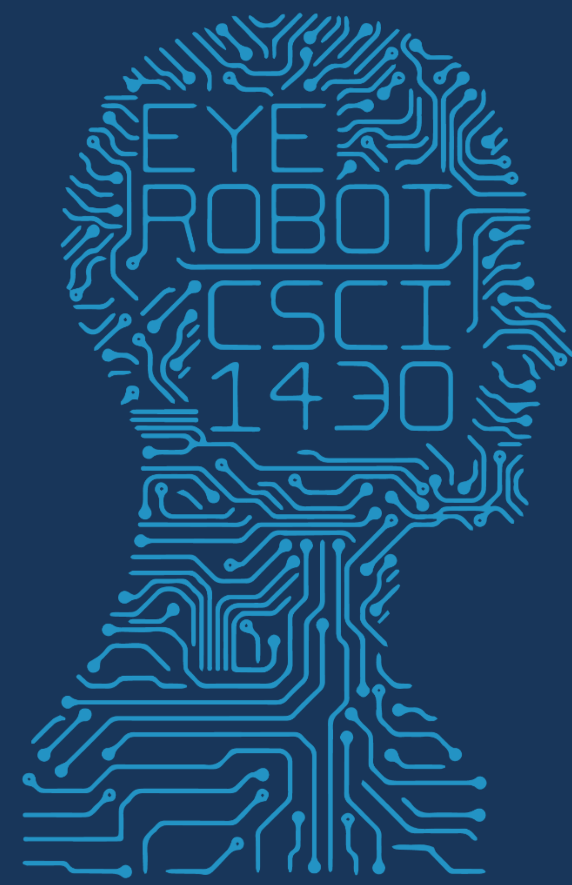
1950

Future Vision



2017 MWF 1pm 368

Computer Vision

Martian lava field, NASA, Wikipedia

Old Man of the Mountain, Franconia, New Hampshire

Pareidolia

http://smrt.ccel.ca/2013/12/16/pareidolia/

Reddit for more :)
https://www.reddit.com/r/Pareidolia/top/

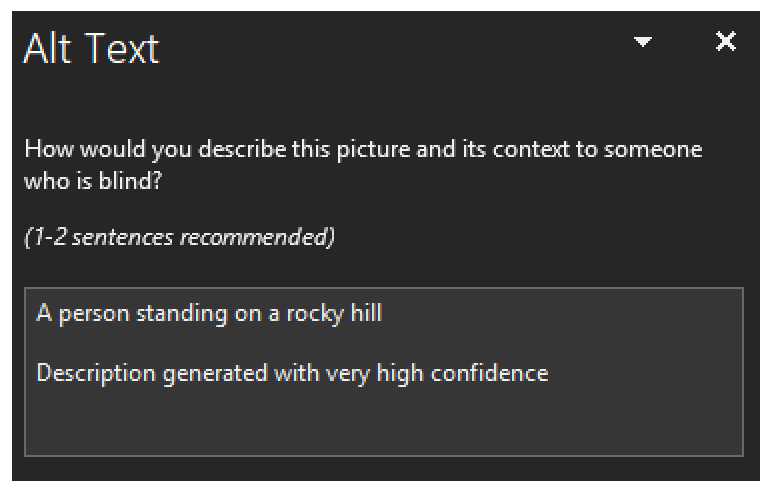
Pareidolia

Seeing things which aren't really there…

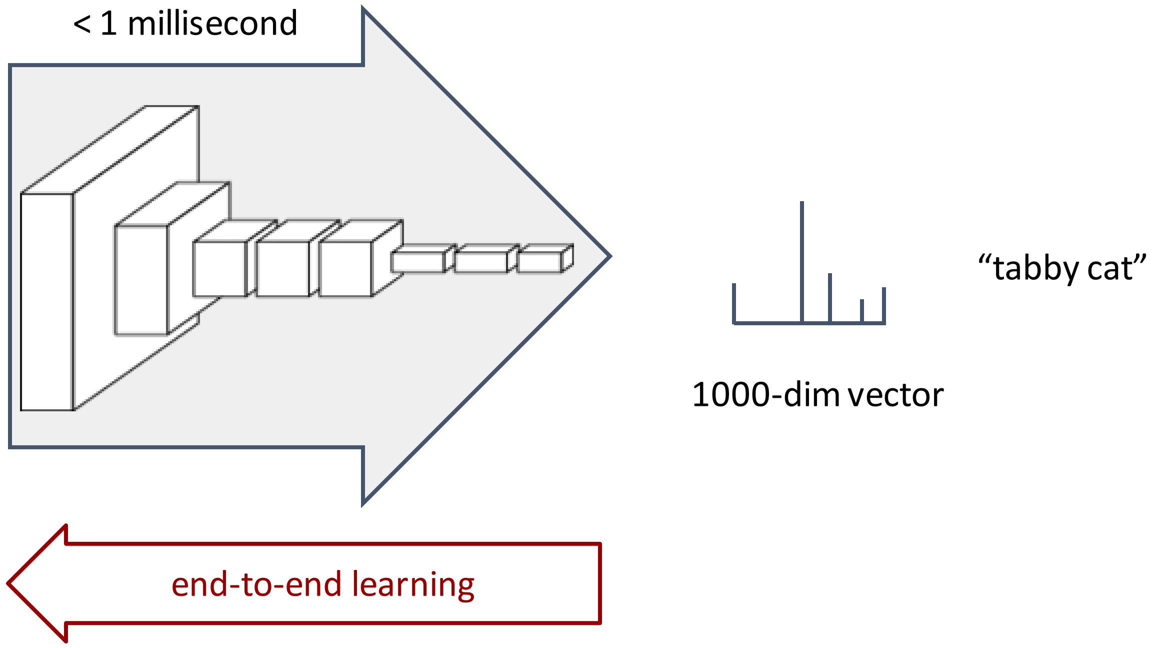
DeepDream as reinforcement pareidolia

Powerpoint Alt-text Generator

Vision-based
caption generator



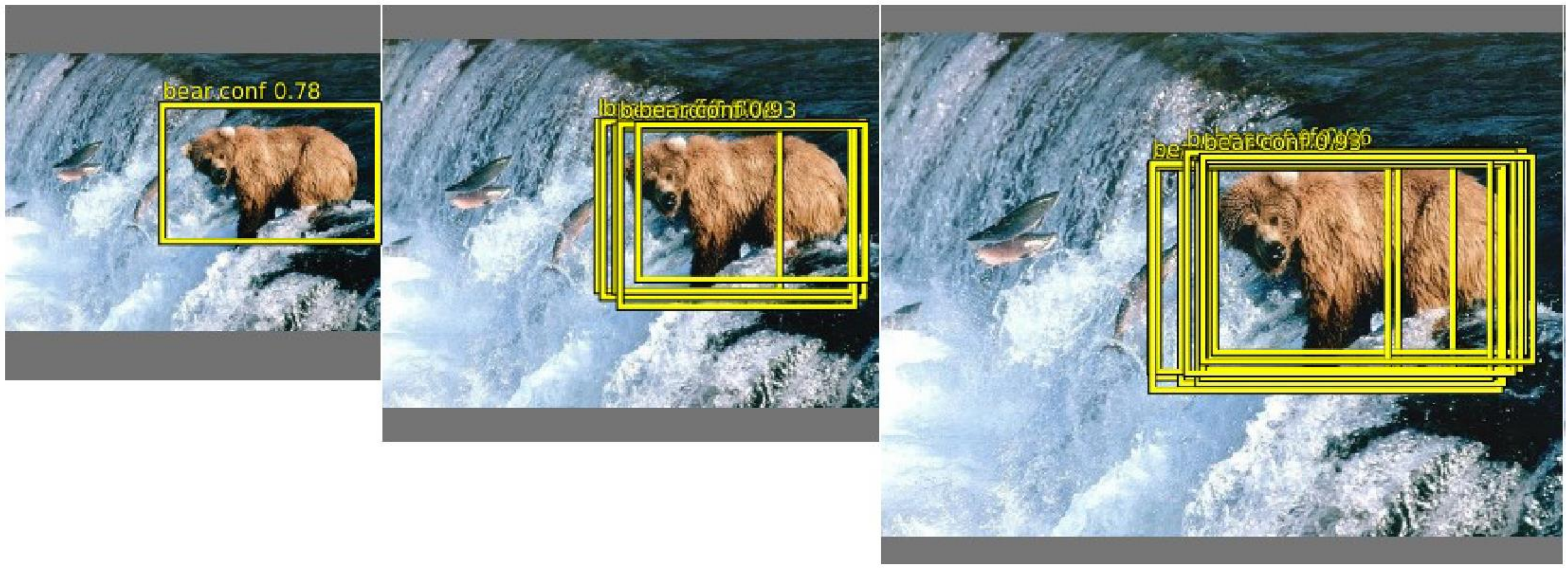
ConvNets perform classification



[Slides from Long, Shelhamer, and Darrell]

