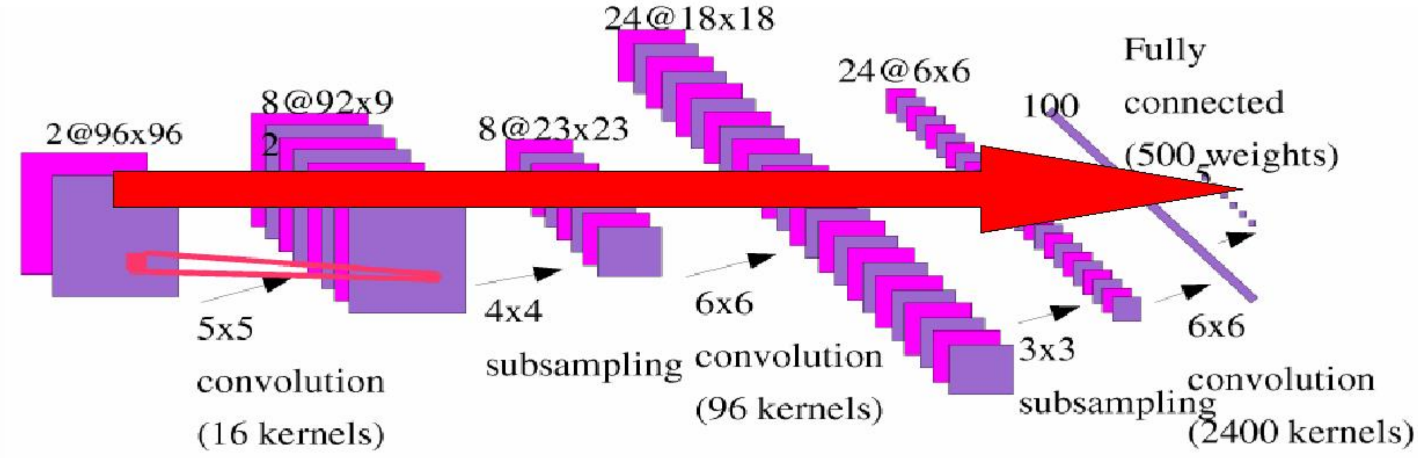
Naturally, convnet can process larger images



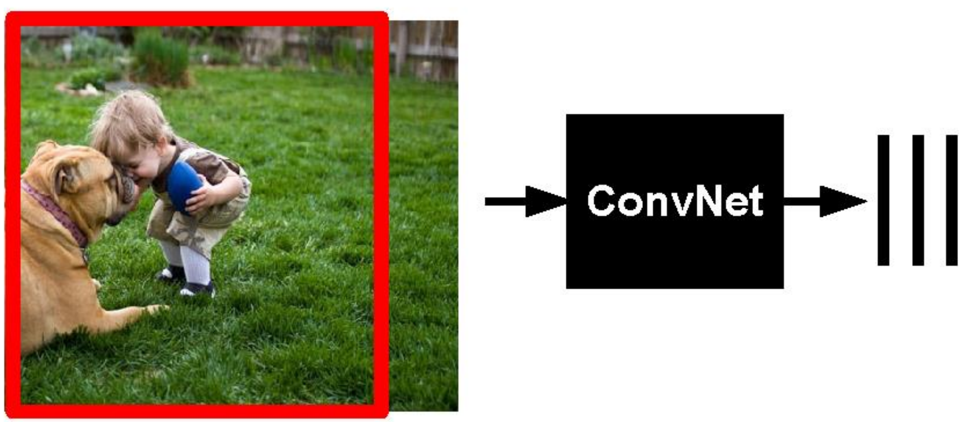
Traditional methods use inefficient sliding windows.

ConvNets: Test

At test time, run only is forward mode (FPROP).



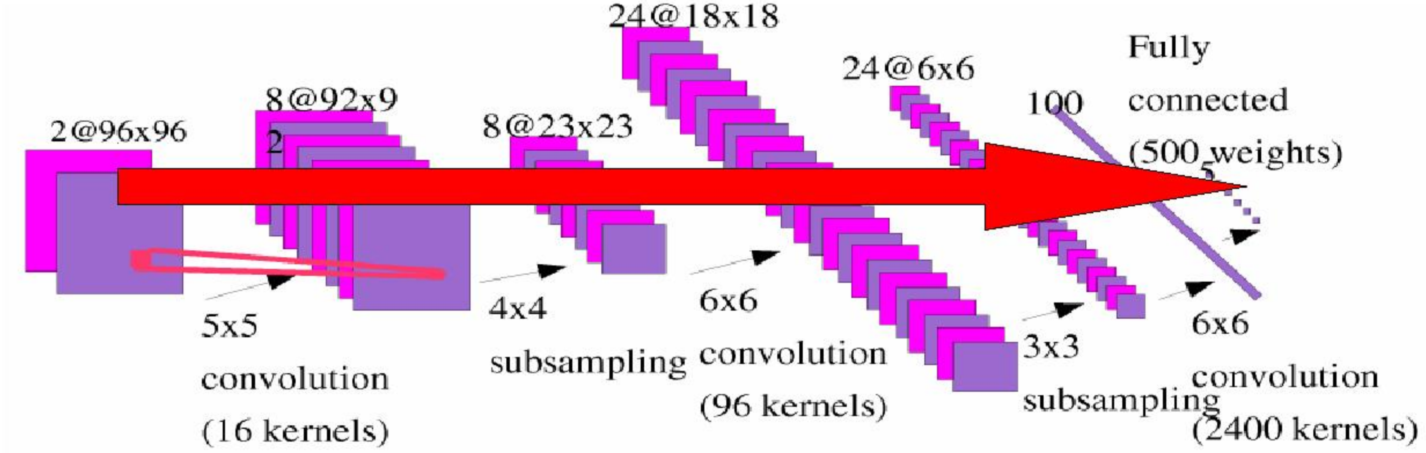
Naturally, convnet can process larger images



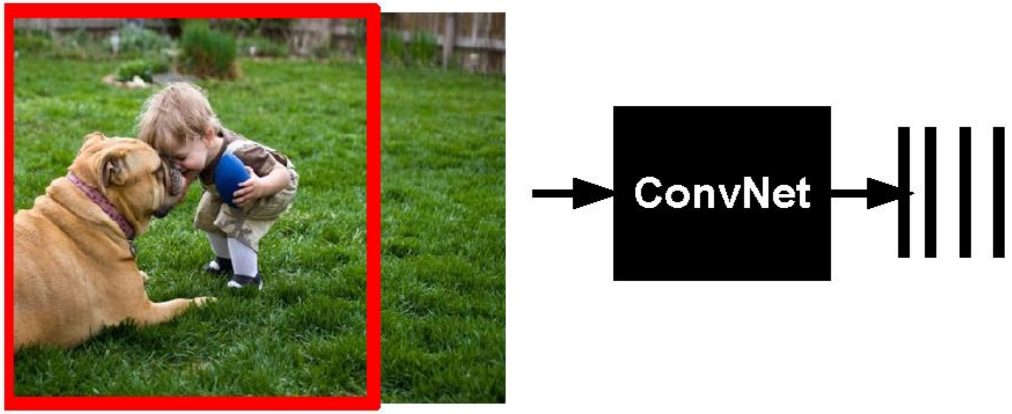
Traditional methods use inefficient sliding windows.

ConvNets: Test

At test time, run only is forward mode (FPROP).

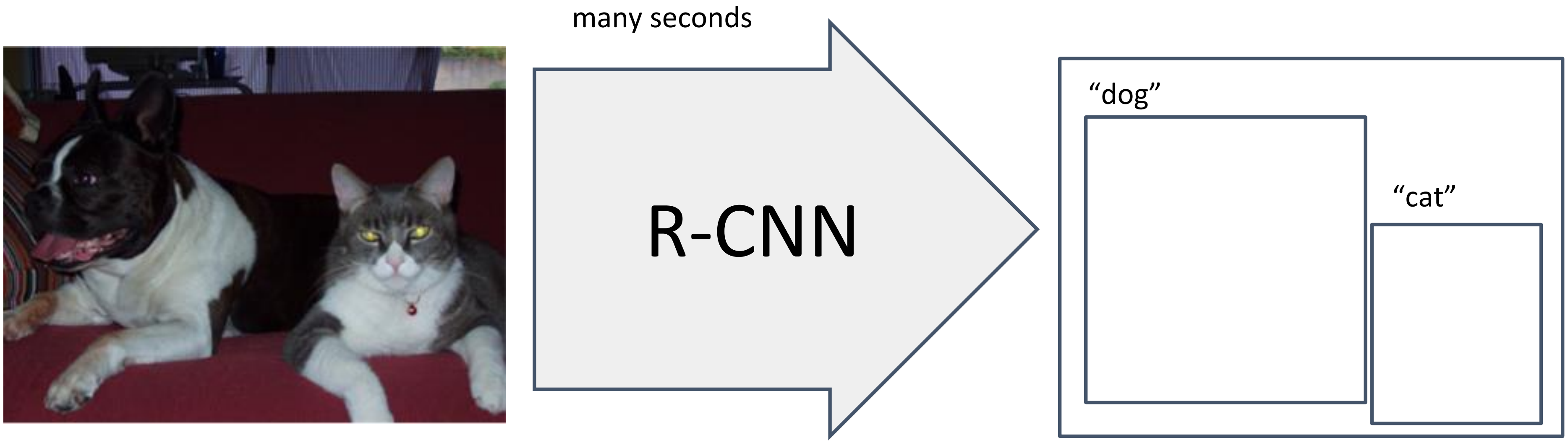


Naturally, convnet can process larger images



Traditional methods use inefficient sliding windows.

R-CNN does detection



[Long et al.]

R-CNN: Region-based CNN

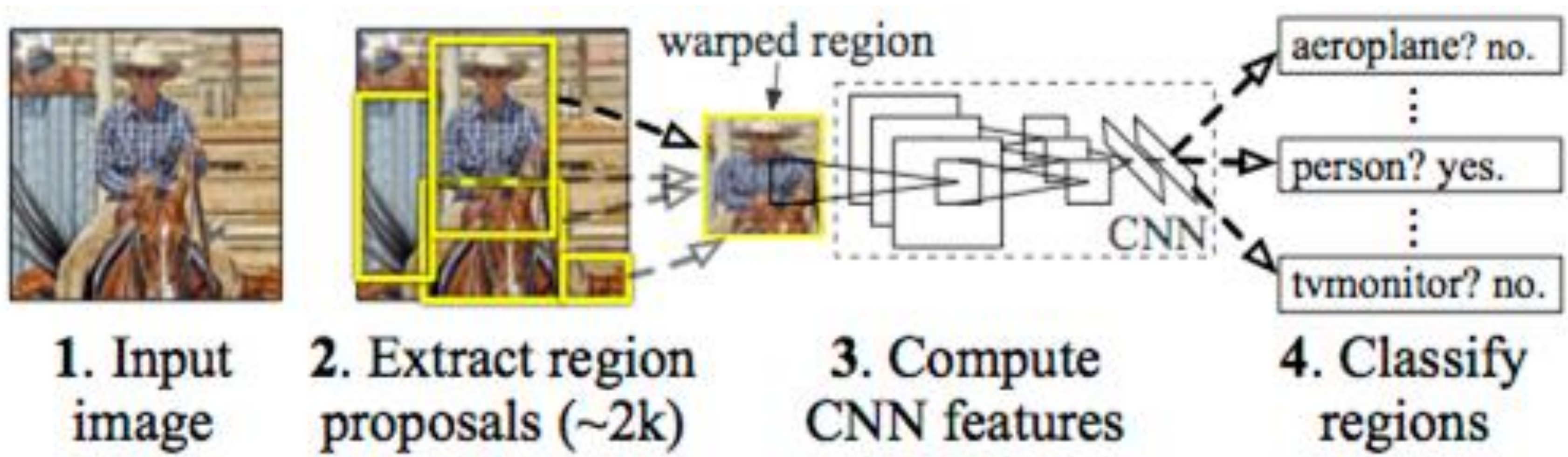
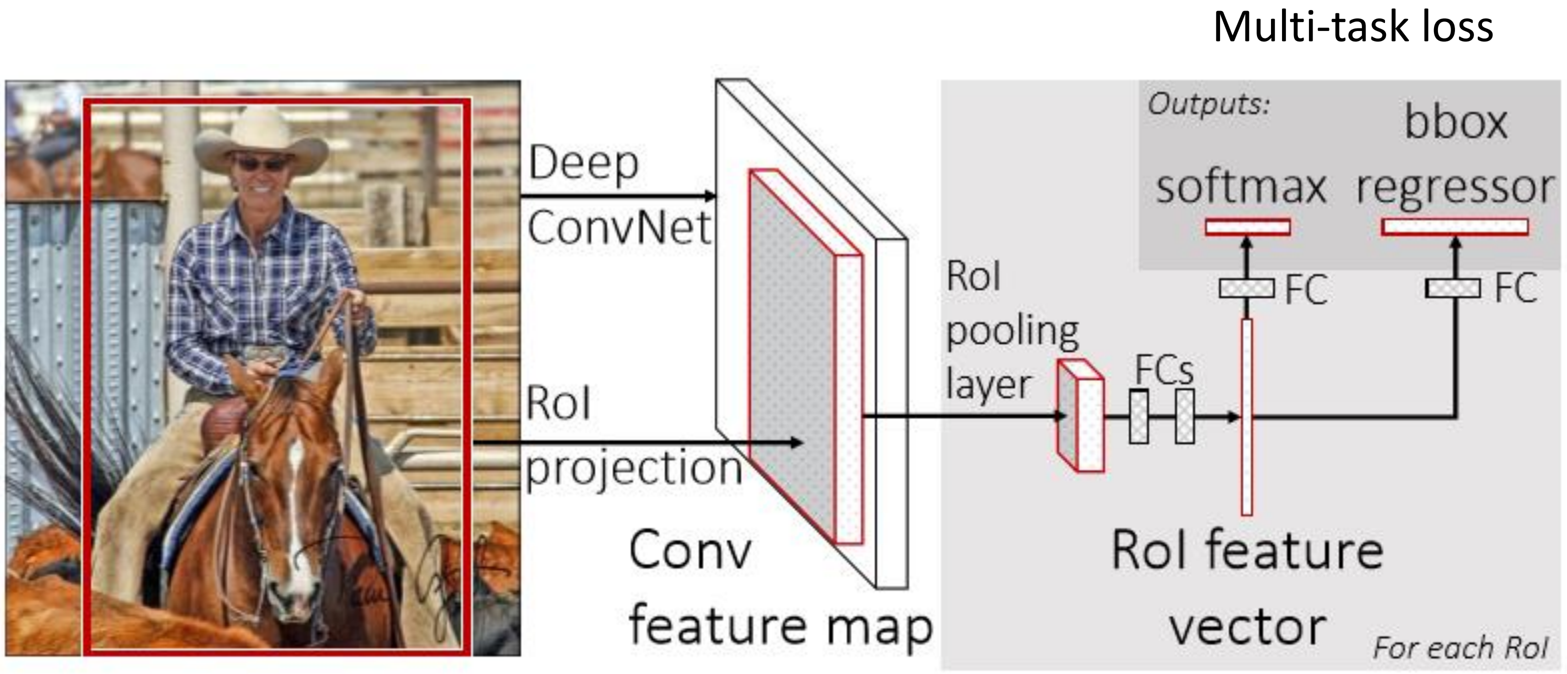


Figure: Girshick et al.