CONV NETS: EXAMPLES

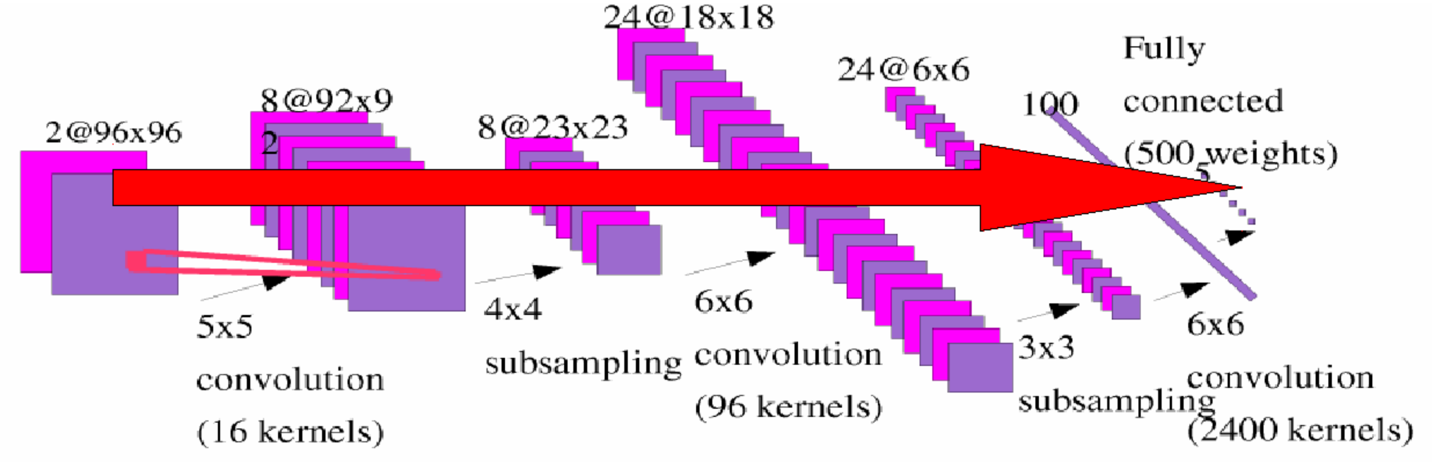
- **Object detection**



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

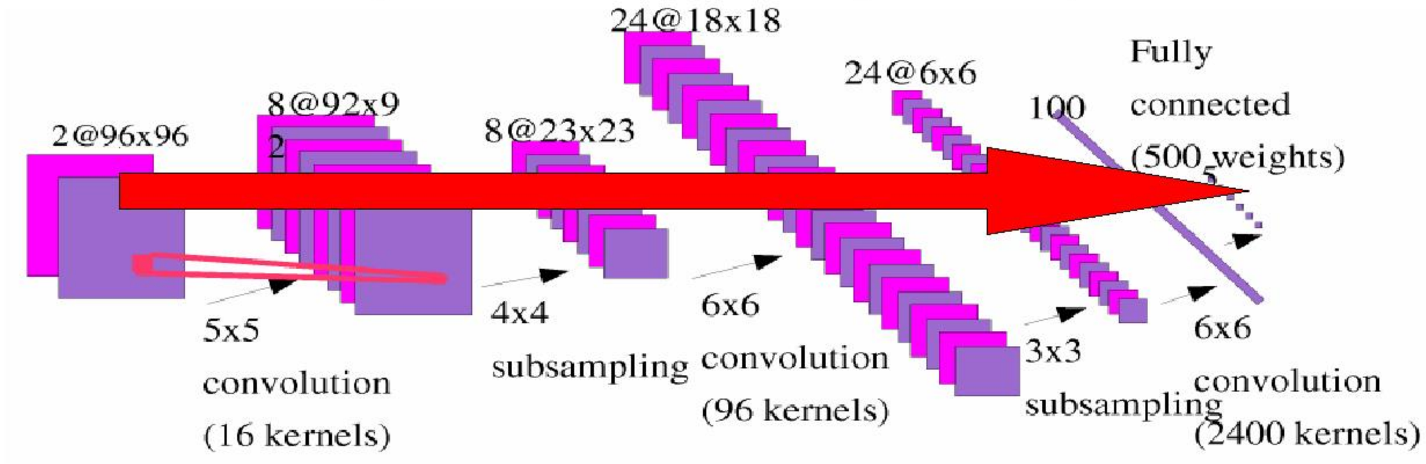
ConvNets: Test

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

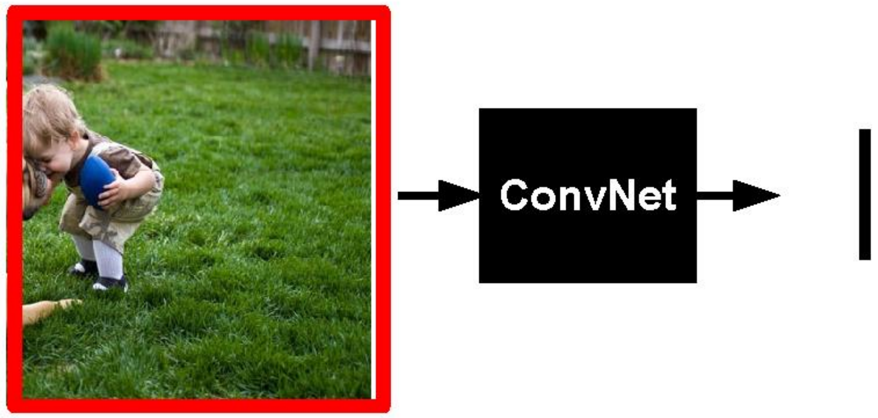


ConvNets: Test

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



Naturally, convnet can process larger images

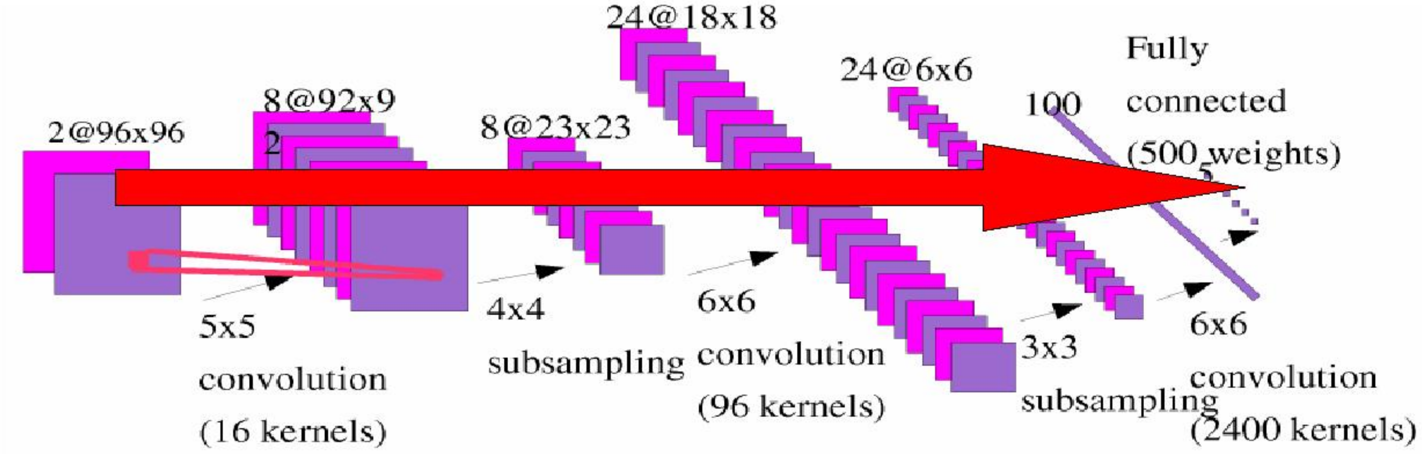


Traditional methods use inefficient sliding windows.

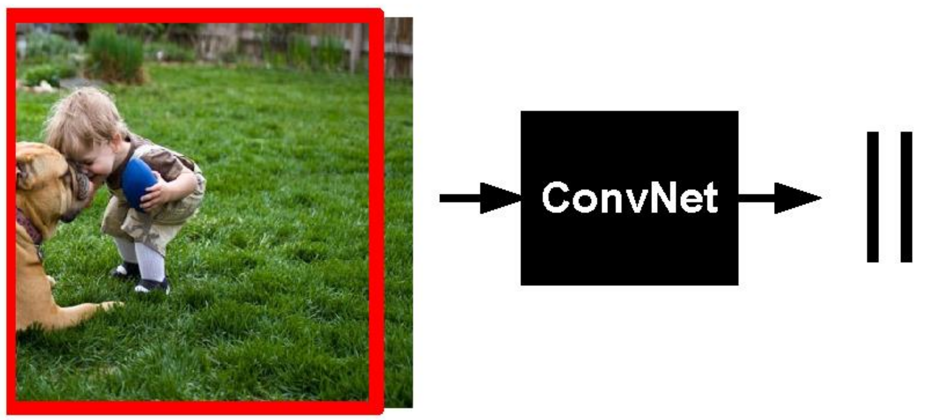
Ranzato

ConvNets: Test

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



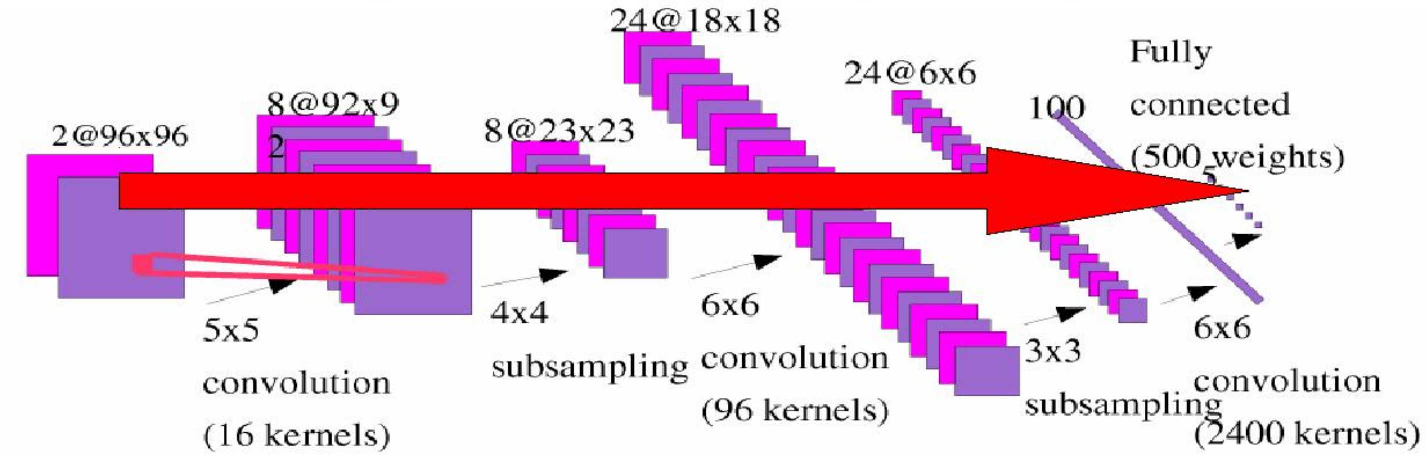
Naturally, convnet can process larger images



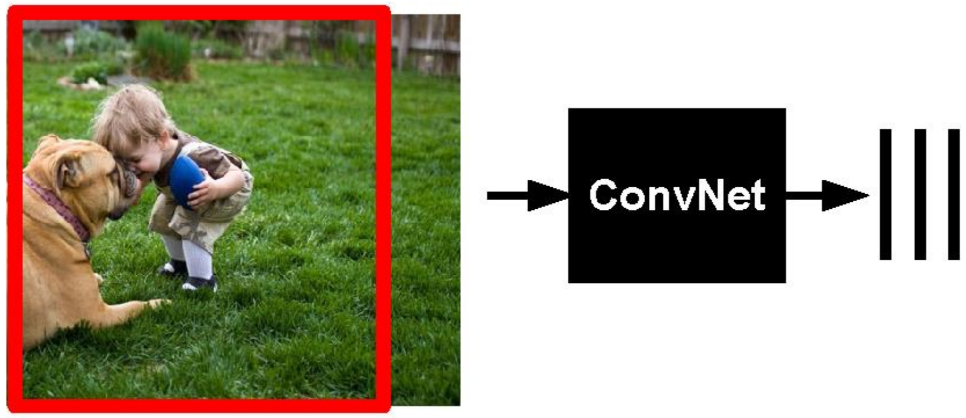
Traditional methods use inefficient sliding windows.

ConvNets: Test

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



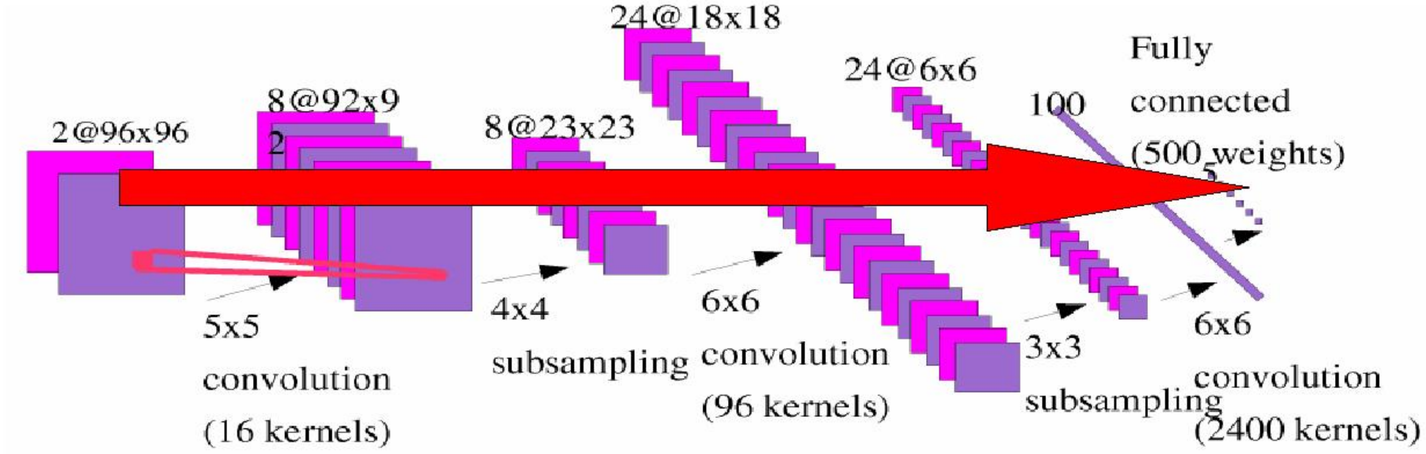
Naturally, convnet can process larger images



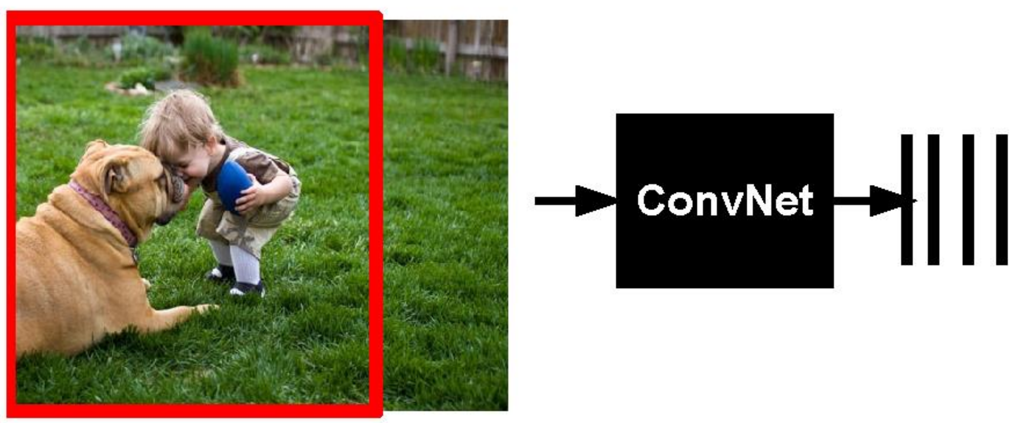
Traditional methods use inefficient sliding windows.

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: Region-based CNN

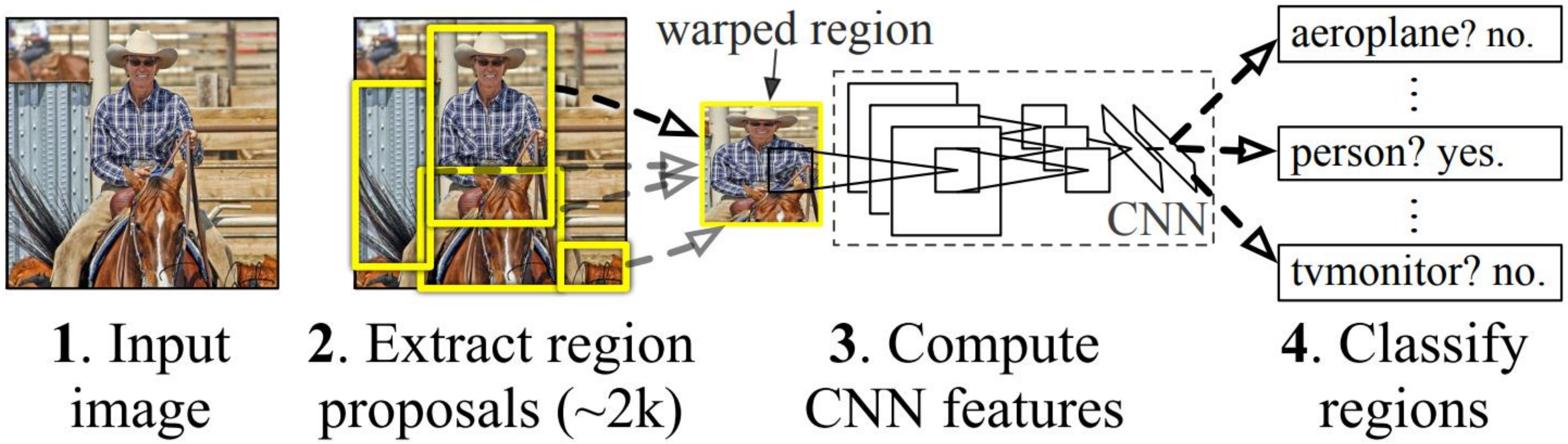


Figure: Girshick et al.

Stage 2: Efficient region proposals?

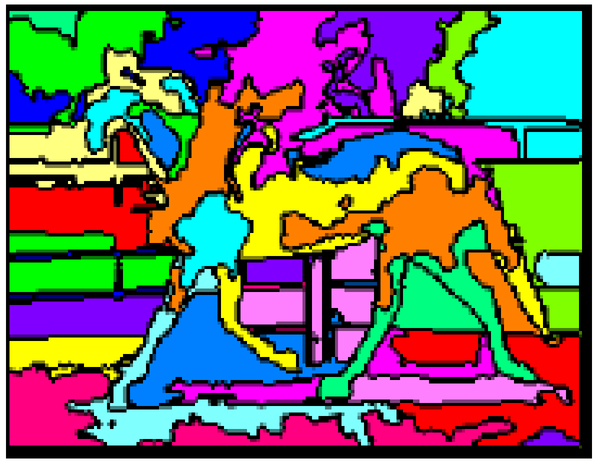
- Brute force on 1000x1000 = 250 billion rectangles
  - Testing the CNN over each one is too expensive

- Let's use B.C. vision! Before CNNs
  - Hierarchical clustering for segmentation

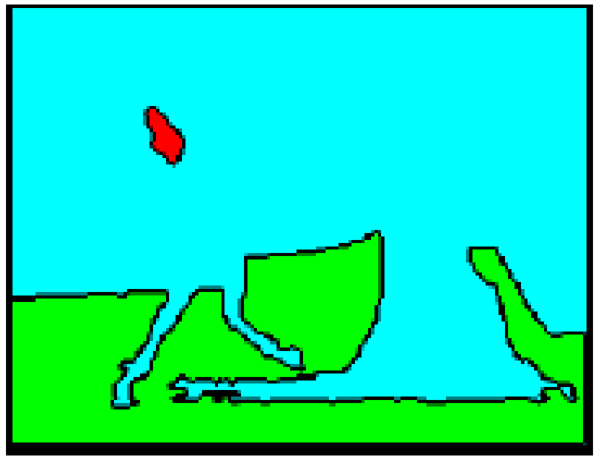
Uijlings et al., 2012, Selection Search

Thanks to Song Cao

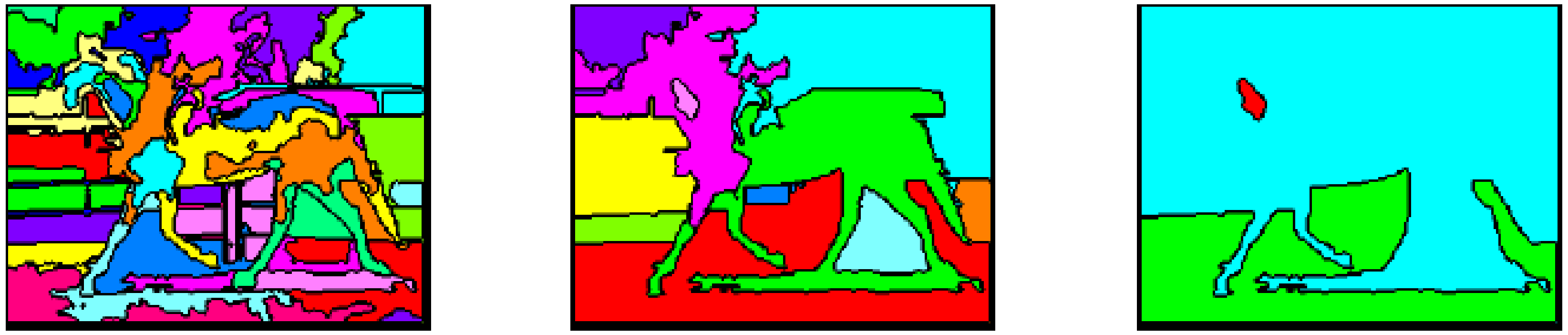
Remember clustering for segmentation?



Oversegmentation



Undersegmentation



Hierarchical Segmentations

Cluster low-level features

- Define similarity on color, texture, size, ‘fill’
- Greedily group regions together by selecting the pair with highest similarity
  - Until the whole image become a single region
- Draw a bounding box around each one
  - Into a hierarchy

Vs Ground Truth

Average Best Overlap (ABO)

$$\text{ABO} = \frac{1}{|G^c|} \sum_{g_i^c \in G^c} \max_{l_j \in L} \text{Overlap}(g_i^c, l_j).$$

$$\text{Overlap}(g_i^c, l_j) = \frac{\text{area}(g_i^c) \cap \text{area}(\mathrm{l_j})}{\text{area}(g_i^c) \cup \text{area}(\mathrm{l_j})}.$$

(a) Bike: 0.863 (b) Cow: 0.874 (c) Chair: 0.884 (d) Person: 0.882 (e) Plant: 0.873

Mean Average Best Overlap (MABO)

Thanks to Song Cao

method	recall	MABO	# windows
Arbelaez *et al*. [3]	0.752	0.649 ± 0.193	418
Alexe *et al*. [2]	0.944	0.694 ± 0.111	1,853
Harzallah *et al*. [16]	0.830	-	200 per class
Carreira and Sminchisescu [4]	0.879	0.770 ± 0.084	517
Endres and Hoiem [9]	0.912	0.791 ± 0.082	790
Felzenszwalb *et al*. [12]	0.933	0.829 ± 0.052	100,352 per class
Vedaldi *et al*. [34]	0.940	-	10,000 per class
Single Strategy	0.840	0.690 ± 0.171	289
Selective search "Fast"	0.980	0.804 ± 0.046	2,134
Selective search "Quality"	0.991	0.879 ± 0.039	10,097

Table 5: Comparison of recall, Mean Average Best Overlap (MABO) and number of window locations for a variety of methods on the Pascal 2007 TEST set.

Thanks to Song Cao

R-CNN: Region-based CNN

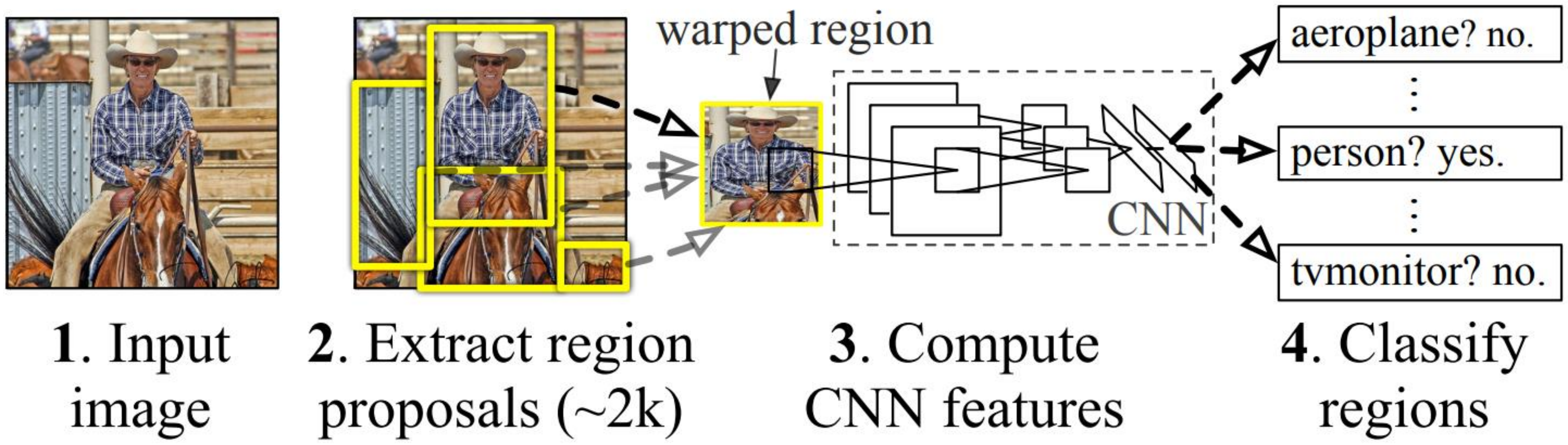
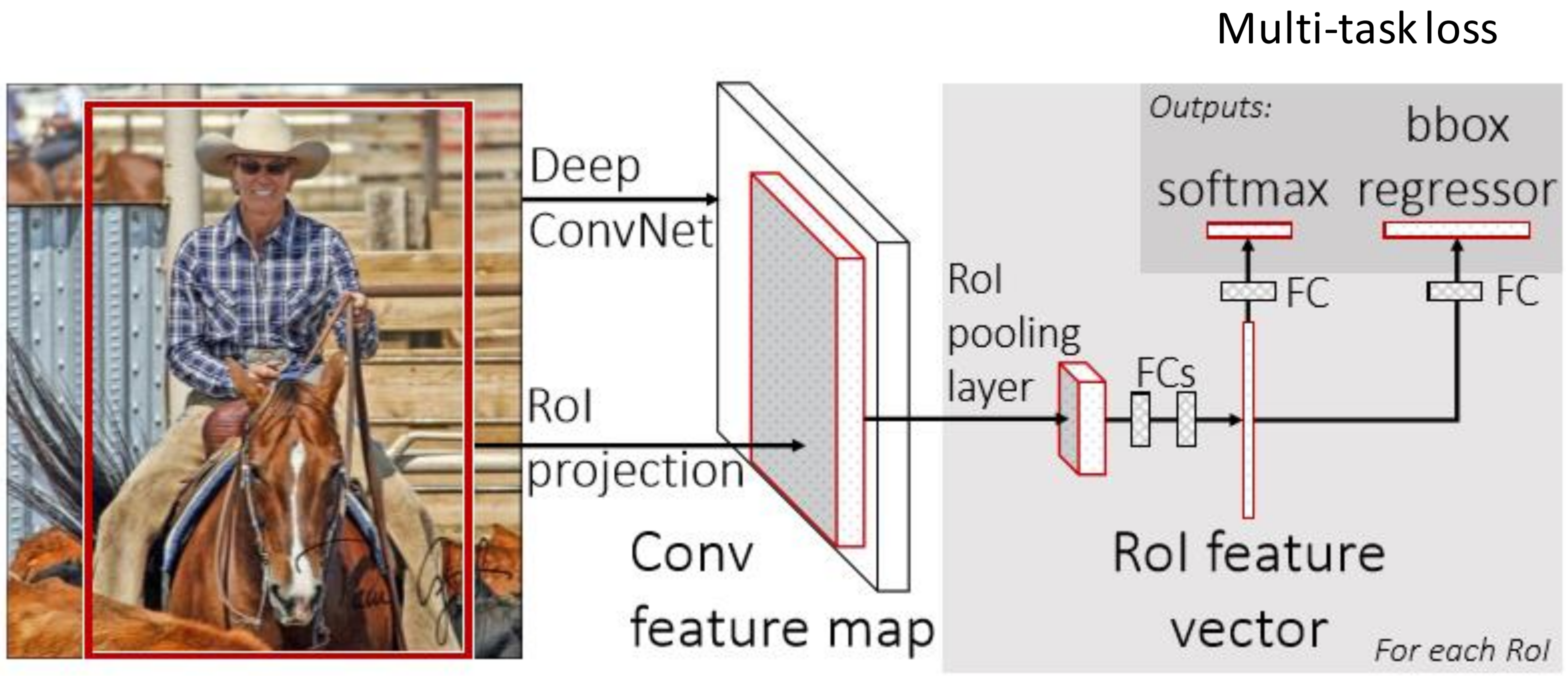


Figure: Girshick et al.

10,000 proposals with recall 0.991 is better….
but still takes 17 seconds per image to generate them.
Then I have to test each one!

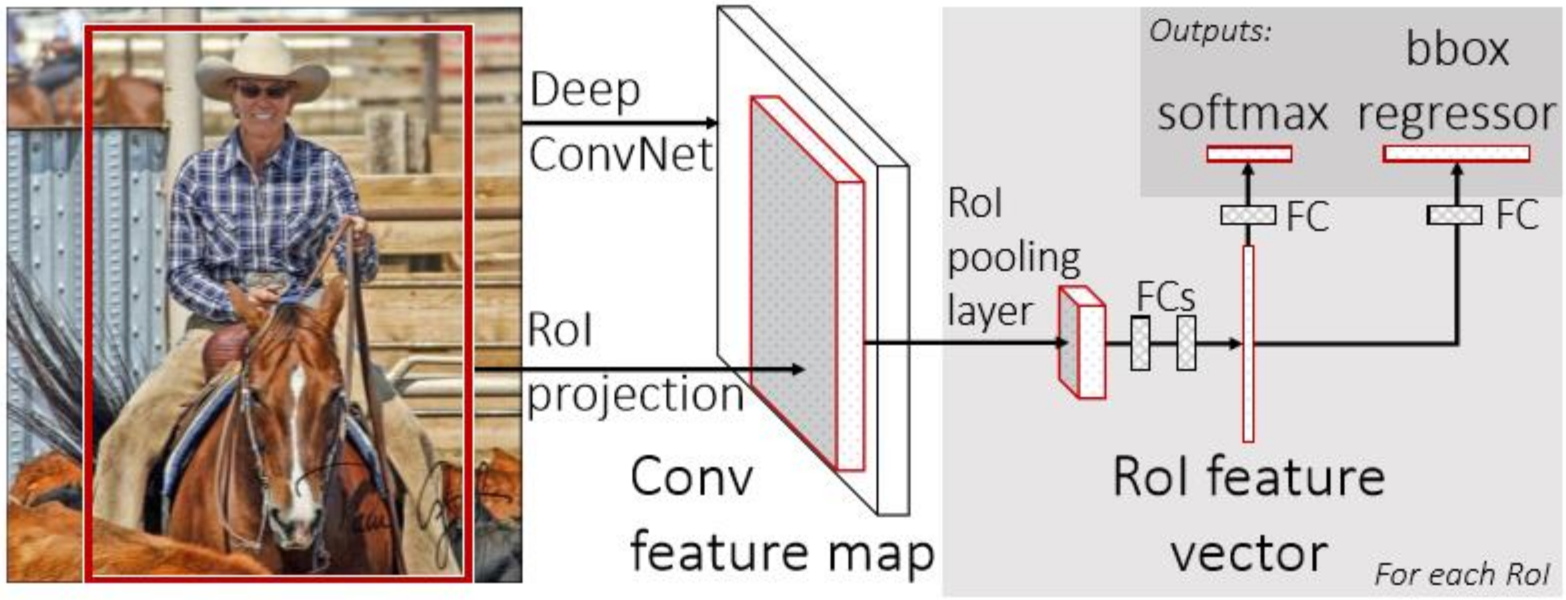
Fast R-CNN



RoI = Region of Interest

Figure: Girshick et al.