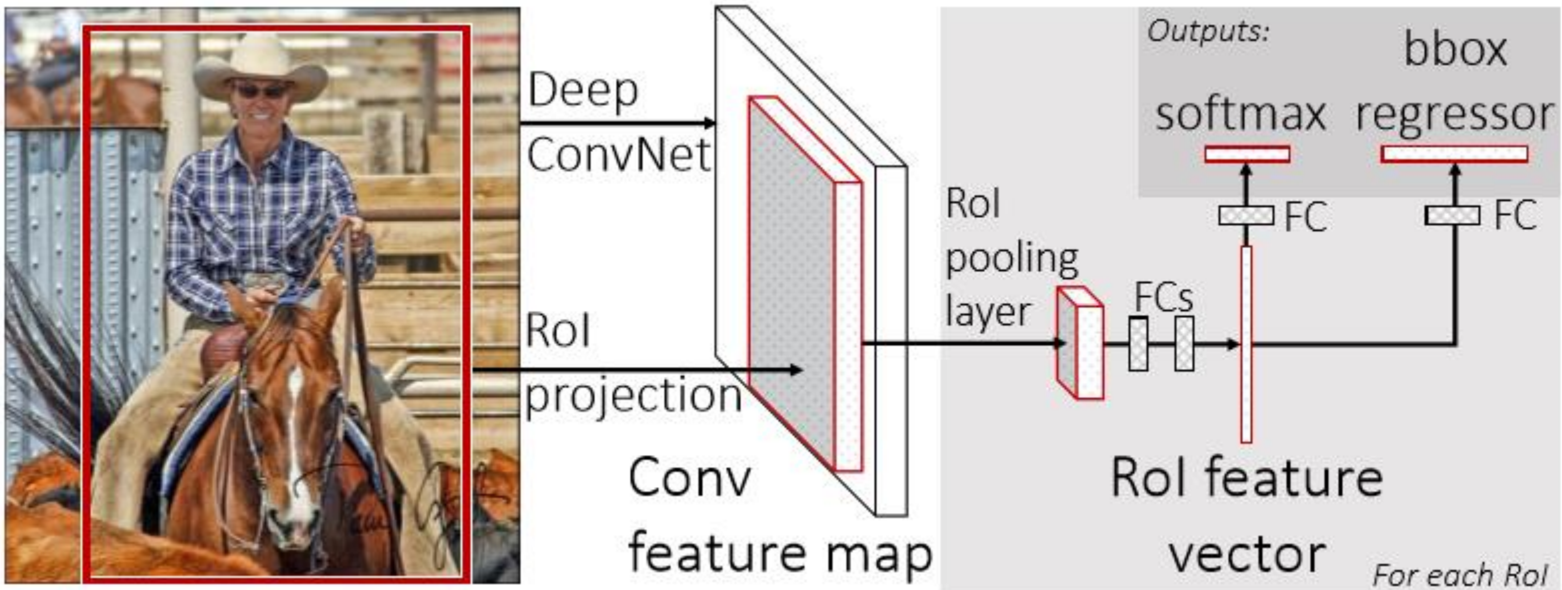
Fast R-CNN



RoI = Region of Interest

Figure: Girshick et al.

Fast R-CNN

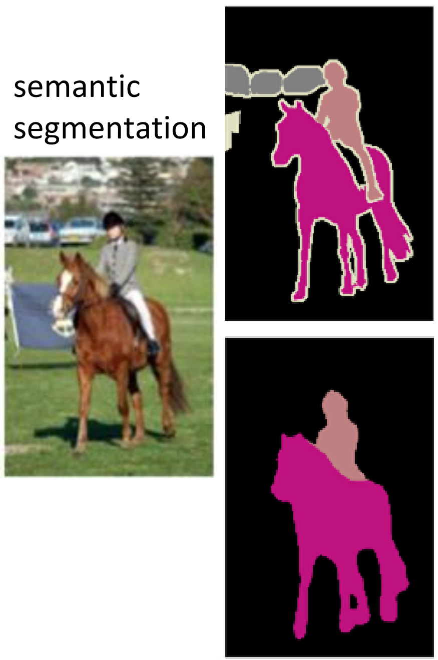


- Convolve whole image into feature map (many layers; abstracted)
- For each candidate RoI:
    - Squash feature map weights into fixed-size ‘RoI pool’ – adaptive subsampling!
        - Divide RoI into H x W subwindows, e.g., 7 x 7, and max pool
    - Learn classification on RoI pool with own fully connected layers (FCs)
    - Output classification (softmax) + bounds (regressor)

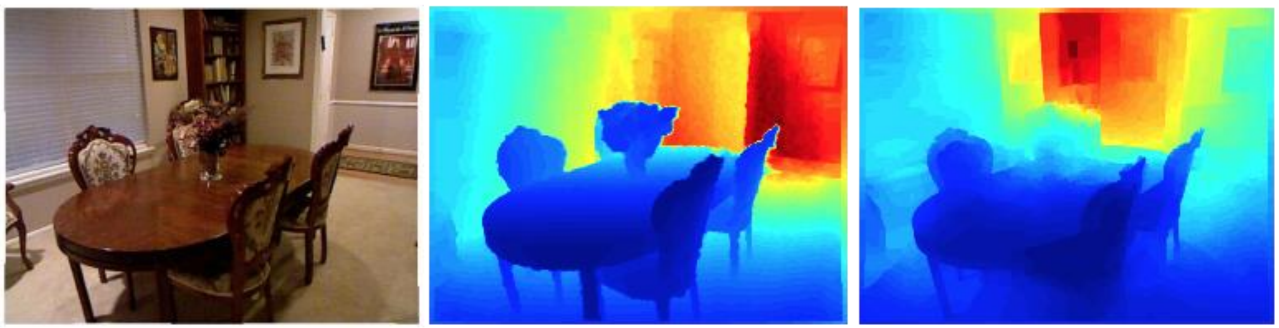
Figure: Girshick et al.

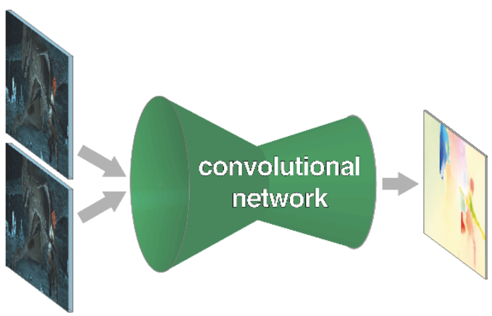
What if we want pixels out?

semantic segmentation



monocular depth estimation Eigen & Fergus 2015

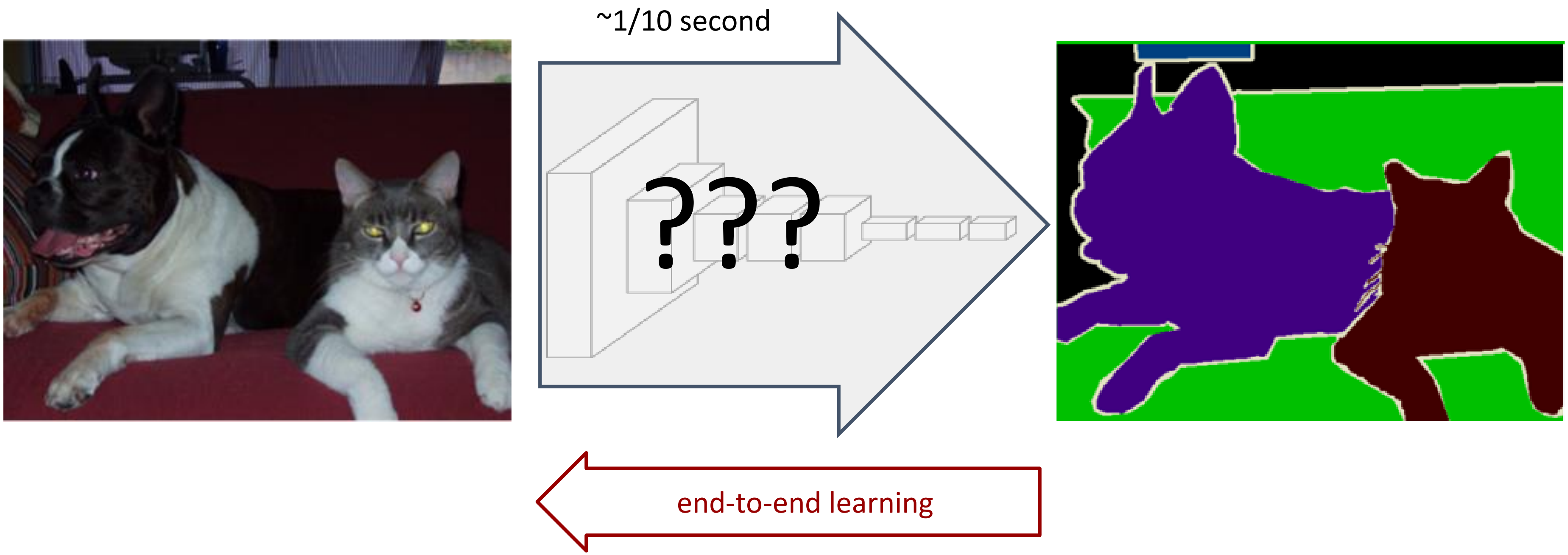




optical flow Fischer et al. 2015

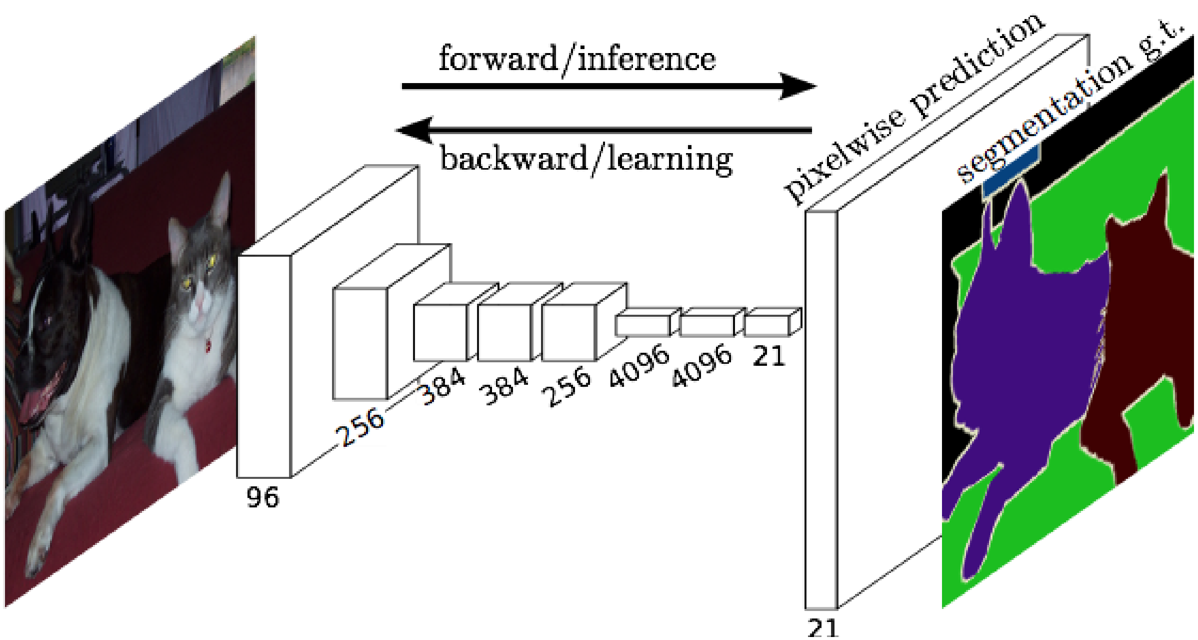
boundary prediction Xie & Tu 2015

[Long et al.]



[Long et al.]

Fully Convolutional Networks for Semantic Segmentation

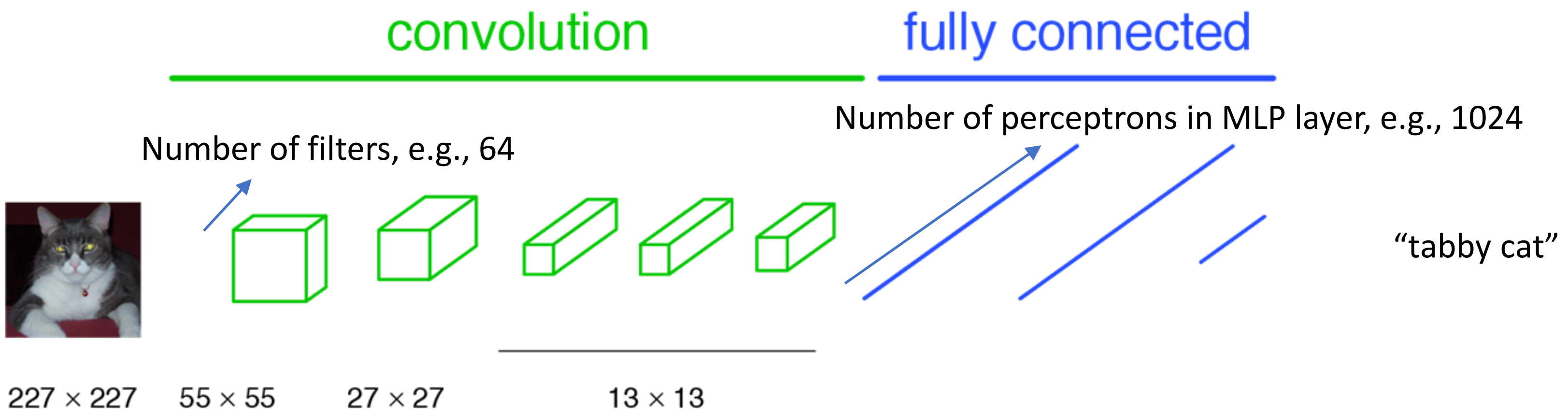


Jonathan Long* Evan Shelhamer* Trevor Darrell
UC Berkeley

[CVPR 2015]

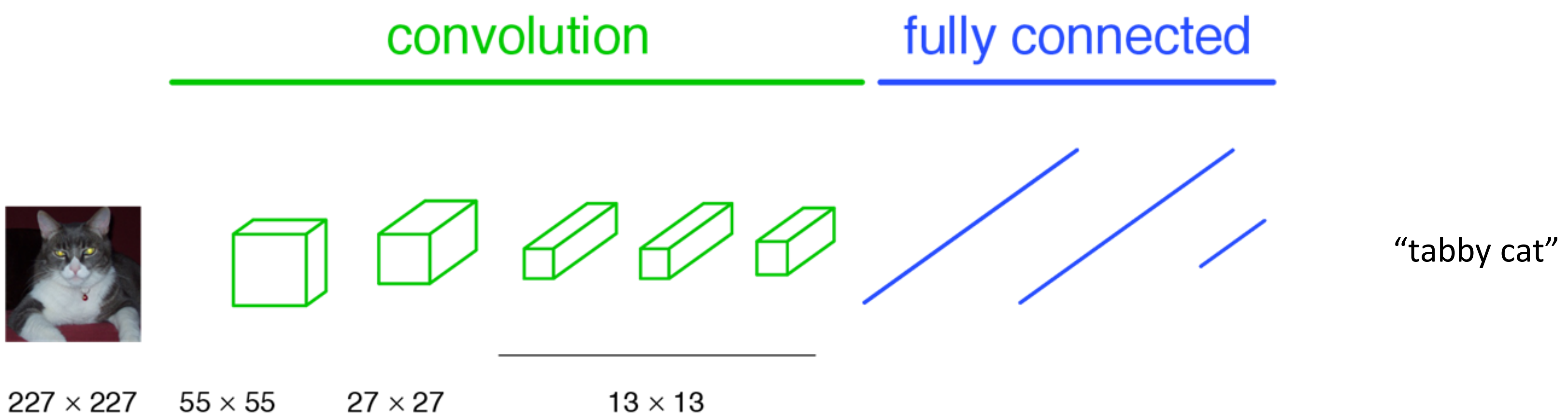
Slides from Long, Shelhamer, and Darrell

A classification network...