Fast R-CNN

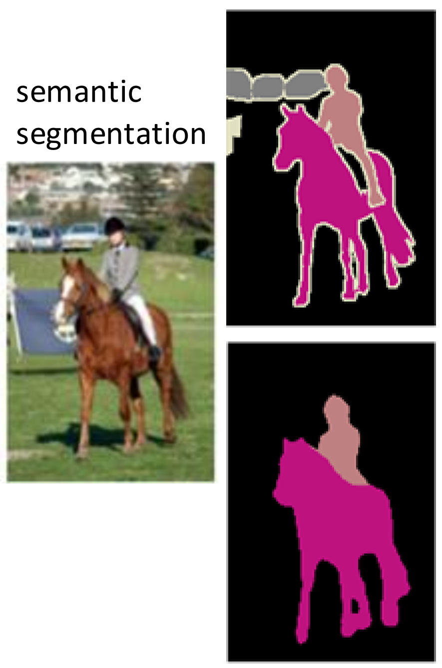


- Convolve whole image into feature map (many layers; abstracted)
- For each candidate RoI:
  - Squash feature map weights into fixed-size 'RoI pool' – adaptive subsampling!
    - Divide RoI into H x W subwindows, e.g., 7 x 7, and max pool
  - Learn classification on RoI pool with own fully connected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

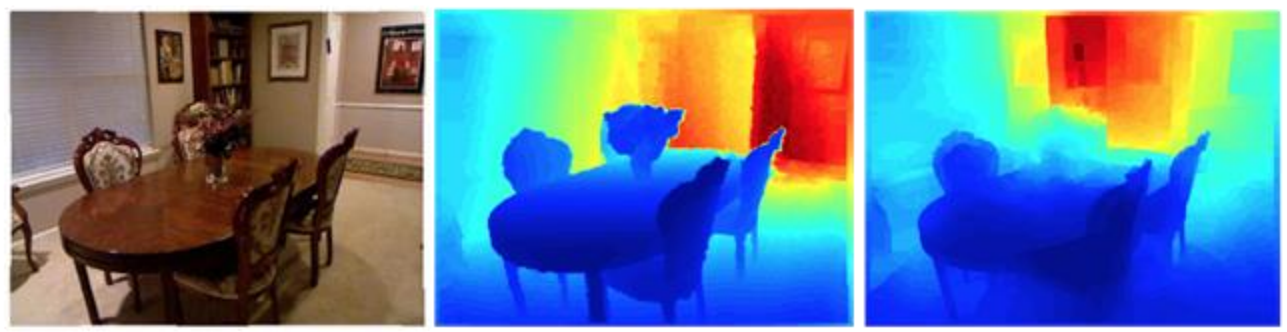
Figure: Girshick et al.

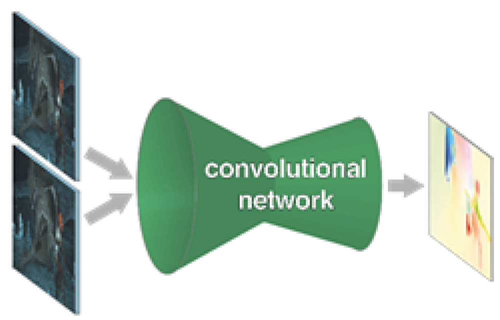
What if we want pixels out?

semantic
segmentation



monocular depth estimation Eigen & Fergus 2015





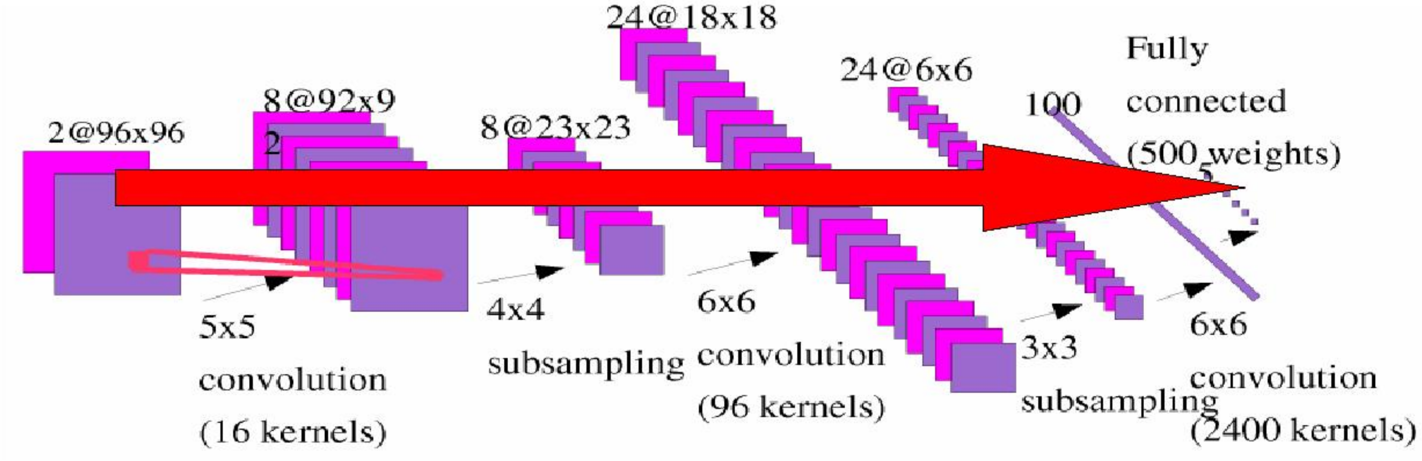
optical flow Fischer et al. 2015

boundary prediction Xie & Tu 2015

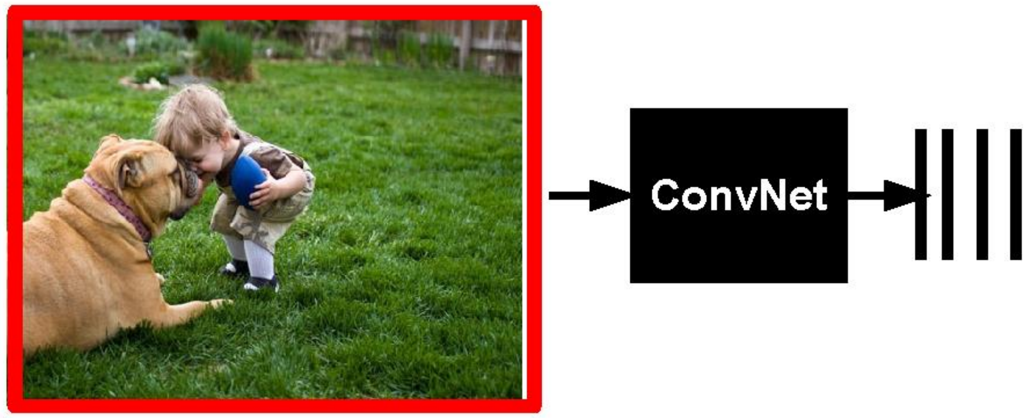
[Long et al.]

ConvNets: Test

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

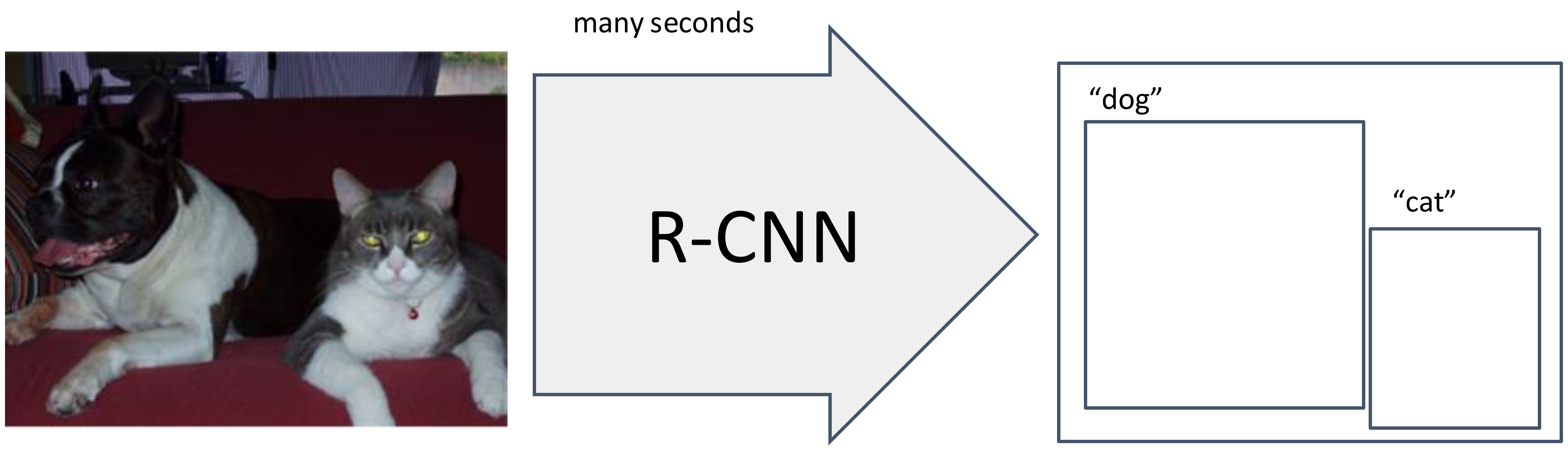


Naturally, convnet can process larger images at little cost.

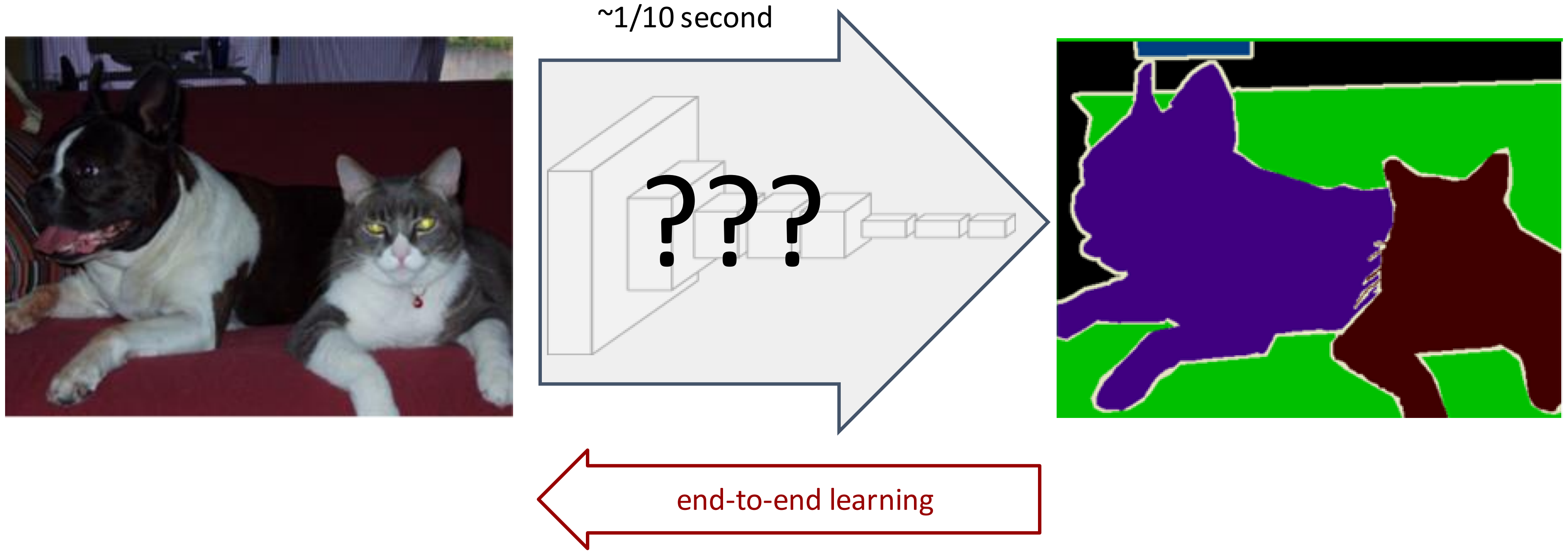


ConvNet: unrolls convolutions over bigger images and produces outputs at several locations.

R-CNN does detection

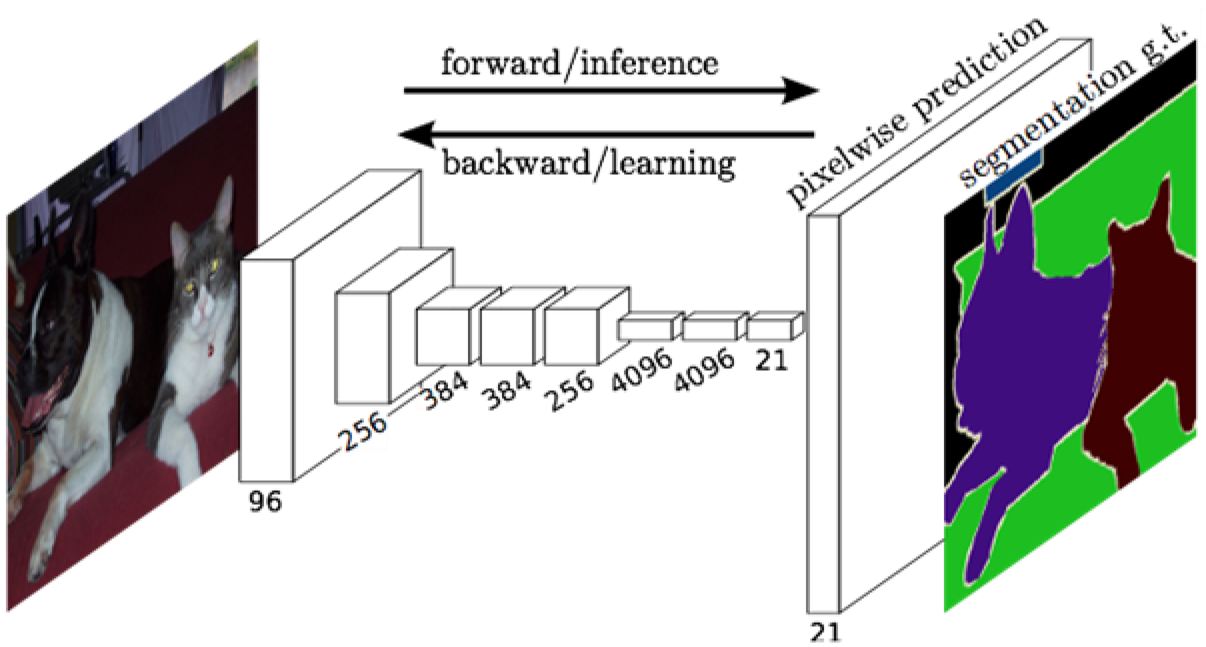


[Long et al.]