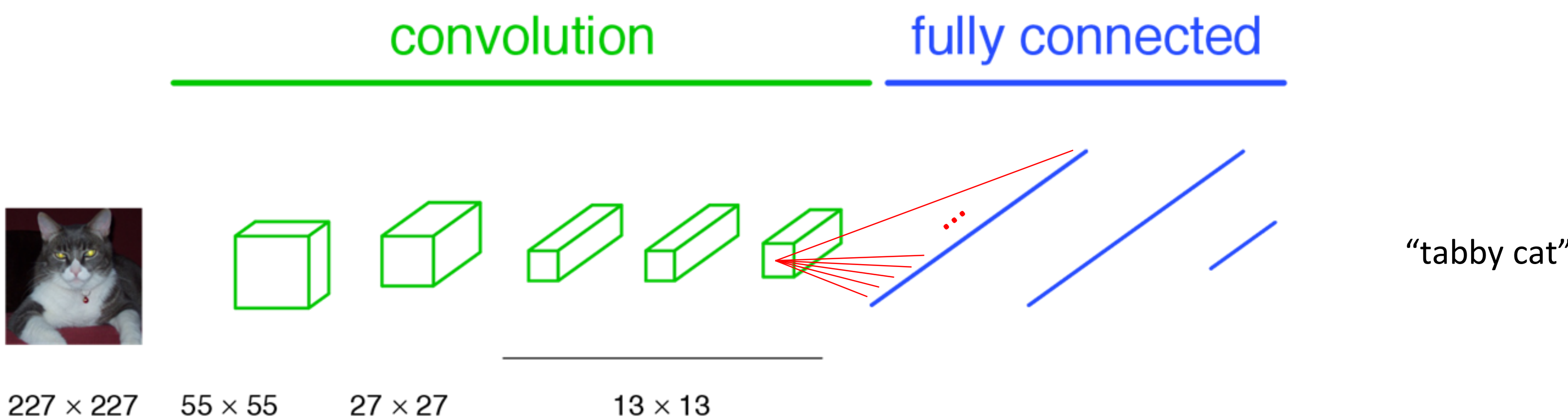
[Long et al.]

A classification network…



[Long et al.]

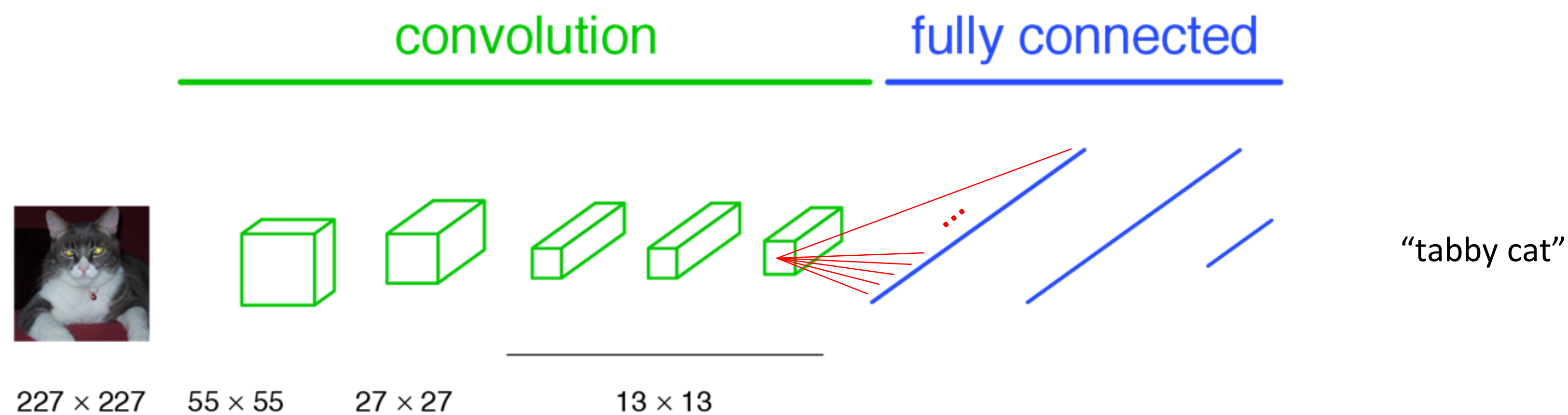
A classification network…



The response of every kernel across all positions are attached densely to the array of perceptrons in the fully-connected layer.

[Long et al.]

A classification network…



The response of every kernel across all positions are attached densely to the array of perceptrons in the fully-connected layer.

AlexNet: 256 filters over 6x6 response map
Each 2,359,296 response is attached to one of 4096 perceptrons, leading to 37 mil params.

[Long et al.]

Problem

- We want a label at every pixel
- Current network gives us a label for the whole image.
- We want a matrix of labels

- Approach:
    - Make CNN for sub-image size
    - ‘Convolutionalize’ all layers of network, so that we can treat it as one (complex) filter and slide around our full image.

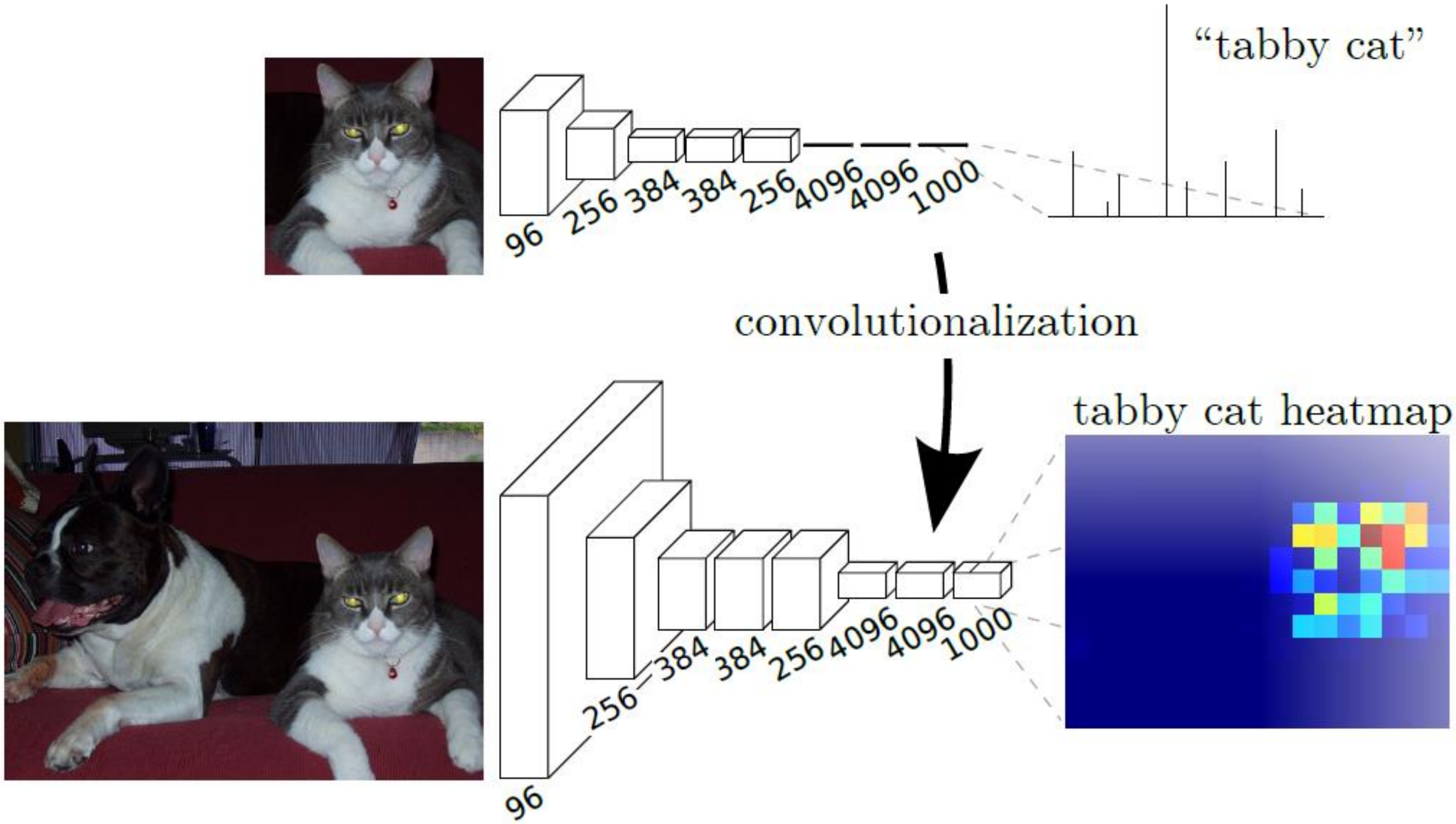
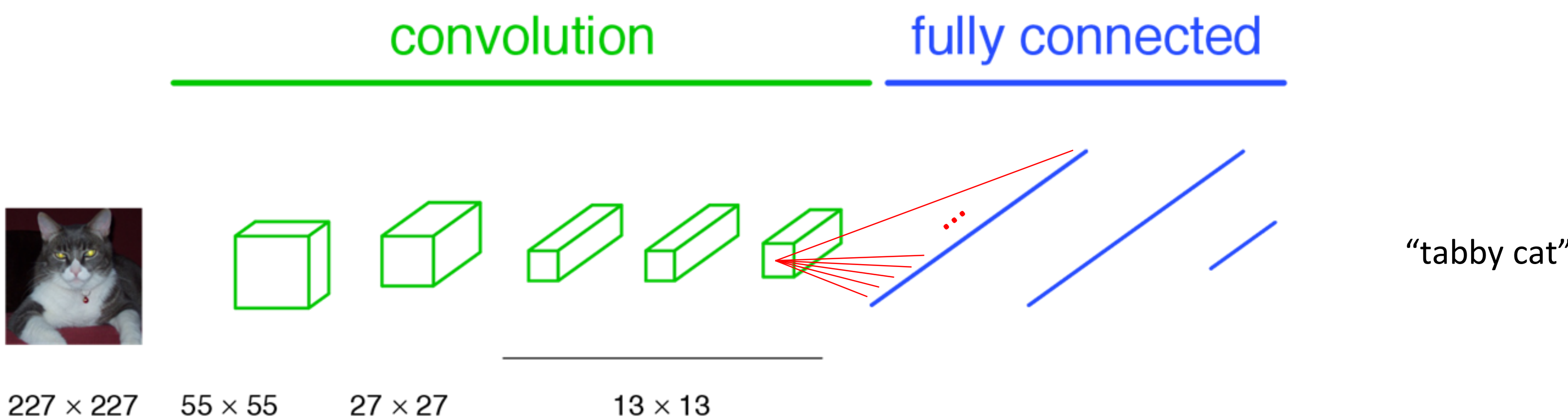


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

Long, Shelhamer, and Darrell 2014

A classification network…



The response of every kernel across all positions are attached densely to the array of perceptrons in the fully-connected layer.

AlexNet: 256 filters over 6x6 response map
Each 2,359,296 response is attached to one of 4096 perceptrons, leading to 37 mil params.

[Long et al.]

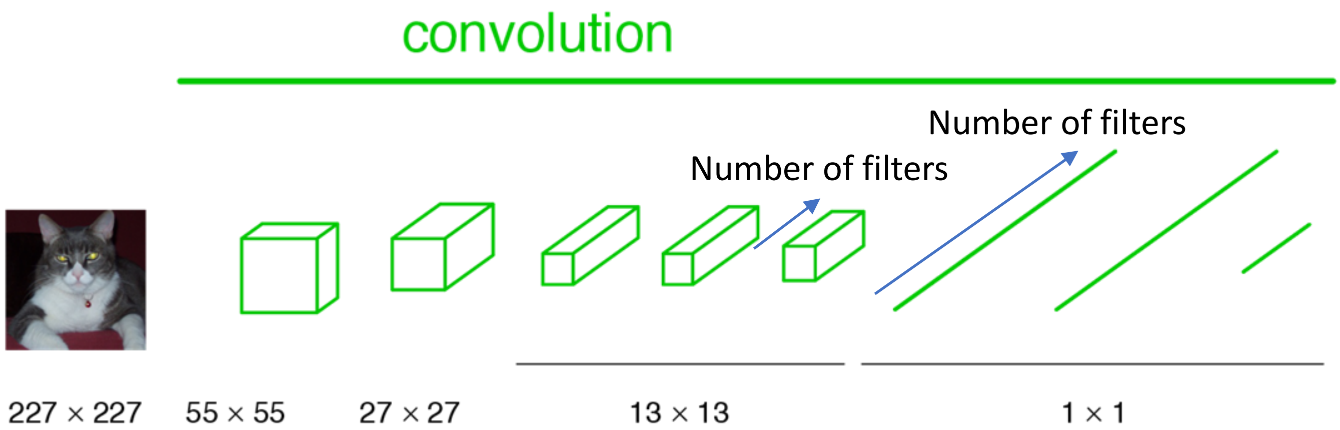
Yann LeCun

6 April 2015 ·

Follow

In Convolutional Nets, there is no such thing as "fully-connected layers". There are only convolution layers with 1x1 convolution kernels and a full connection table.

Convolutionalization



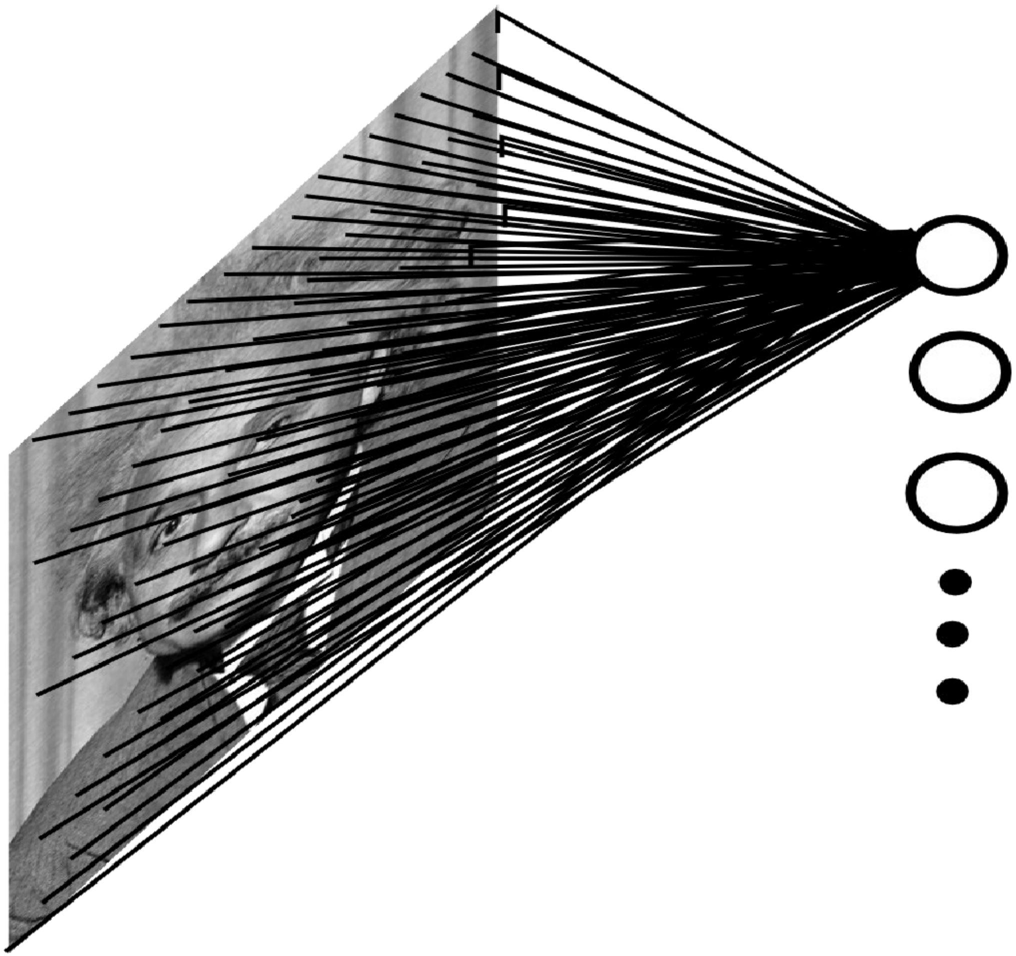
1x1 convolution operates across all filters in the previous layer, and is slid across all positions.

[Long et al.]

Back to the fully-connected perceptron…

$$\text{output} = \begin{cases} 0 & \text{if } w \cdot x \quad \leq 0 \\ 1 & \text{if } w \cdot x \quad > 0 \end{cases}$$

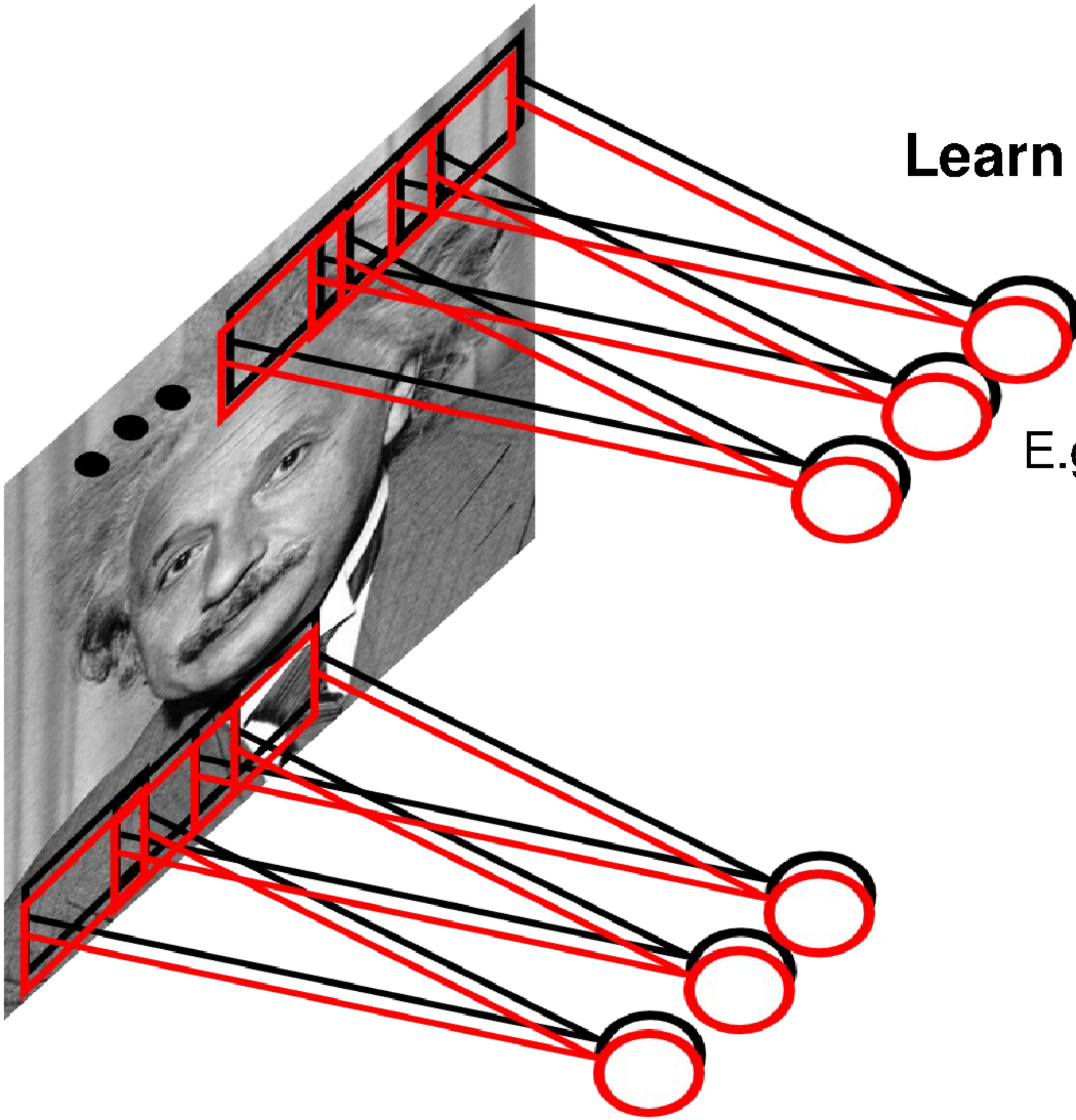
$$w \cdot x \equiv \sum_j w_j x_j,$$



Perceptron is connected to every value in the previous layer (across all channels; 1 visible).

[Long et al.]

Convolutional Layer

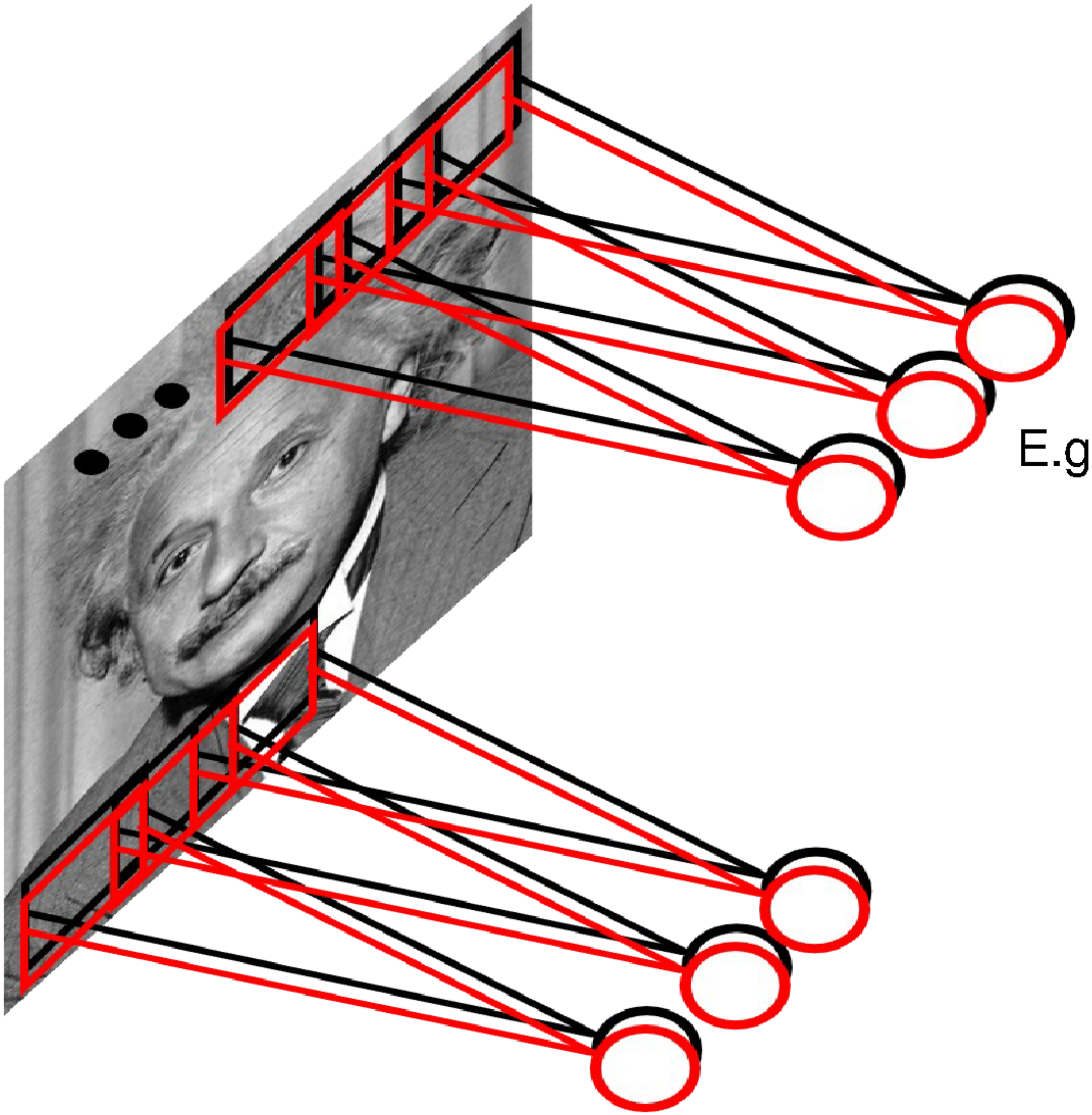


Learn multiple filters.

E.g.: 200x200 image
100 Filters
Filter size: 10x10
10K parameters

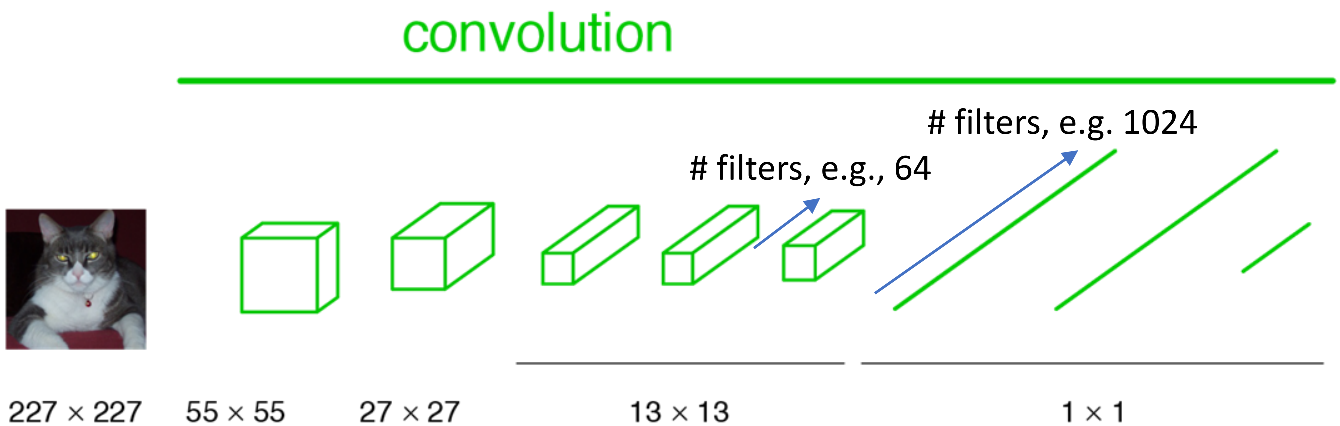
Ranzato

Convolutional Layer



E.g.: 200x200 image
100 Filters
Filter size: 1x1
100 parameters

Convolutionalization

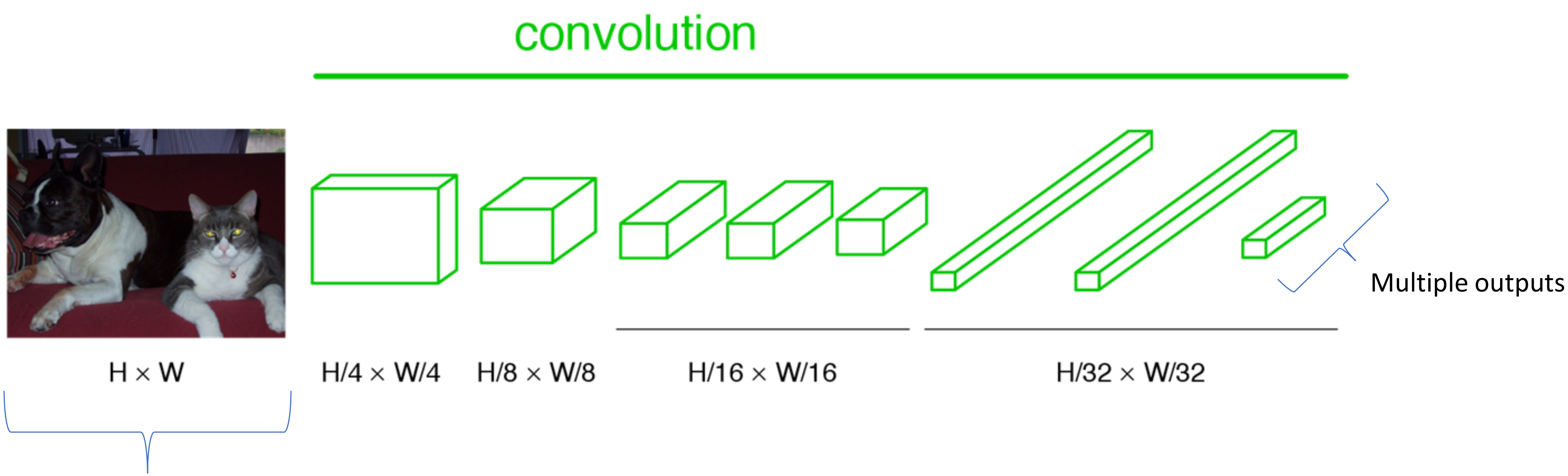


1x1 convolution operates across all filters in the previous layer, and is slid across all positions.

e.g., 64x1x1 kernel, with shared weights over 13x13 output, x1024 filters = 11mil params.