[Long et al.]

Fully Convolutional Networks for Semantic Segmentation

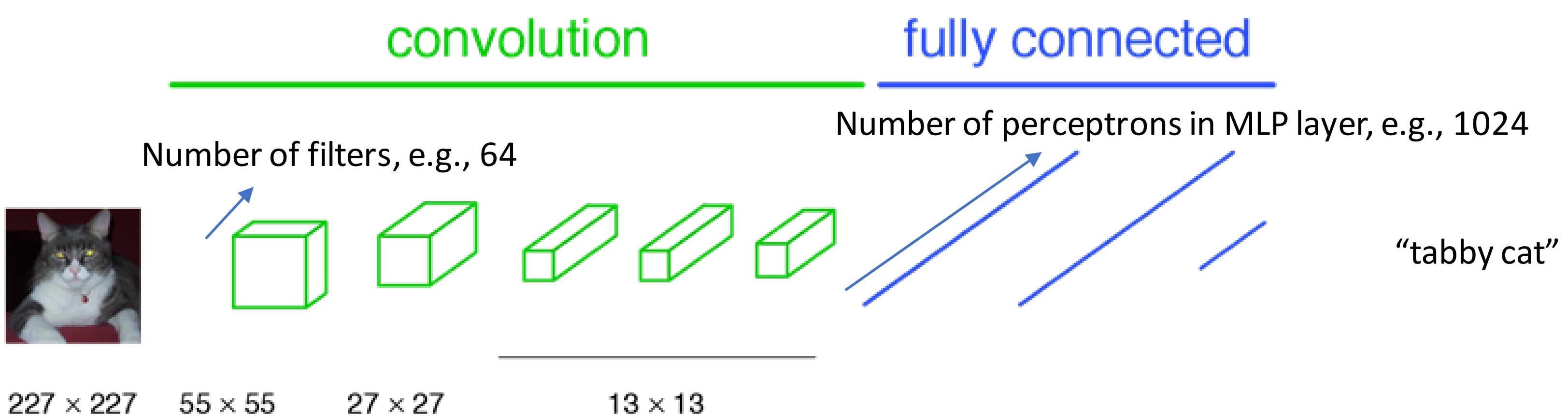


Jonathan Long* Evan Shelhamer* Trevor Darrell
UC Berkeley

[CVPR 2015]

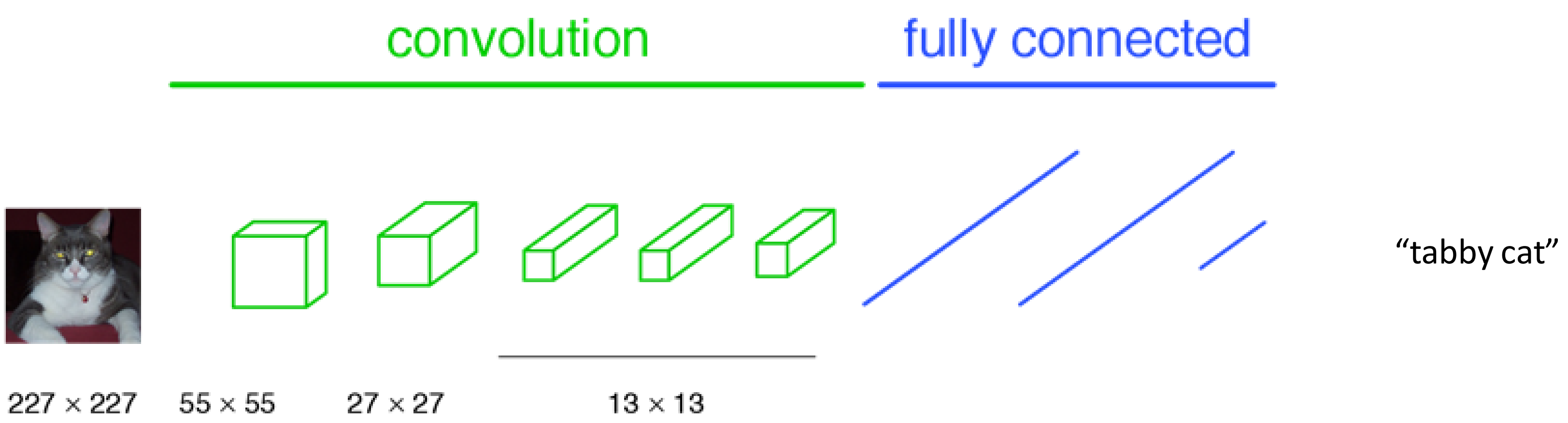
Slides from Long, Shelhamer, and Darrell

A classification network…



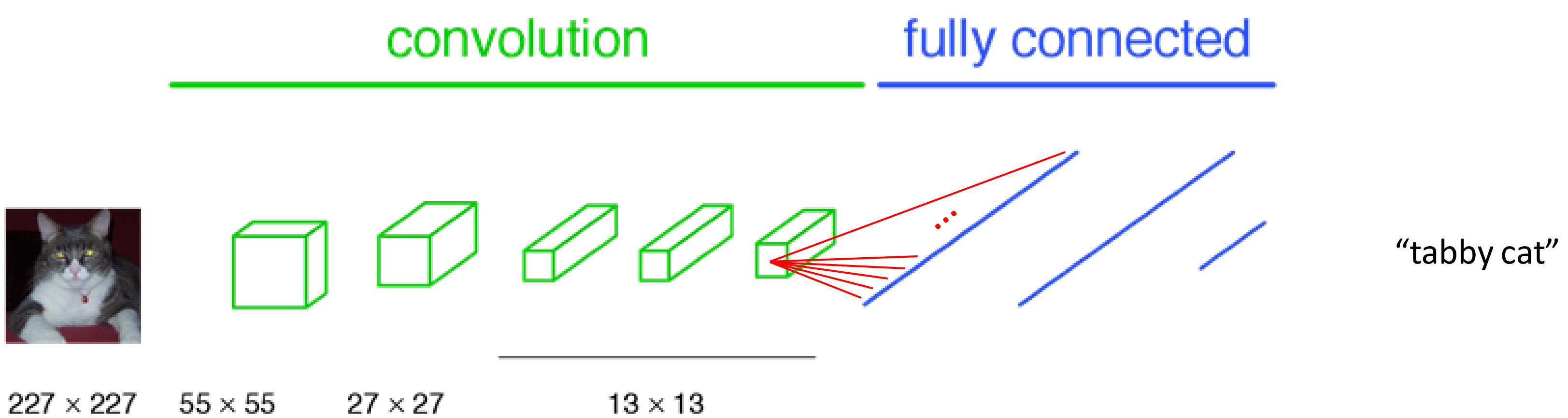
[Long et al.]

A classification network…



[Long et al.]

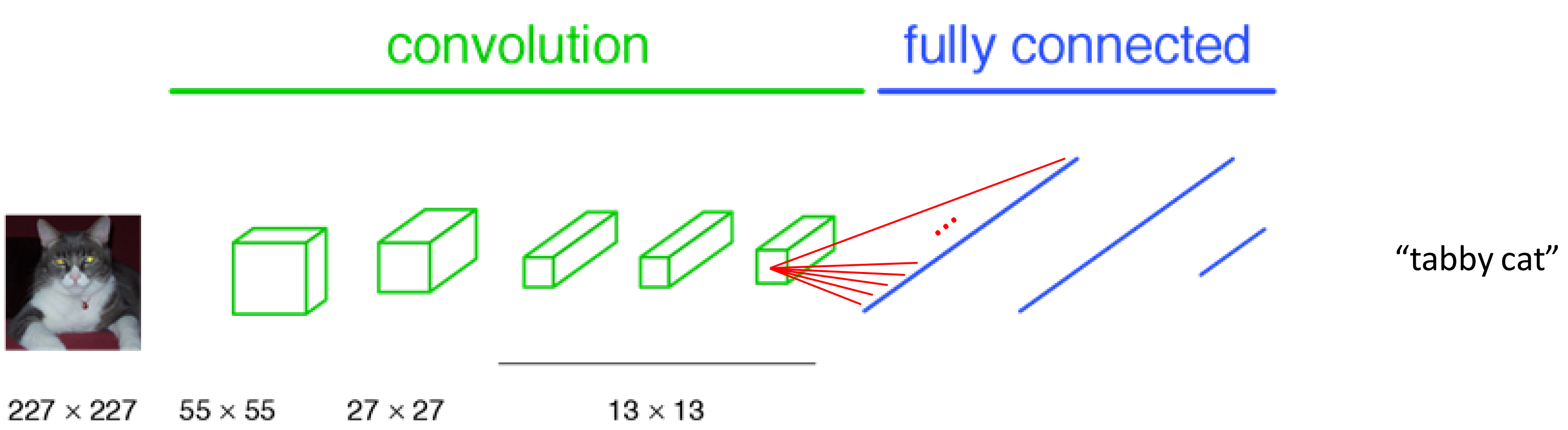
A classification network…



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.

Approach:

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

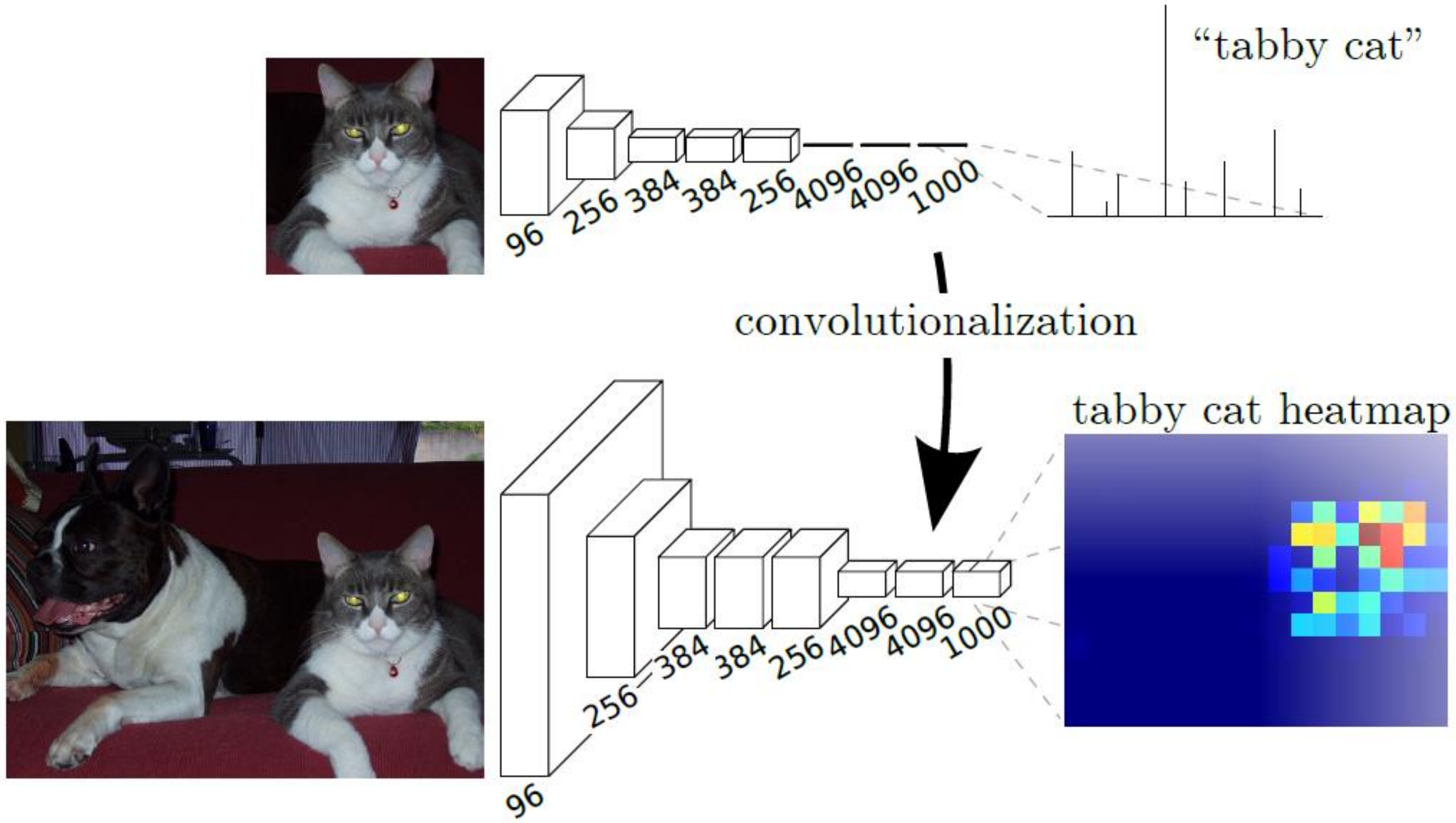
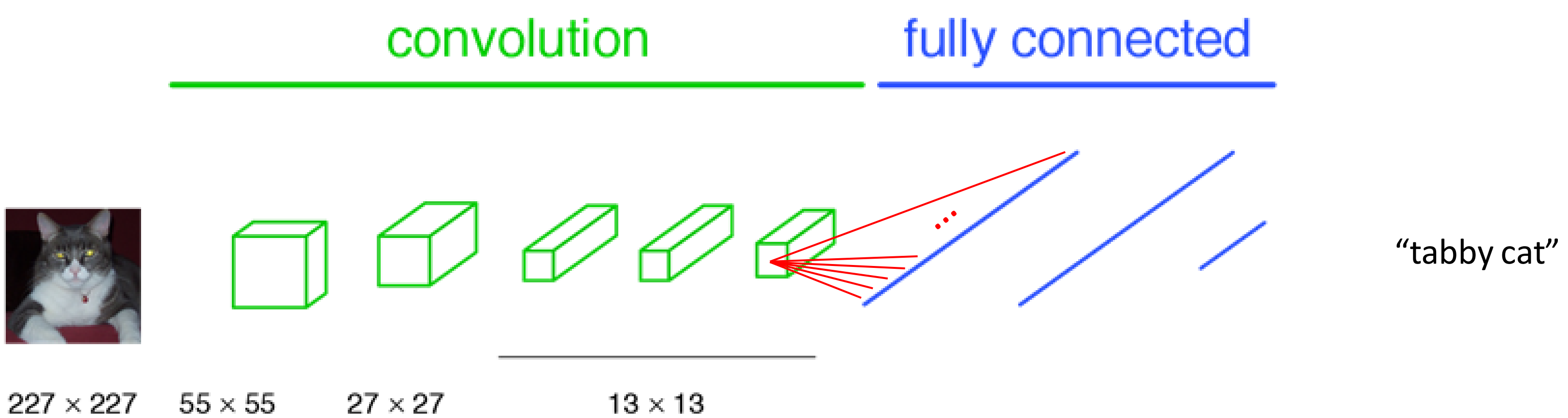


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

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.