[Long et al.]

Becoming fully convolutional



Arbitrary-sized image

When we turn these operations into a convolution, the 13x13 just becomes another parameter and our output size adjust dynamically.

Now we have a *vector/matrix* output, and our network acts itself like a complex filter.

[Long et al.]

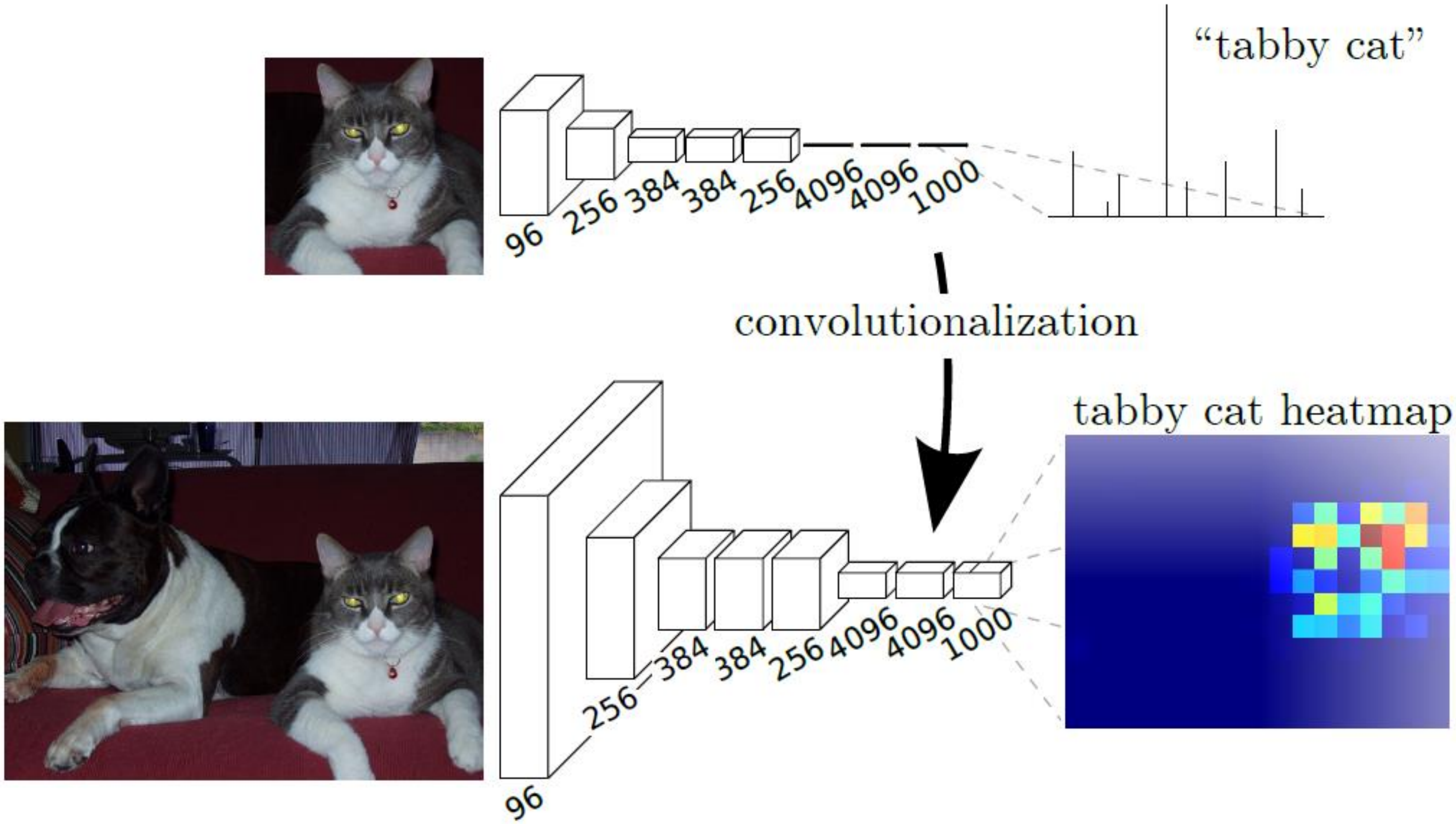
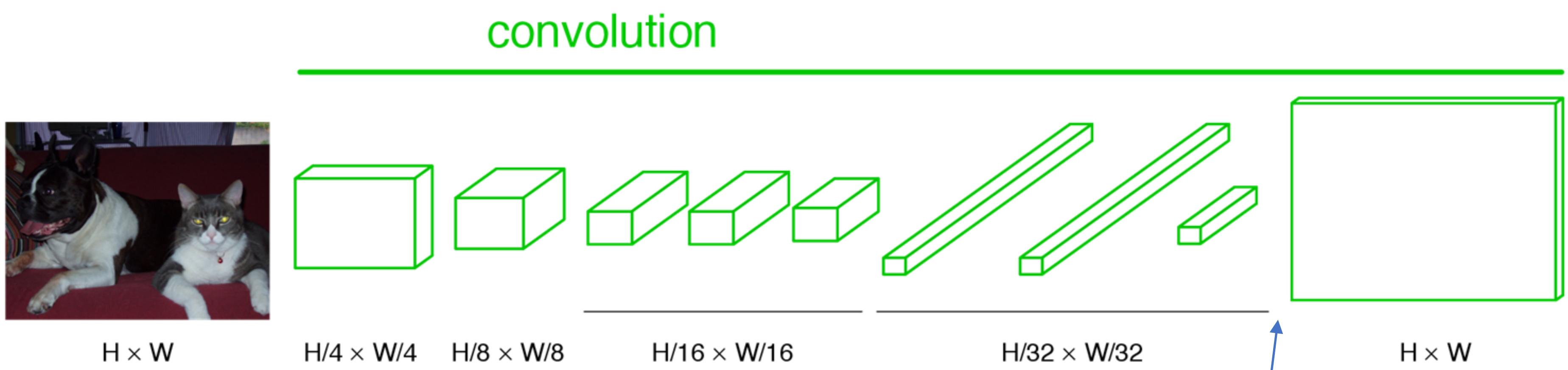


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

Long, Shelhamer, and Darrell 2014

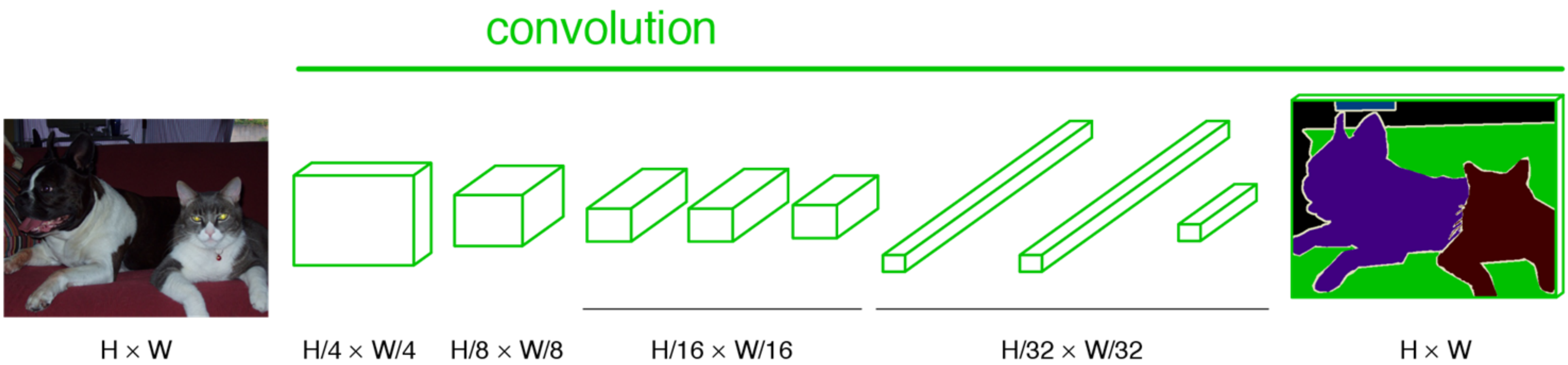
Upsampling the output



Some upsampling algorithm to return us to H x W

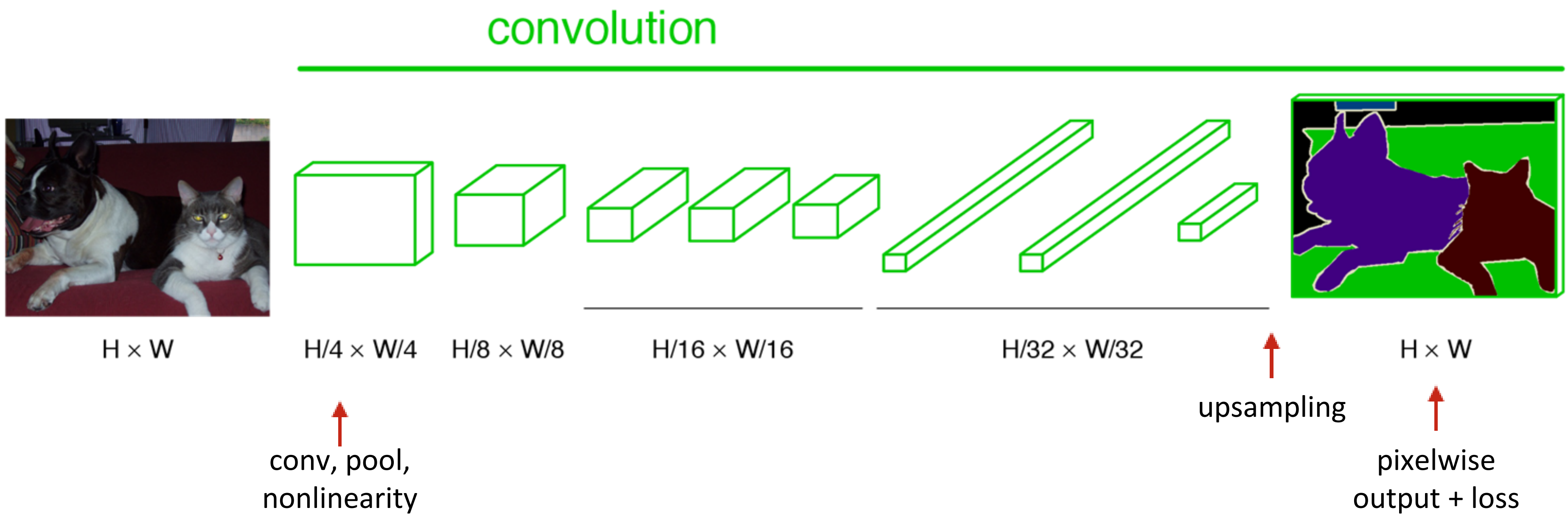
[Long et al.]

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



[Long et al.]

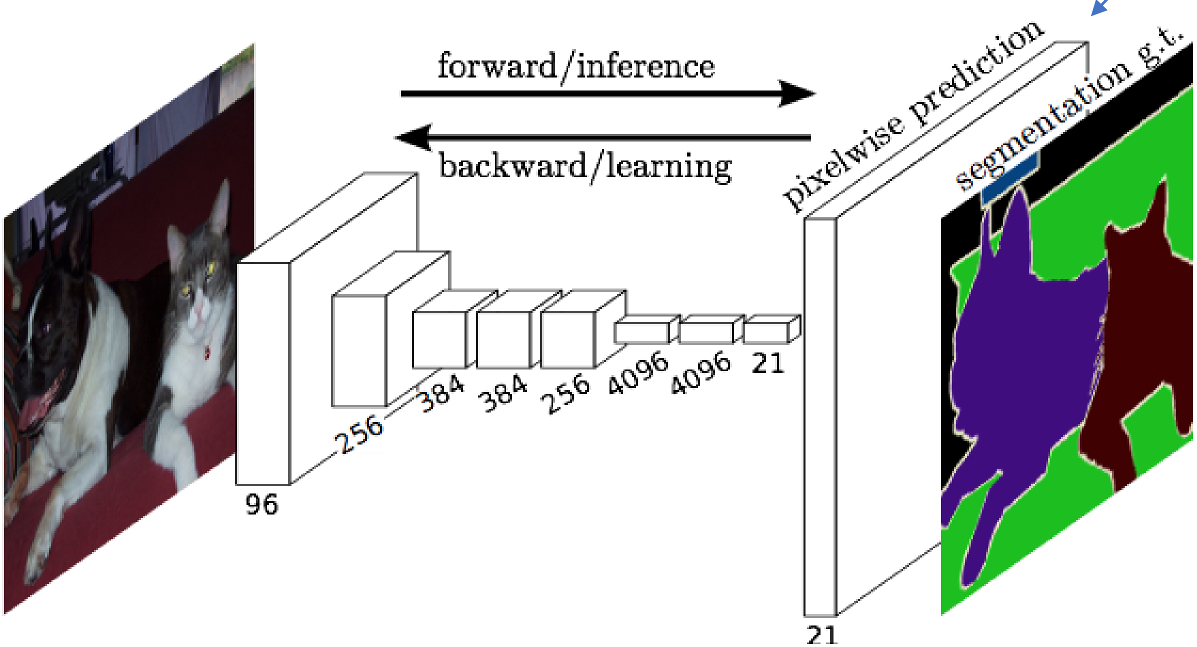
End-to-end, pixels-to-pixels network



[Long et al.]

What is the upsampling layer?

This one.