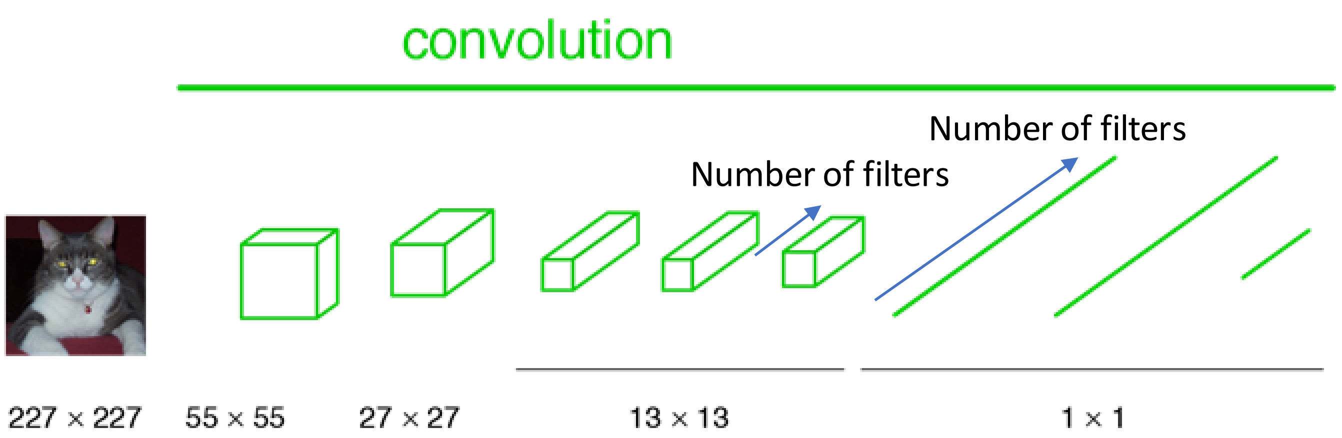
[Long et al.]

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



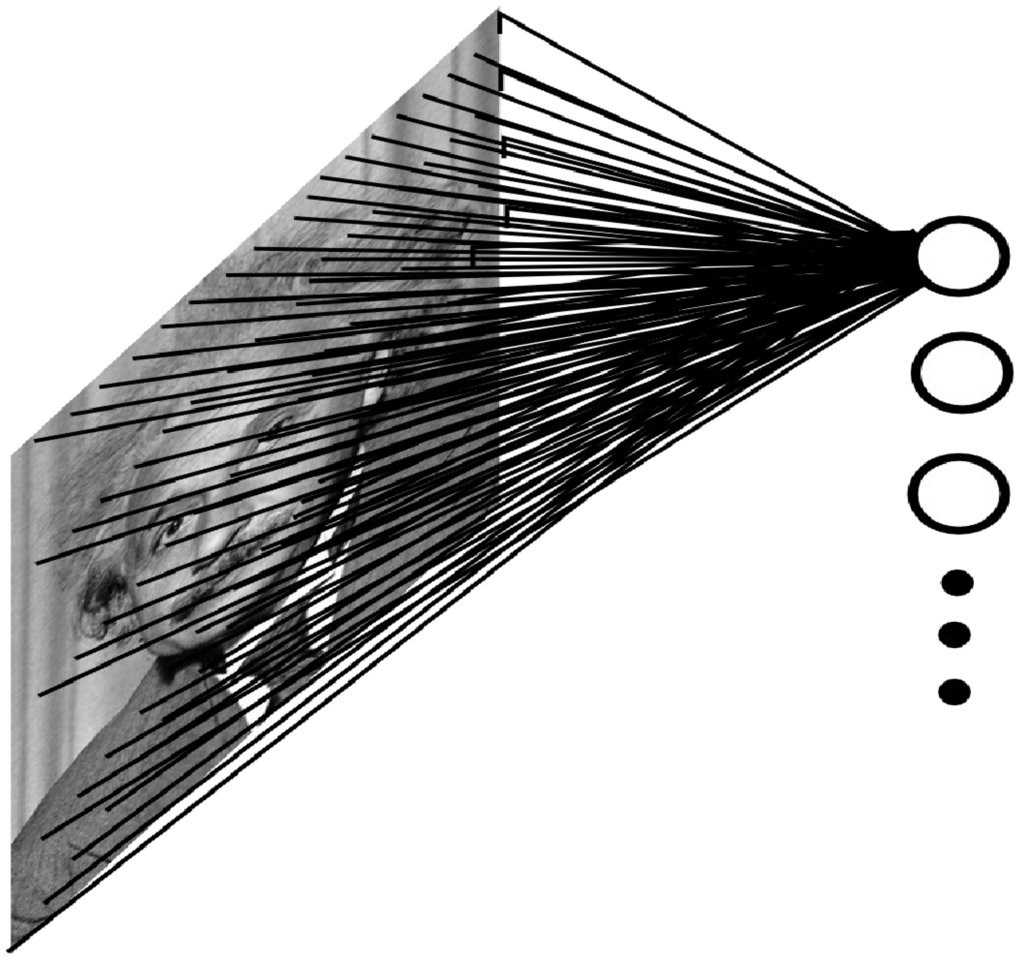
1x1 convolution operates across all filters in the previous layer, and is slid across all positions.

[Long et al.]

Back to the fully-connected perceptron…

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

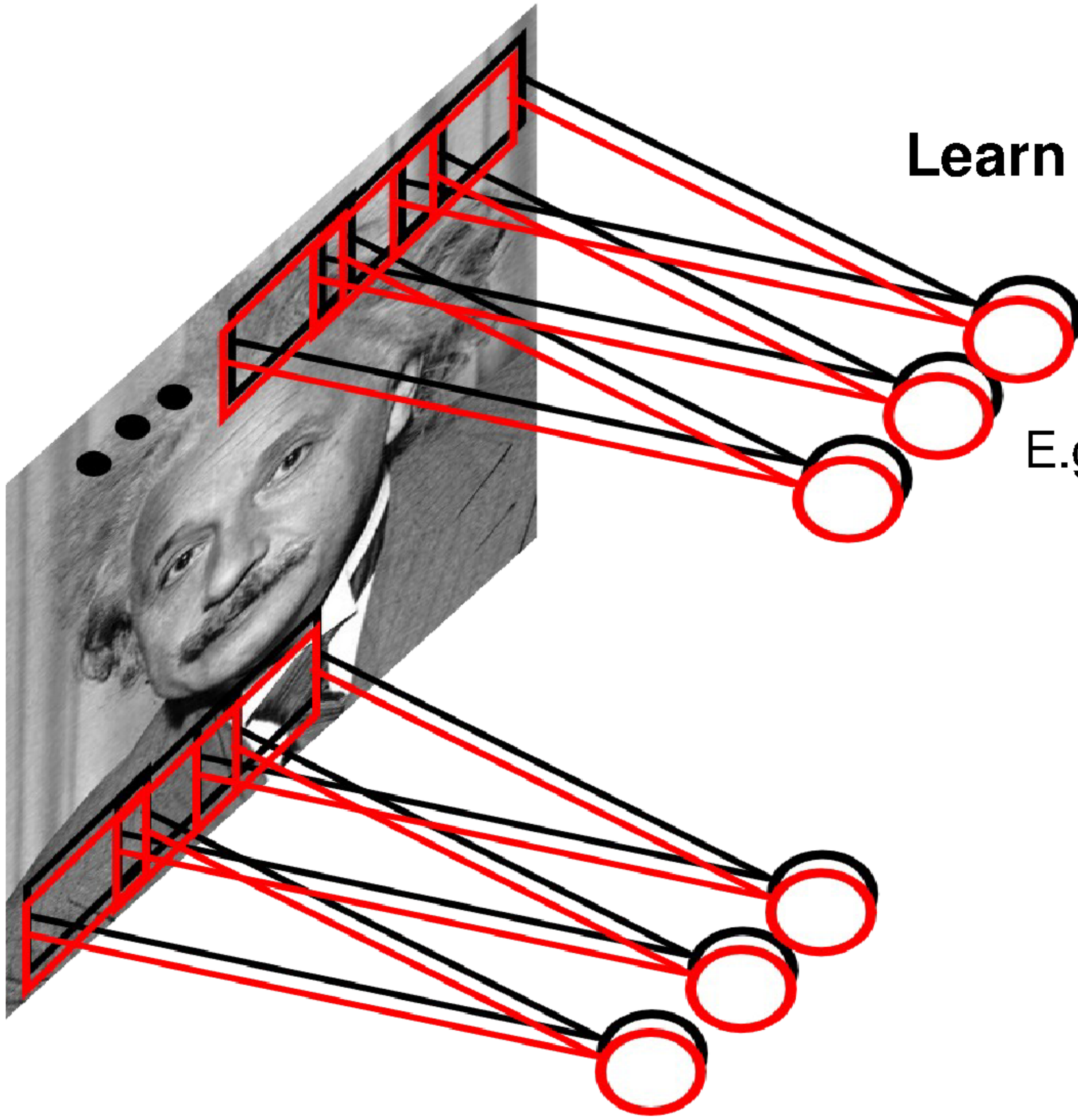
$$w \cdot x \equiv \sum_j w_j x_j,$$



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

[Long et al.]

Convolutional Layer

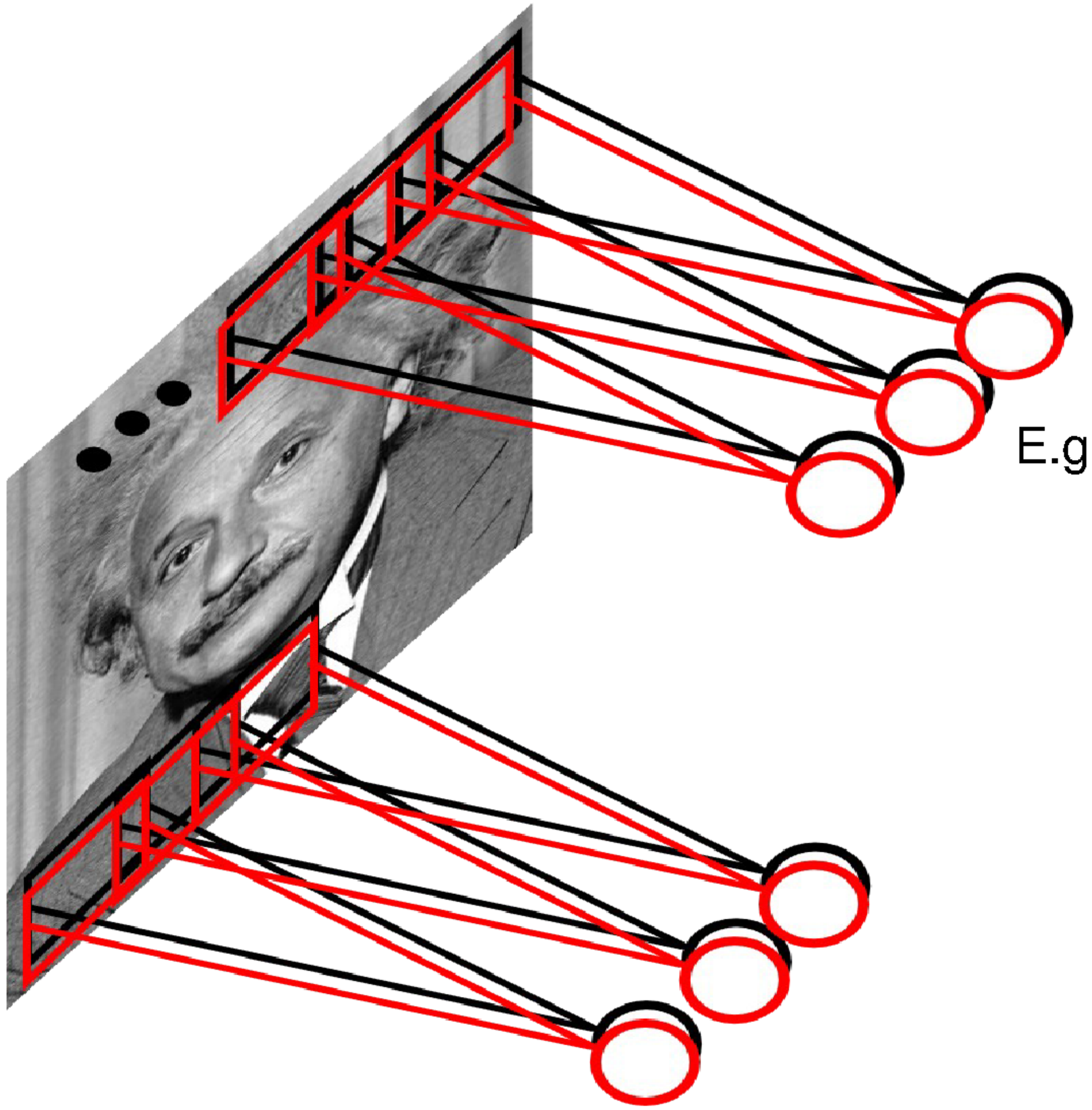


Learn multiple filters.

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

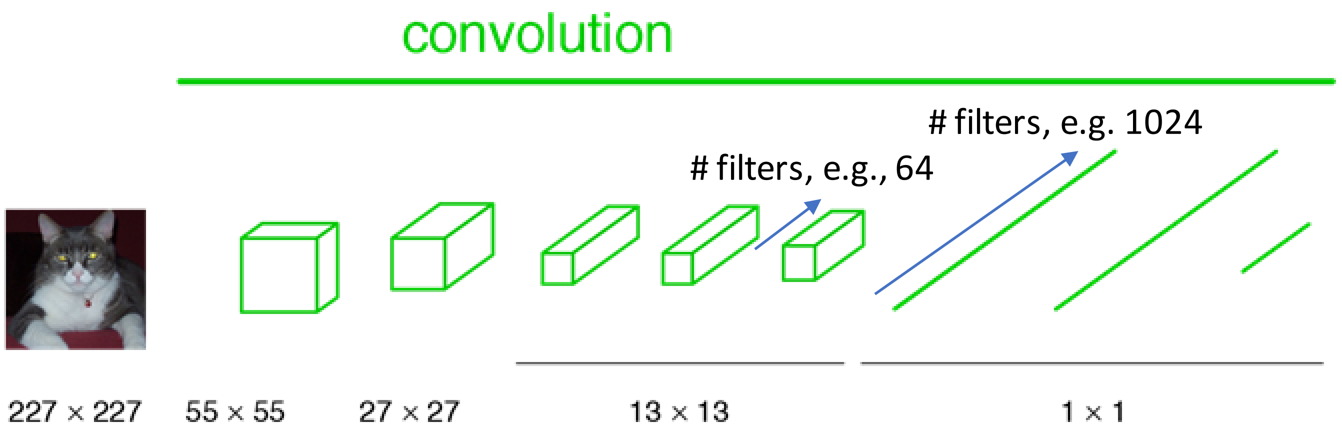
Ranzato

Convolutional Layer



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

Convolutionalization

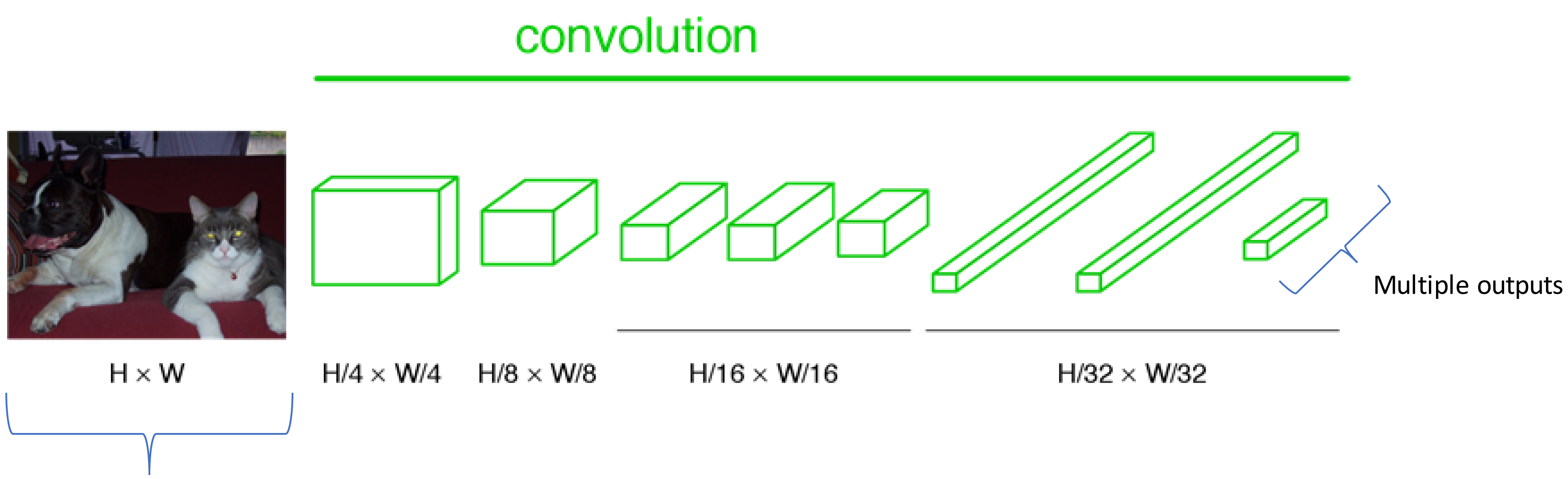


1x1 convolution operates across all filters in the previous layer, and is slid across all positions.

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.

46

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

[Long et al.]

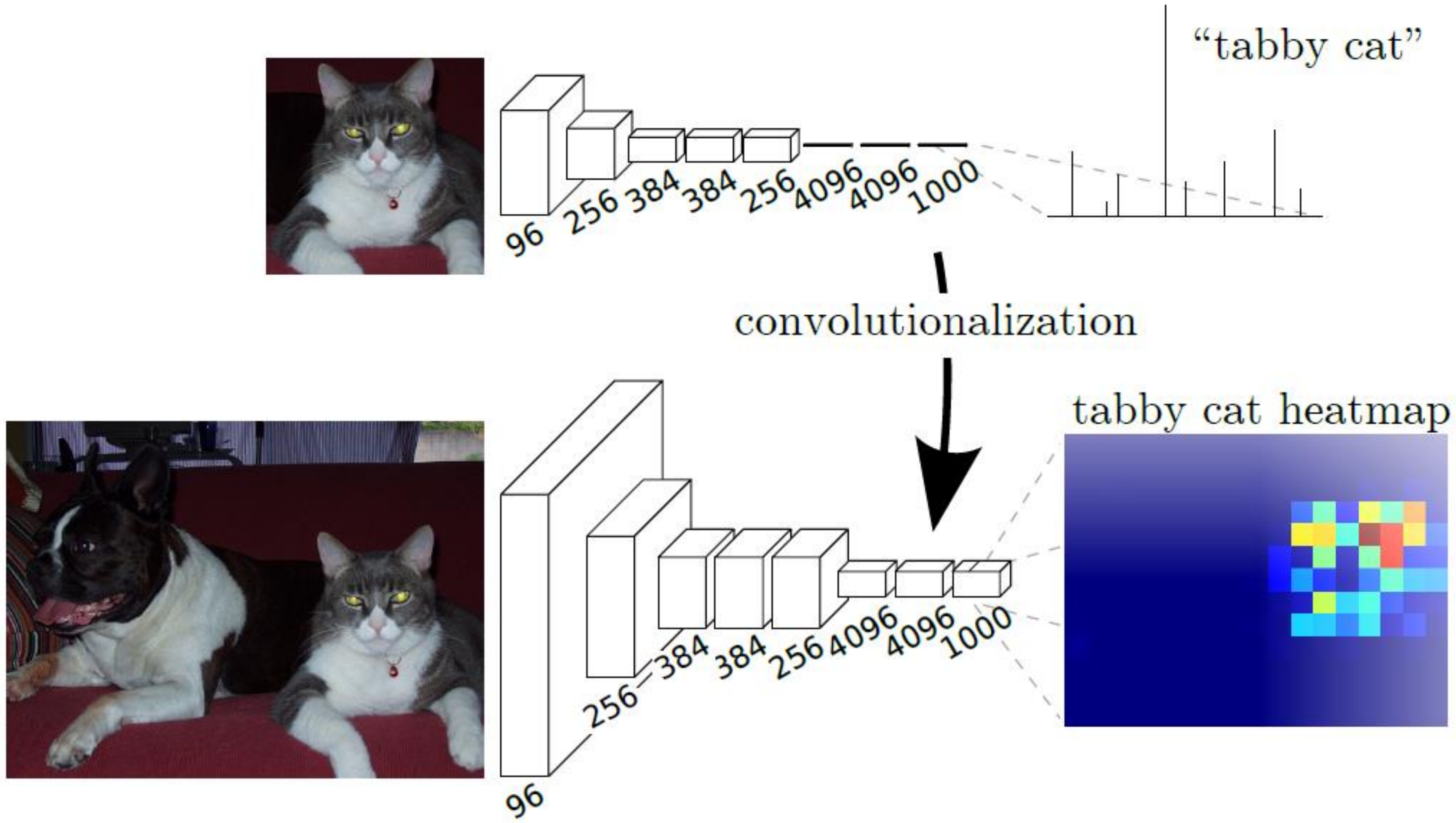
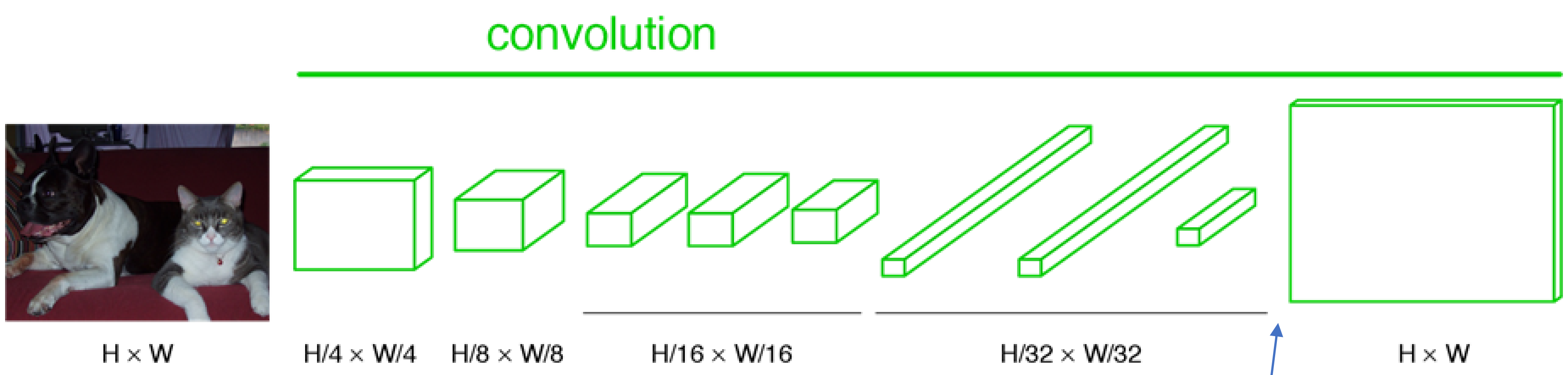


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

Long, Shelhamer, and Darrell 2014

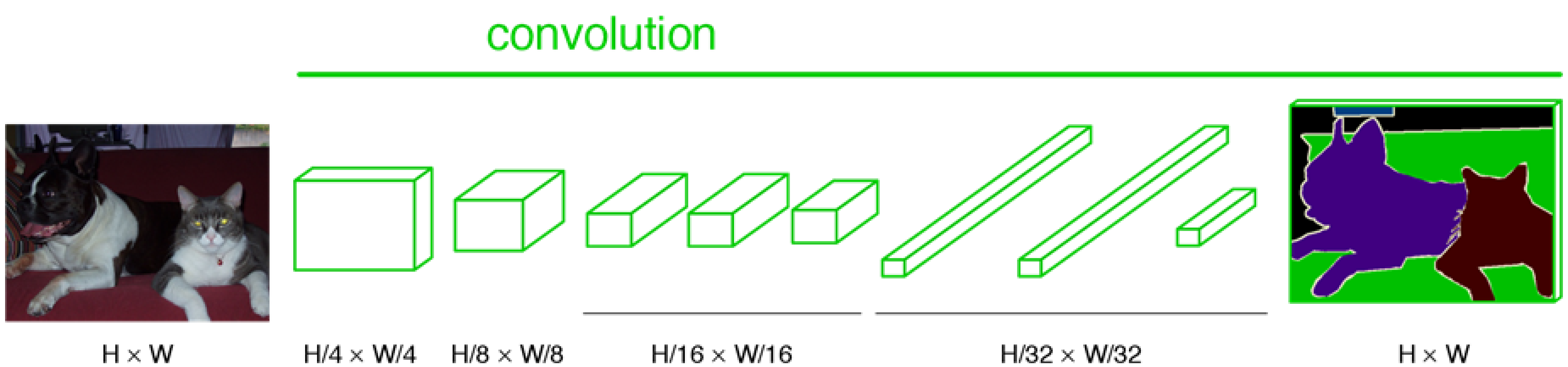
Upsampling the output



Some upsampling
algorithm to
return us to H x W

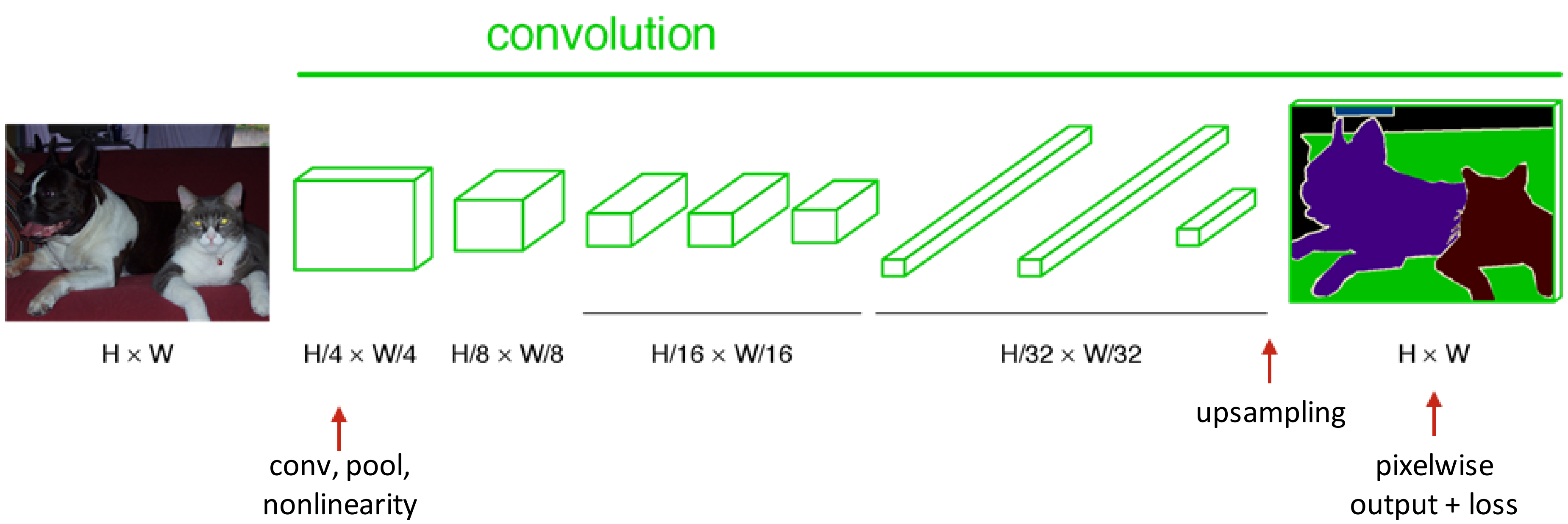
[Long et al.]