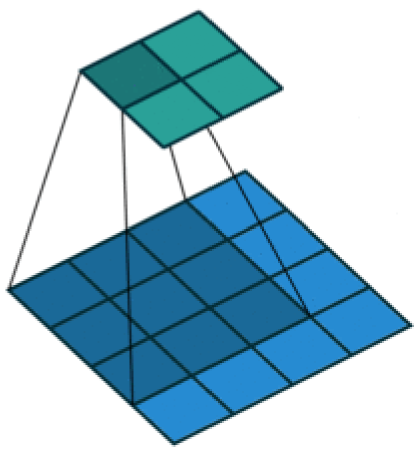


Hint: it's actually an upsampling _network_

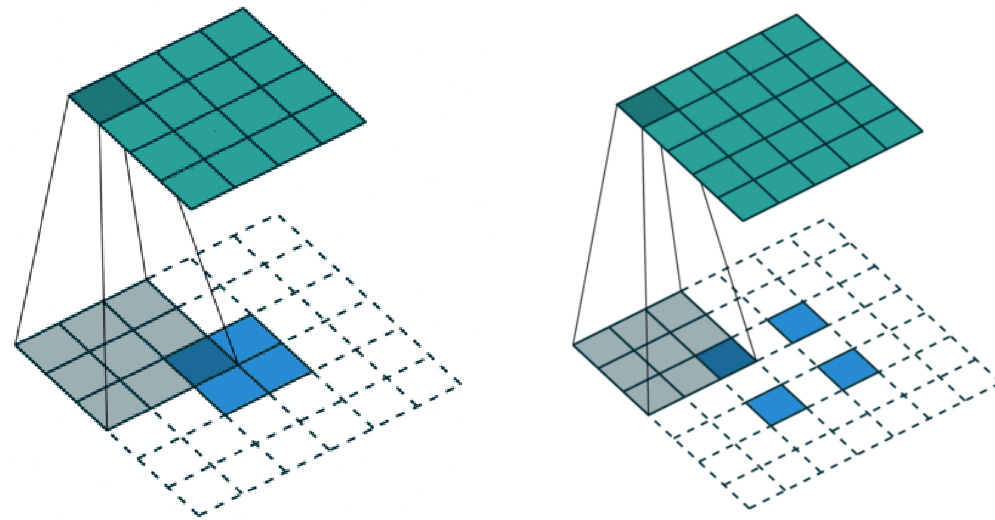
[Long et al.]

Upsampling with convolution

Convolution



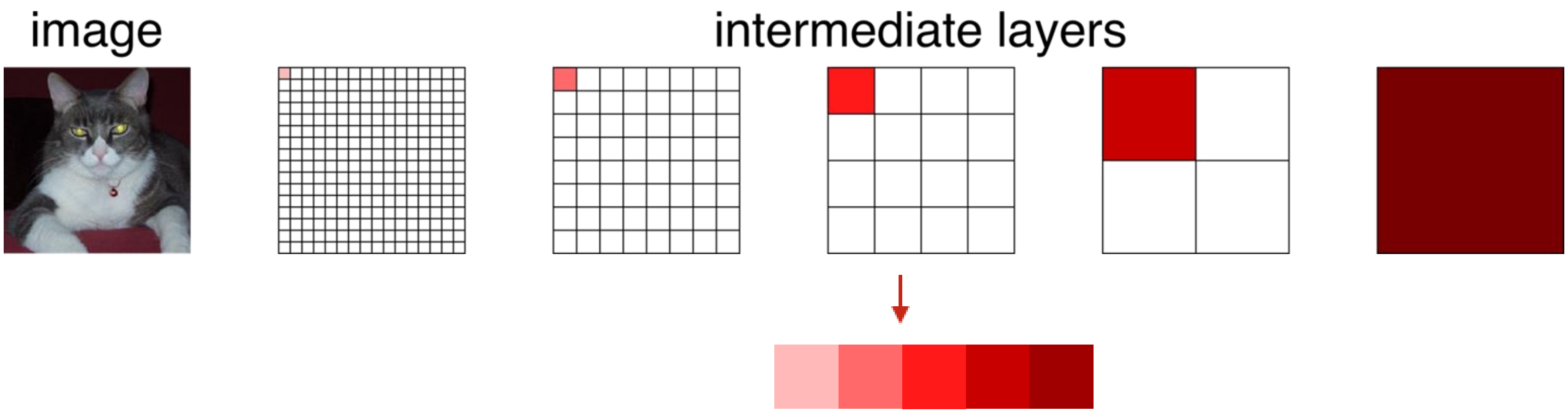
Transposed convolution =
weighted kernel ‘stamp’



Often called “deconvolution”,
but not actually the deconvolution
that we previously saw in deblurring ->
that is division in the Fourier domain.

Spectrum of deep features

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

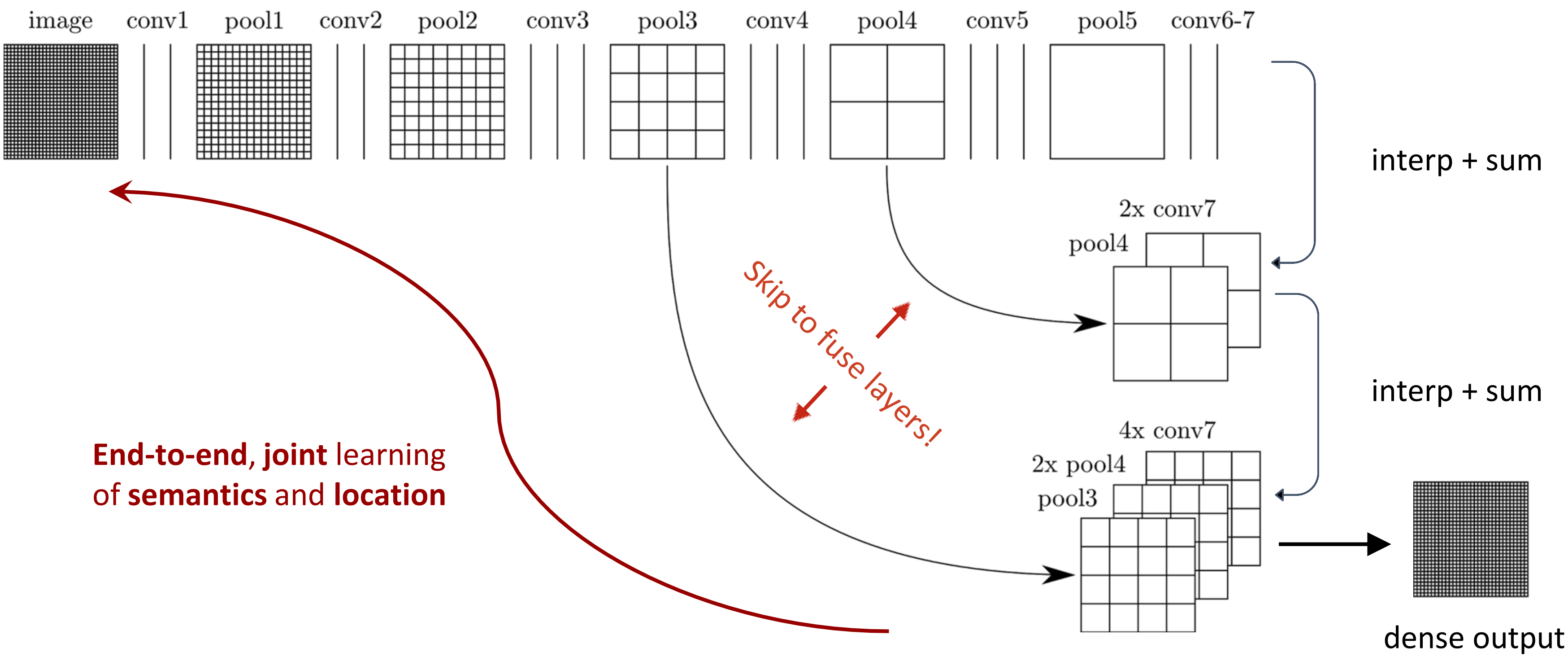


Fuse features into **deep jet**

(cf. Hariharan et al. CVPR15 “hypercolumn”)

[Long et al.]

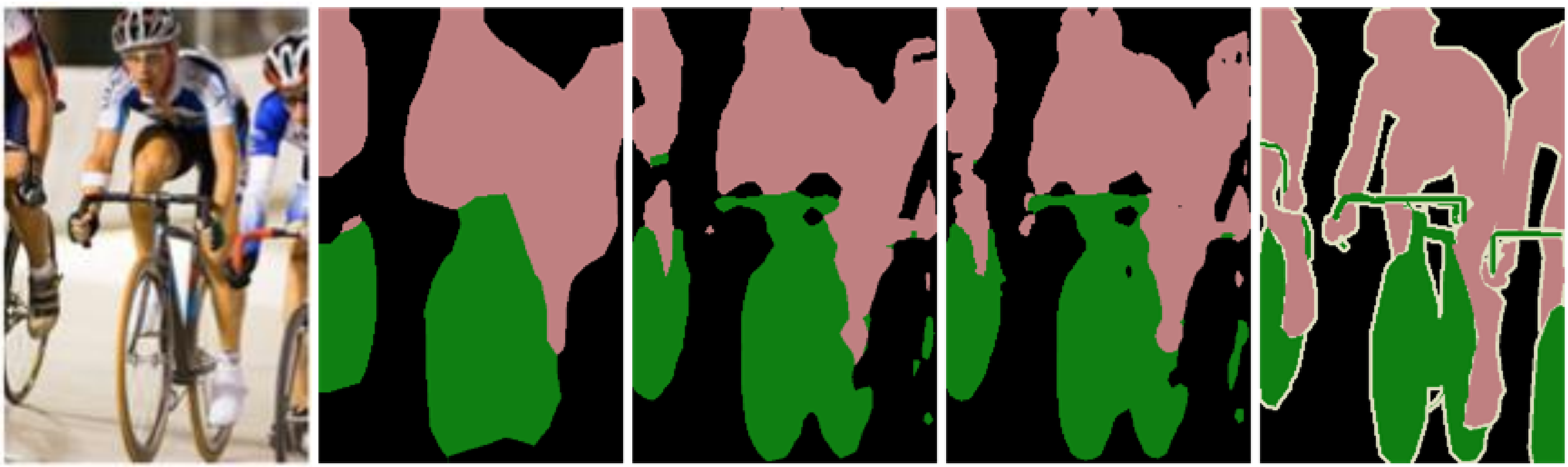
Learning upsampling kernels with skip layer refinement



[Long et al.]

Skip layer refinement

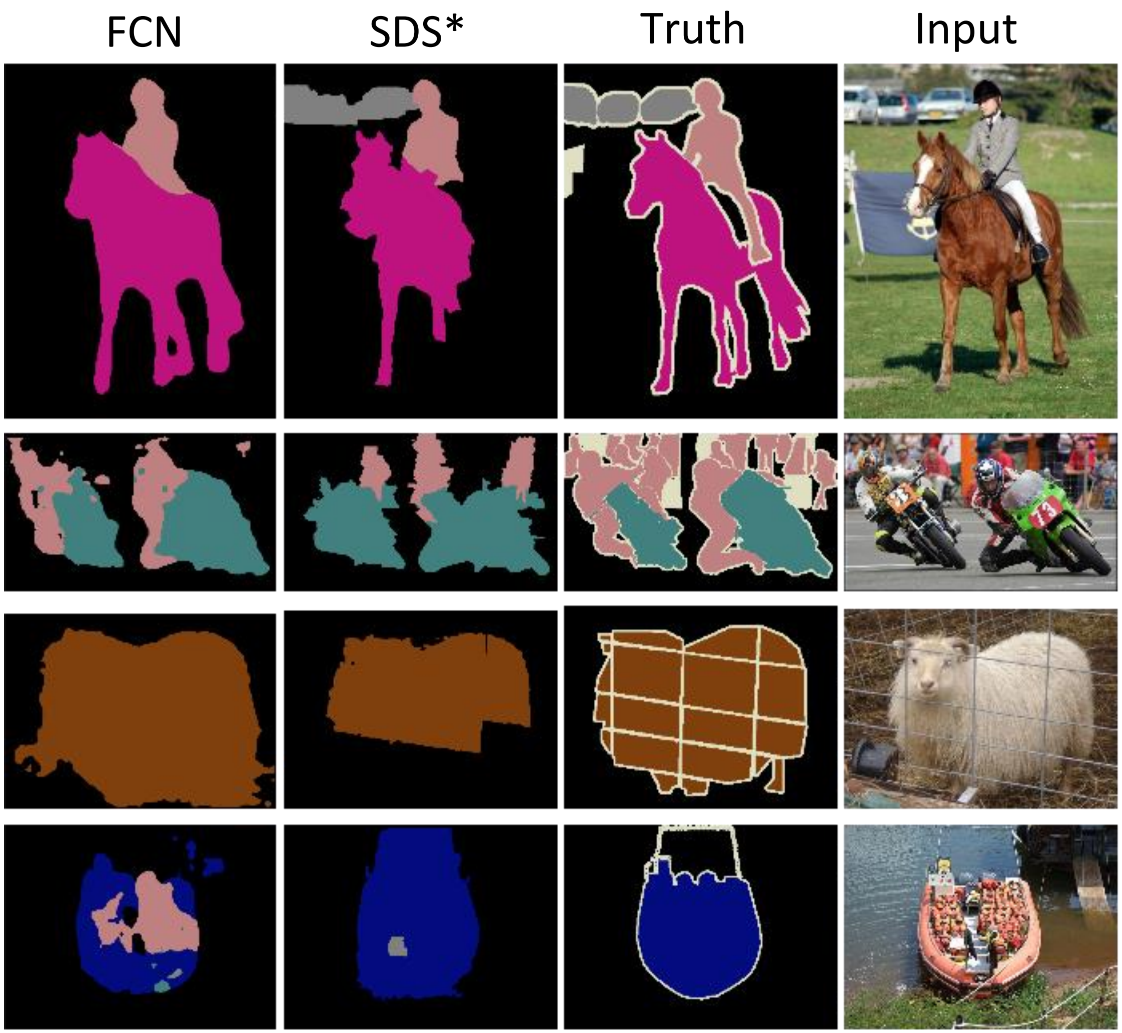
input image | stride 32 | stride 16 | stride 8 | ground truth



no skips | 1 skip | 2 skips

[Long et al.]

Results

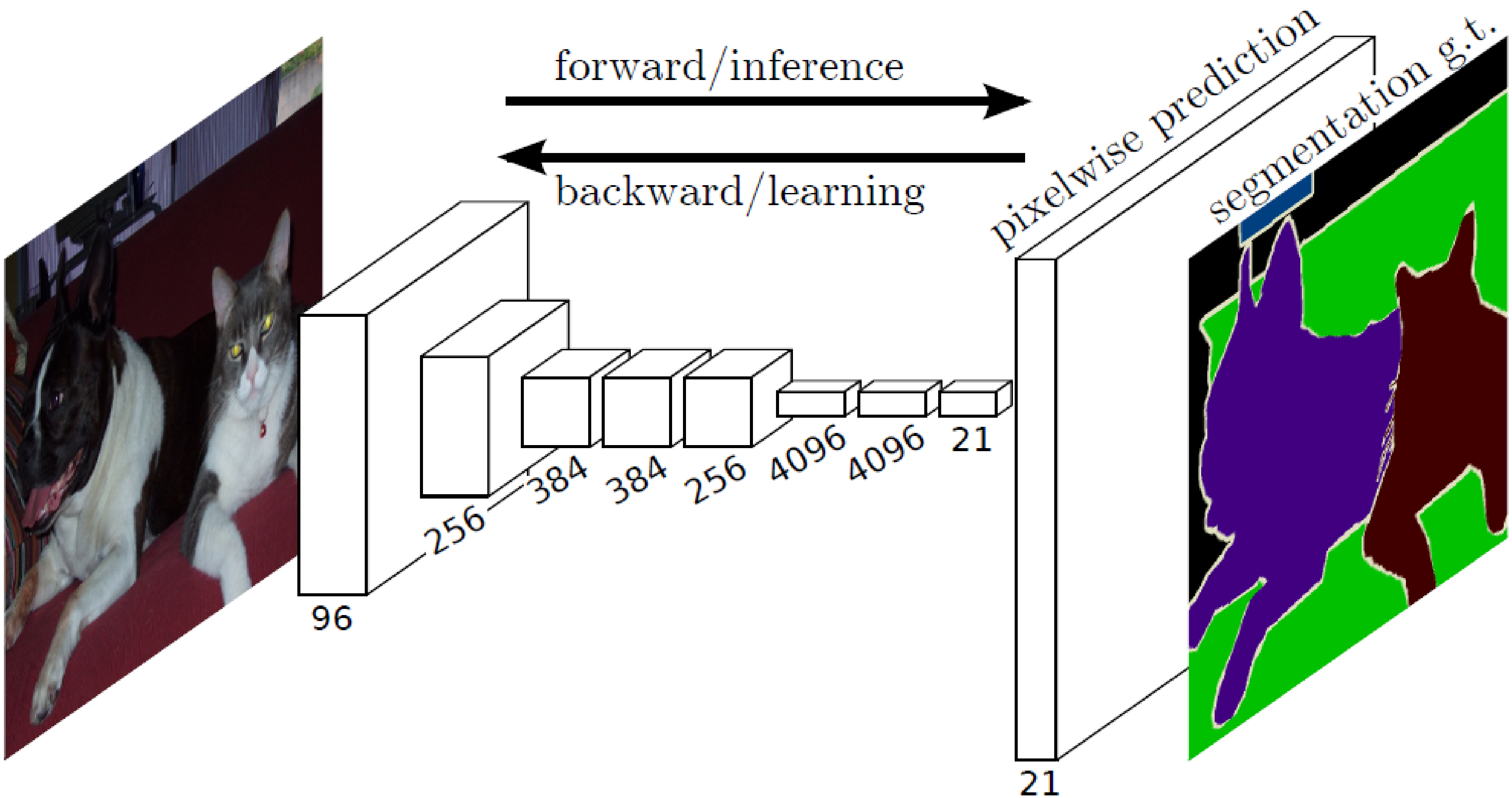


Relative to prior state-of-the-art SDS:

- 30% relative improvement for mean IoU
- 286× faster

*Simultaneous Detection and Segmentation Hariharan et al. ECCV14

[Long et al.]



What can we do with an FCN?

Long, Shelhamer, and Darrell 2014

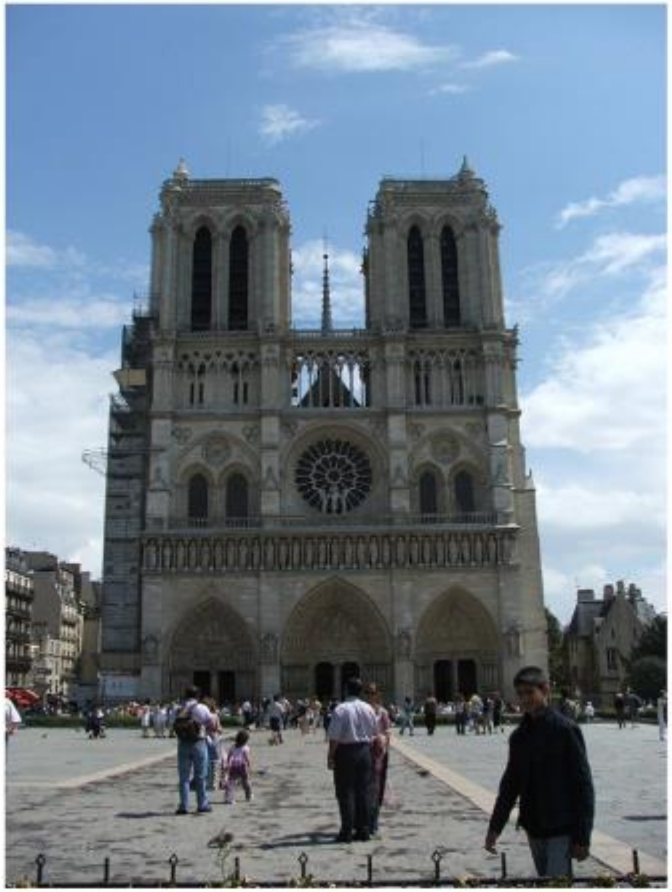
How much can an image tell about its geographic location?

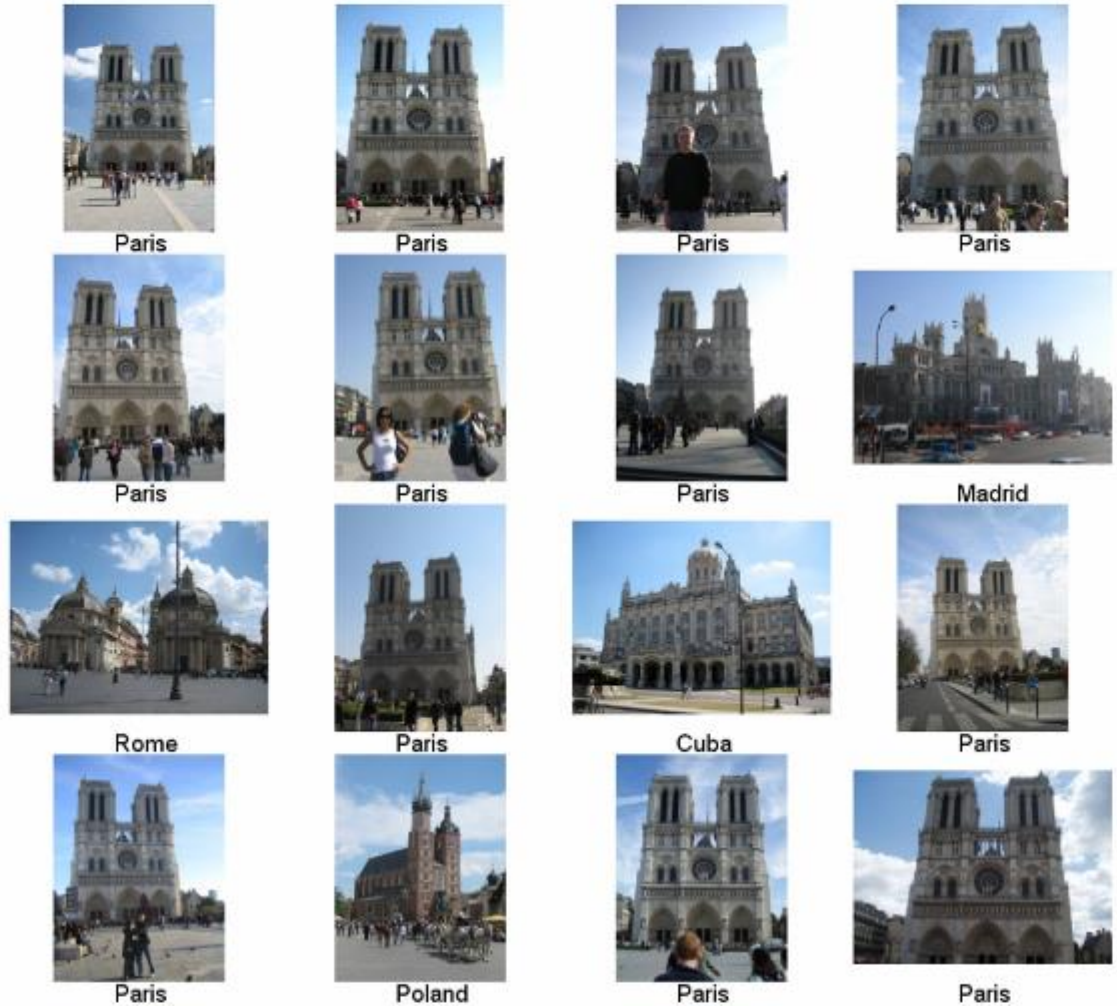
6 million geo-tagged Flickr images

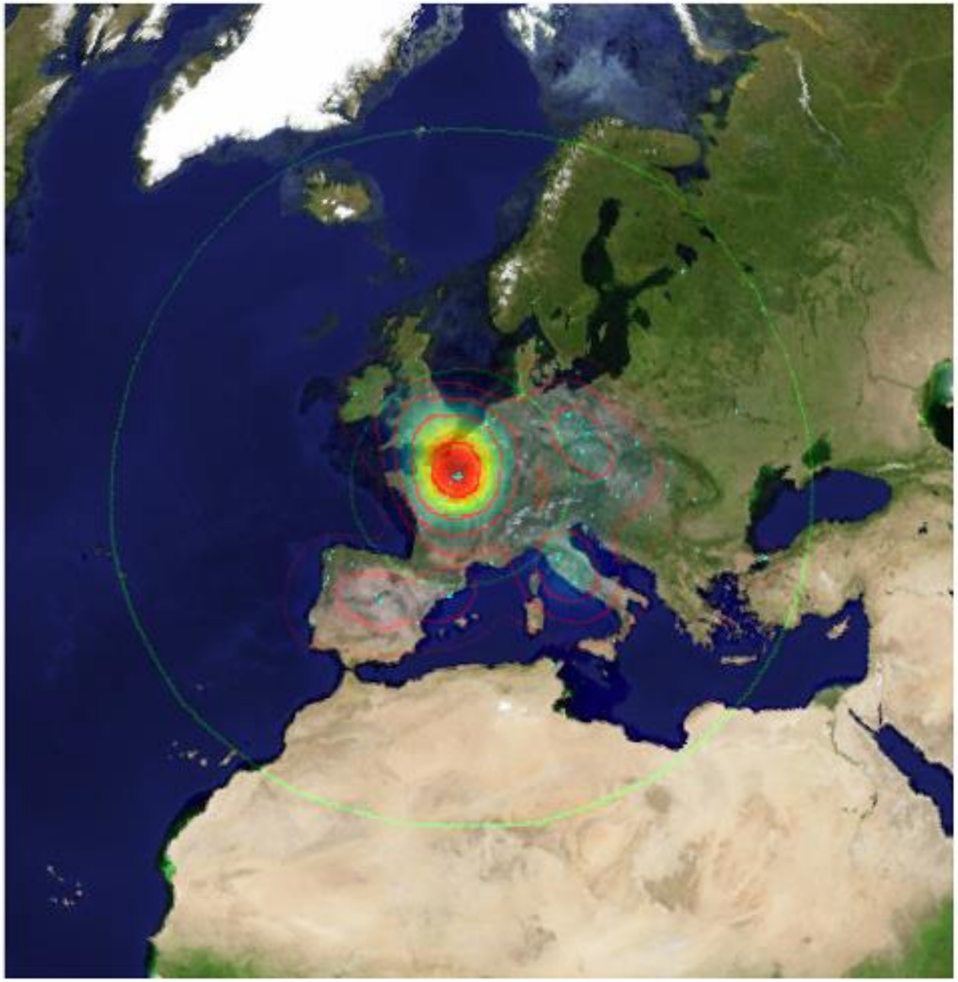
http://graphics.cs.cmu.edu/projects/im2gps/

im2gps (Hays & Efros, CVPR 2008)

Nearest Neighbors according to gist + bag of SIFT + color histogram + a few others







PlaNet - Photo Geolocation with Convolutional Neural Networks

Tobias Weyand, Ilya Kostrikov, James Philbin

ECCV 2016

Discretization of Globe

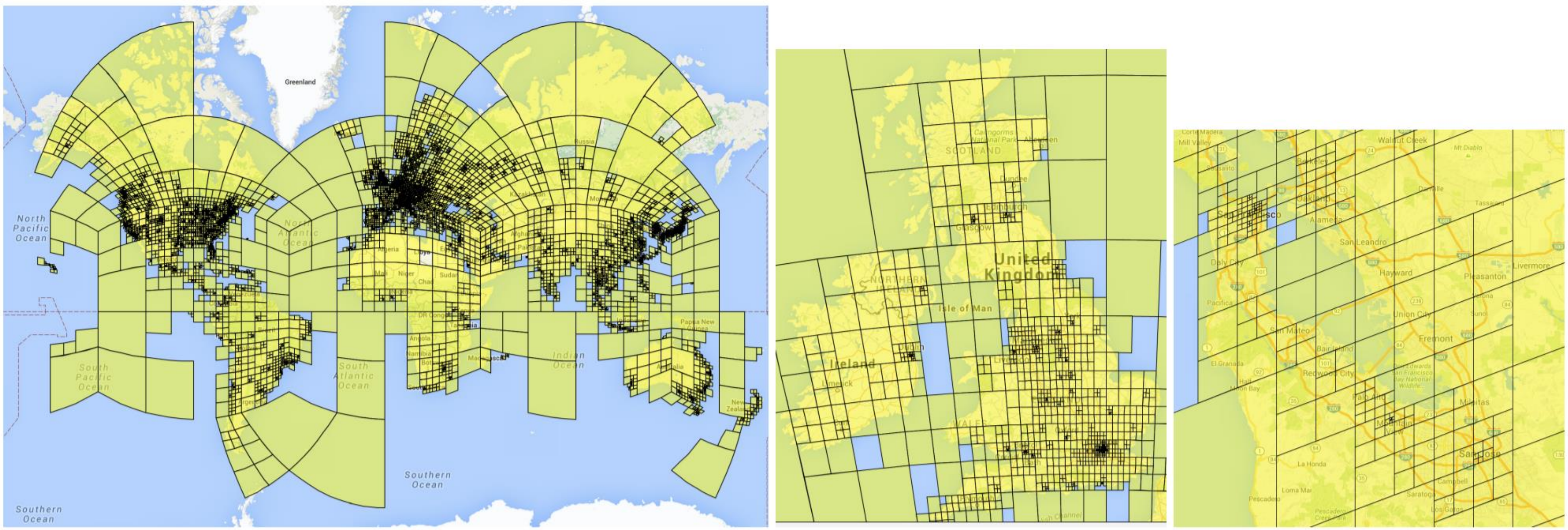


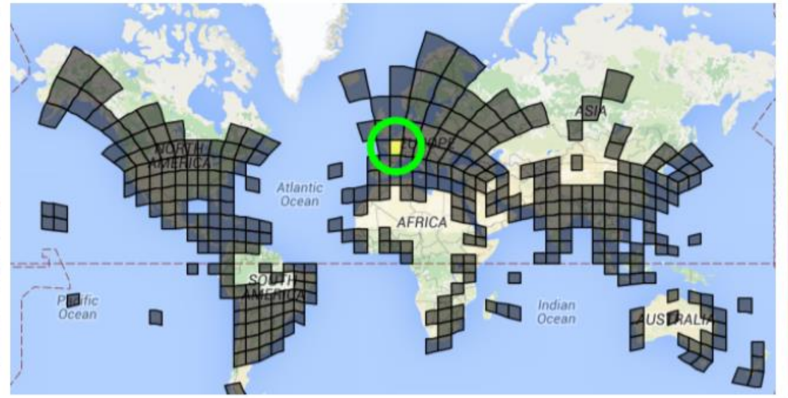
Figure 2. Left: Adaptive partitioning of the world into 26,263 S2 cells. Right: Detail views of Great Britain and Ireland and the San

Network and Training

- Network Architecture: Inception with 97M parameters
- 26,263 “categories” – places in the world

- 126 Million Web photos
- 2.5 months of training on 200 CPU cores

Photo CC-BY-NC by stevekc



(a)

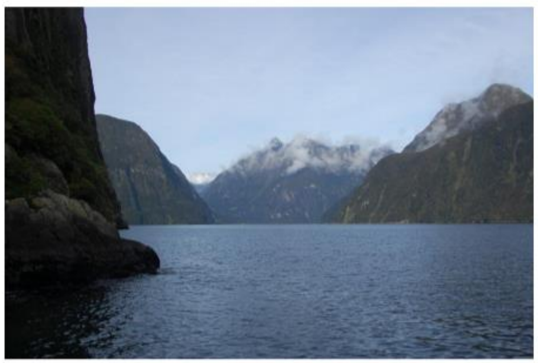
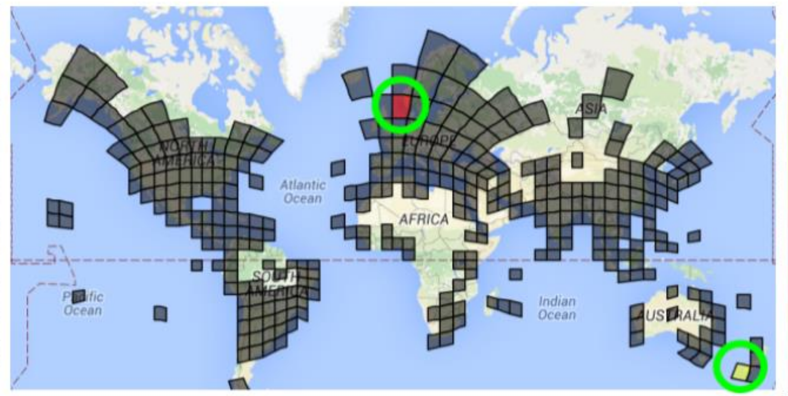


Photo CC-BY-NC by edwin.11



(b)

Photo CC-BY-NC by jonathanfh

Namibia / Botswana


Photo by jamie.lovelock / CC BY NC


Photo by MongoosePhotography / CC BY NC


Photo by Mister-E / CC BY NC


Photo by dalangalma / CC BY NC


Photo by siamjack / CC BY NC

Kauai, Hawaii

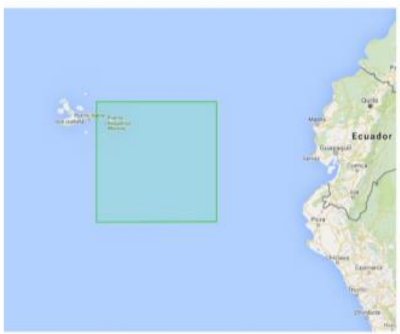

Photo by ryan + sarah / CC BY NC


Photo by stuartlchambers / CC BY NC


Photo by samgrover / CC BY NC


Photo by steuben / CC BY NC


Photo by steve-stevens / CC BY NC



Galapagos Islands

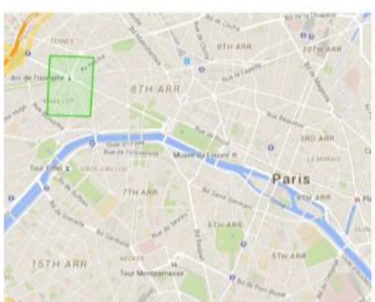

Photo by p.j.k. / CC BY NC


Photo by victor408 / CC BY NC


Photo by Domen Jakus / CC BY NC


Photo by cvanholder / CC BY NC


Photo by rwoan / CC BY NC



Paris


Photo by feliven / CC BY NC


Photo by fred_v / CC BY NC


Photo by Turansa Tours / CC BY NC


Photo by JA_FS / CC BY NC

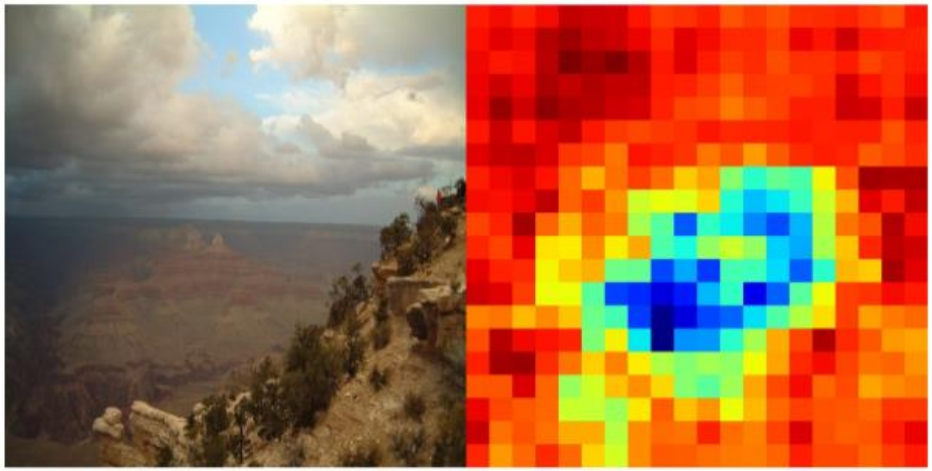

Photo by CedEm photographies / CC BY NC

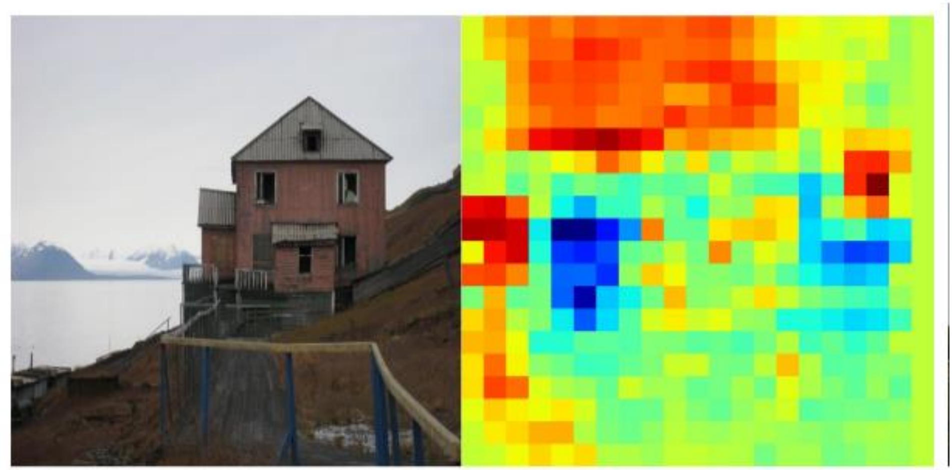
PlaNet vs im2gps (2008, 2009)

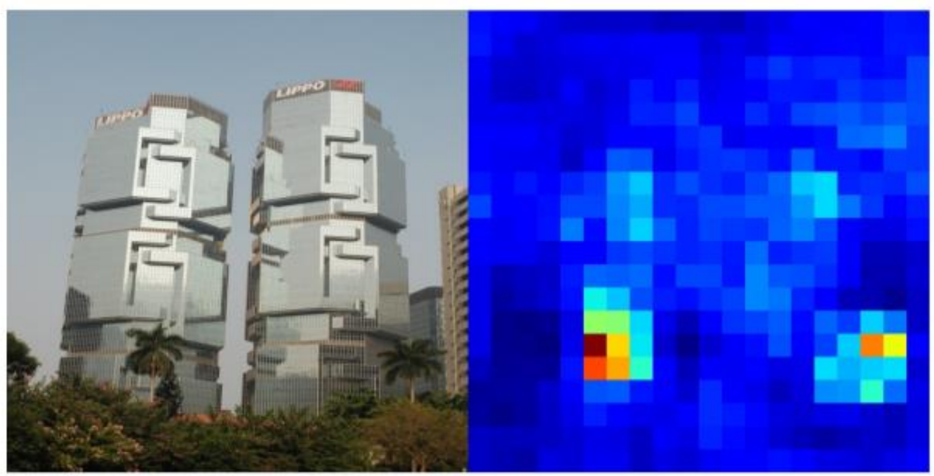
Method	Street 1 km	City 25 km	Region 200 km	Country 750 km	Continent 2500 km
Im2GPS (orig) [17]		12.0%	15.0%	23.0%	47.0%
Im2GPS (new) [18]	2.5%	21.9%	32.1%	35.4%	51.9%
PlaNet	**8.4%**	**24.5%**	**37.6%**	**53.6%**	**71.3%**

Method	Manmade Landmark	Natural Landmark	City Scene	Natural Scene	Animal
Im2GPS (new)	61.1	37.4	3375.3	5701.3	6528.0
PlaNet	74.5	61.0	212.6	1803.3	1400.0

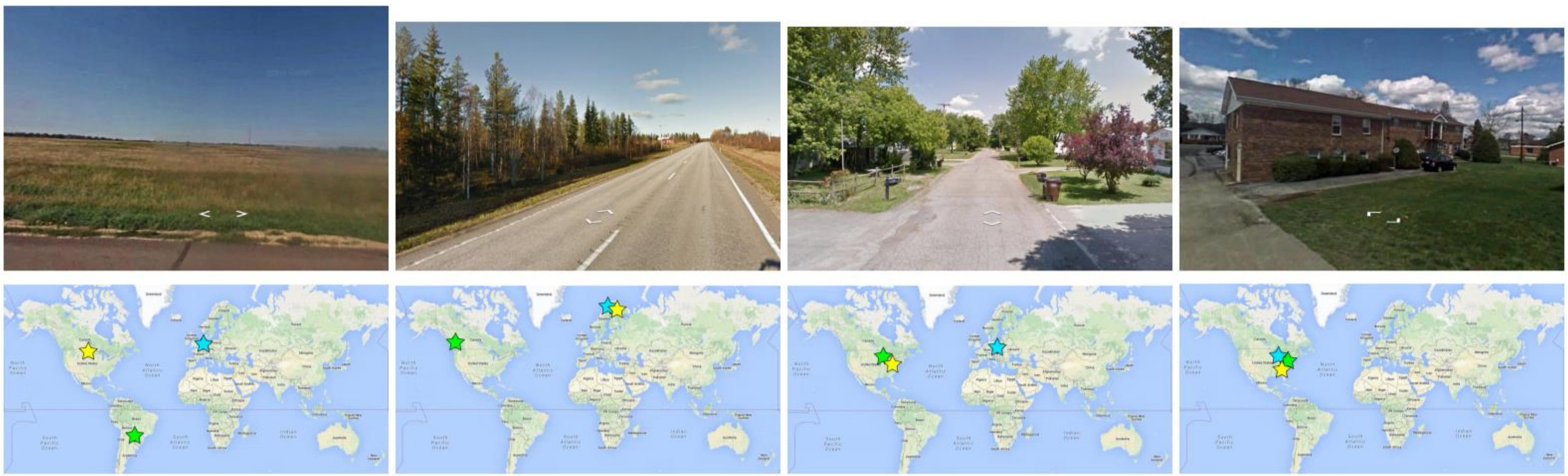
Spatial support for decision



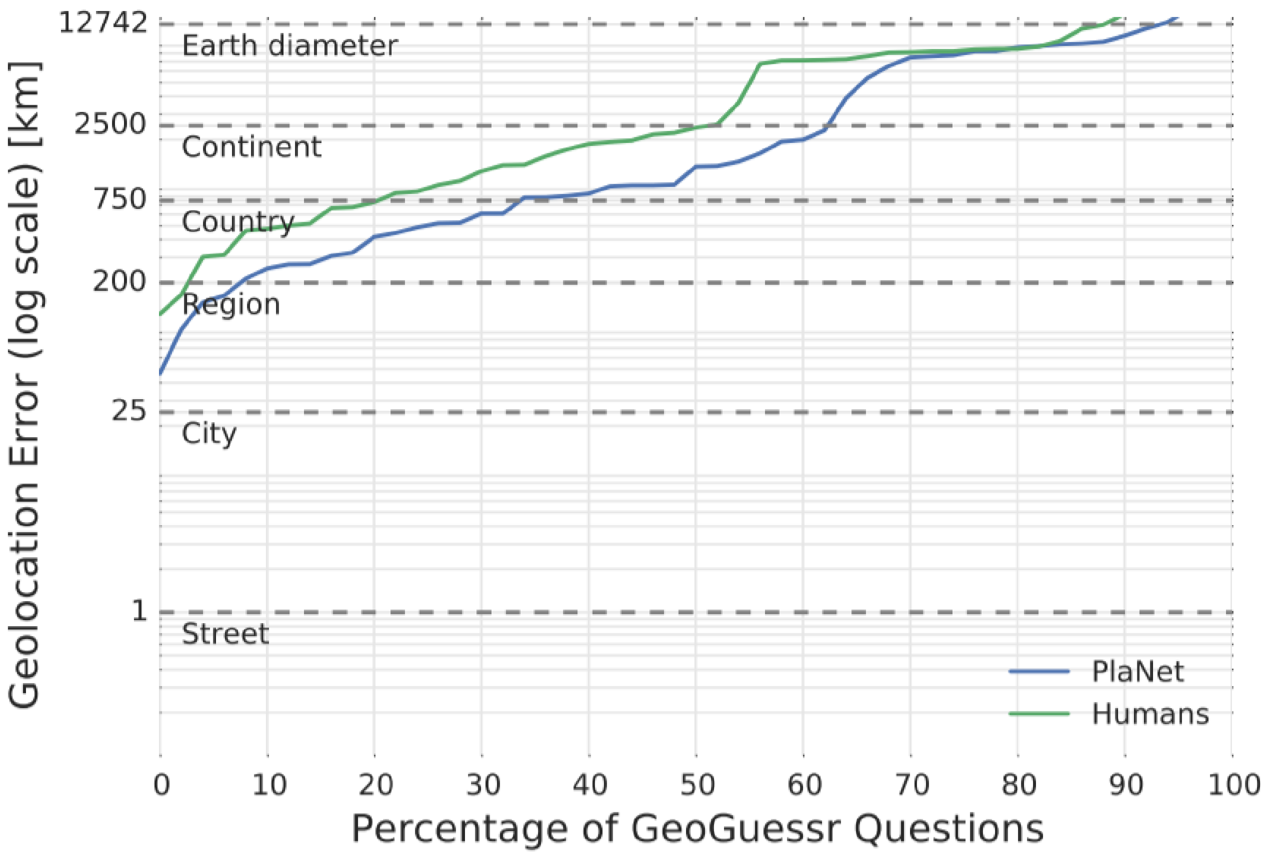




PlaNet vs Humans



PlaNet vs. Humans



PlaNet summary

- Very fast geolocalization method by categorization.
- Uses far more training data than previous work (im2gps)
- Better than humans!