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



[Long et al.]

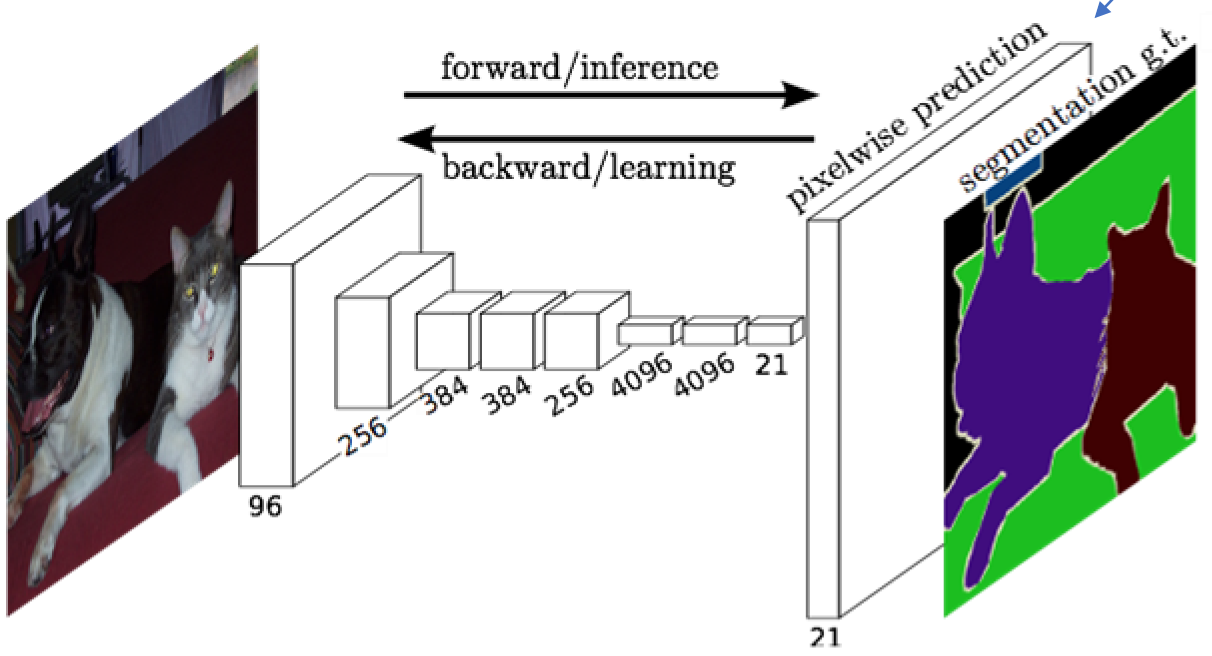
End-to-end, pixels-to-pixels network



[Long et al.]

What is the upsampling layer?

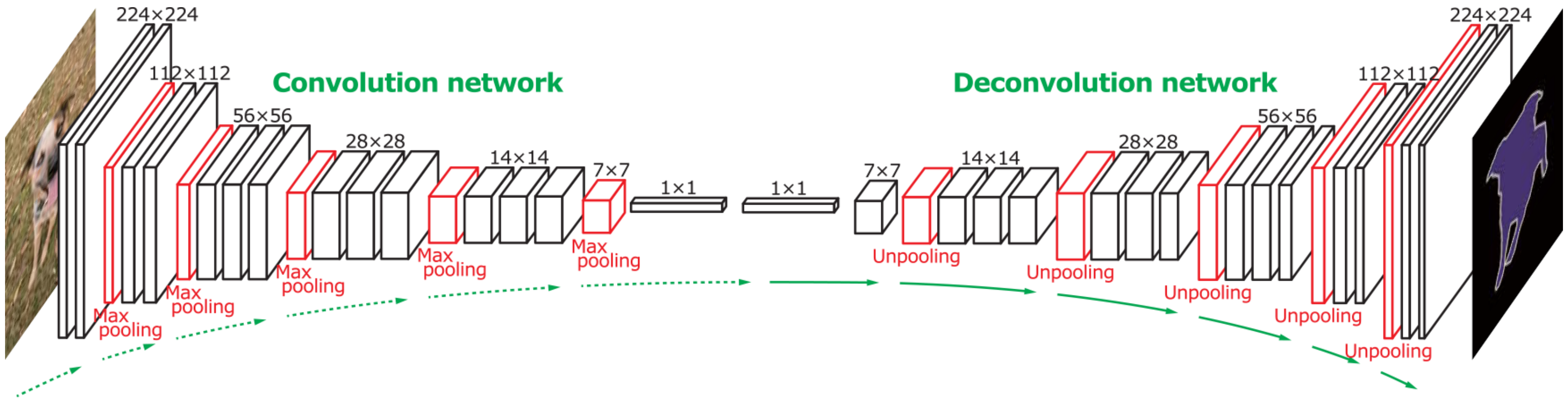
This one.



Hint: it’s actually an upsampling *network*

[Long et al.]

‘Deconvolution’ networks *learn to upsample*



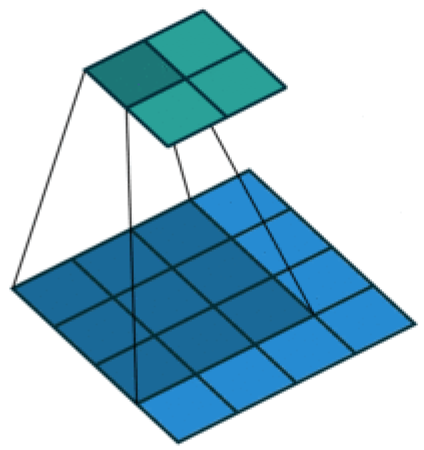
Often called “deconvolution”, but misnomer.
Not the deconvolution that we saw in deblurring -> that is division in the Fourier domain.

‘Transposed convolution’ is better.

Zeiler et al., Deconvolutional Networks, CVPR 2010
Noh et al., Learning Deconvolution Network for Semantic Segmentation, ICCV 2015

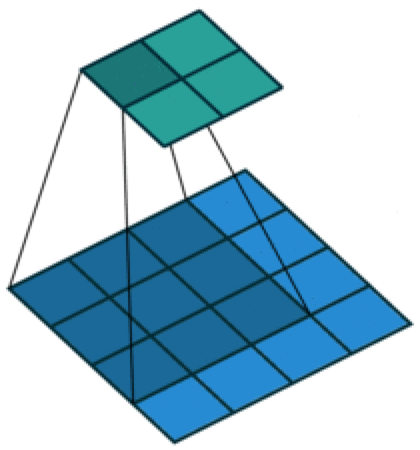
Upsampling with transposed convolution

Convolution

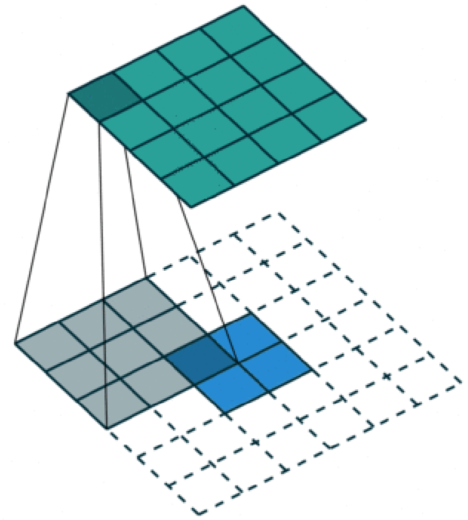


Upsampling with transposed convolution

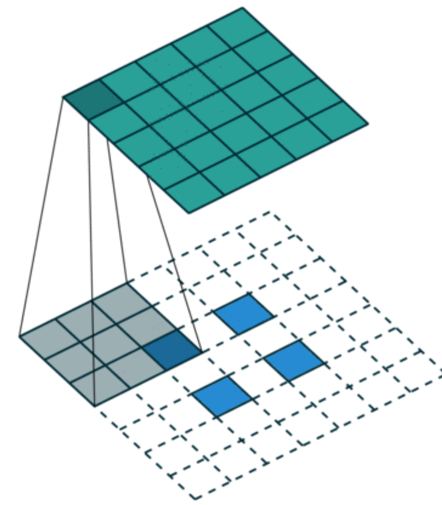
Convolution



Transposed convolution = padding/striding smaller image then weighted sum of input x filter: 'stamping' kernel



2x2, stride 1, 3x3 kernel, upsample to 4x4



2x2, stride 2, 3x3 kernel, upsample to 5x5.

Kernel

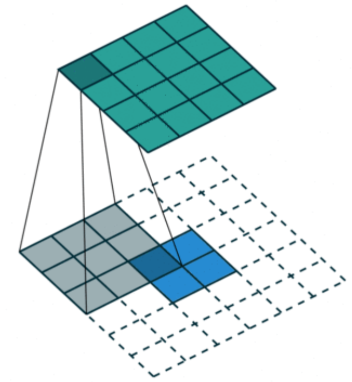
1	1	1
1	1	1
1	1	1

Feature map

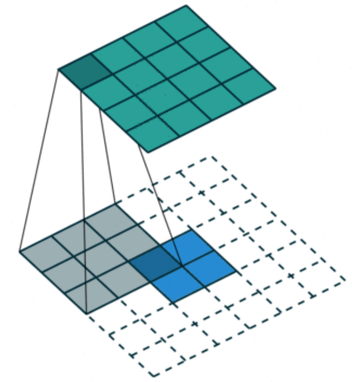
1	2
3	4

Padded feature map

		1	2		
		3	4		



Inspired by andriys



Kernel

1	1	1
1	1	1
1	1	1

Input feature map

1	2
3	4

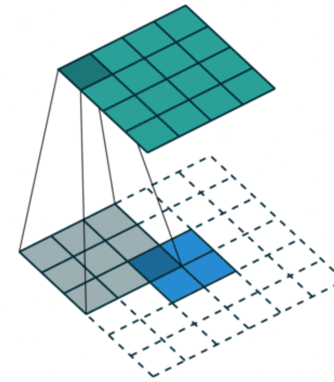
Padded input feature map

		1	2		
		3	4		

Output feature map

1	1	1			
1	1	1			
1	1	1			

Inspired by andriys



Kernel

1	1	1
1	1	1
1	1	1

Input feature map

1	2
3	4

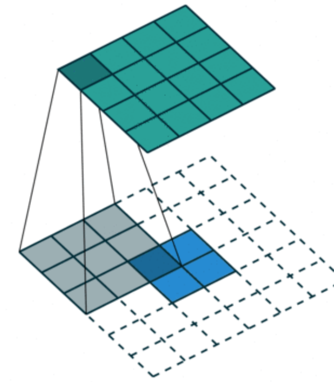
Padded input feature map

		1	2		
		3	4		

Output feature map

1	4	4	3		
1	4	4	3		
1	4	4	3		

Inspired by andriys



Kernel

1	1	1
1	1	1
1	1	1

Input feature map

1	2
3	4

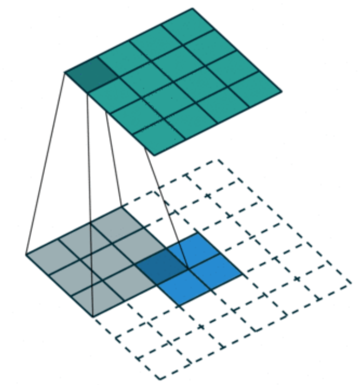
Padded input feature map

		1	2		
		3	4		

Output feature map

1	4	7	6	3	
1	4	7	6	3	
1	4	7	6	3	

Inspired by andriys



Kernel

1	1	1
1	1	1
1	1	1

Input feature map

1	2
3	4

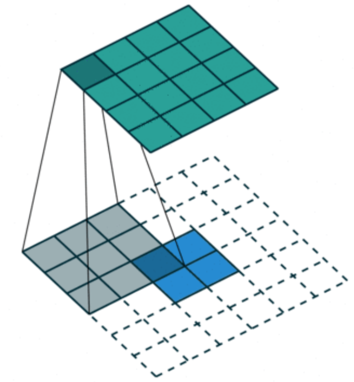
Padded input feature map

		1	2		
		3	4		

Output feature map

1	4	7	8	5	2
1	4	7	8	5	2
1	4	7	8	5	2

Inspired by andriys



Kernel

1	1	1
1	1	1
1	1	1

Input feature map

1	2
3	4

Padded input feature map

		1	2		
		3	4		

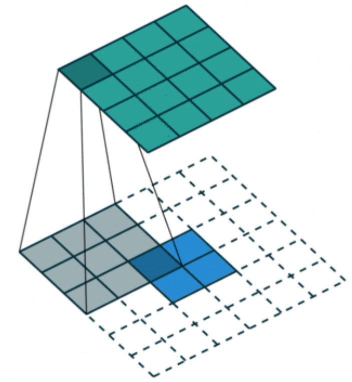
Output feature map

1	4	7	8	5	2
5	8	11	8	5	2
5	8	11	8	5	2
4	4	4			

Inspired by andriys

Kernel

1	1	1
1	1	1
1	1	1



Input feature map

1	2
3	4

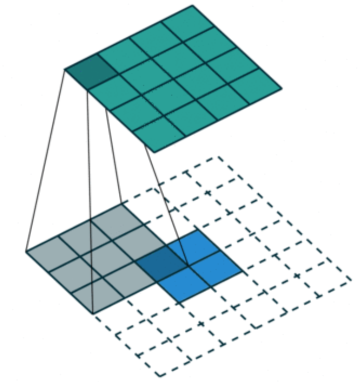
Padded input feature map

		1	2		
		3	4		

Output feature map

1	4	7	8	5	2
5	18	21	18	5	2
5	18	21	18	5	2
4	14	14	10		

Inspired by andriys



Kernel

1	1	1
1	1	1
1	1	1

Input feature map

1	2
3	4

Padded input feature map

		1	2		
		3	4		

Output feature map

1	4	7	8	5	2
5	18	31	34	21	8
9	32	55	60	37	14
11	38	66	64	43	16
7	24	41	44	27	10
3	10	17	18	11	4

Inspired by andriys

Kernel

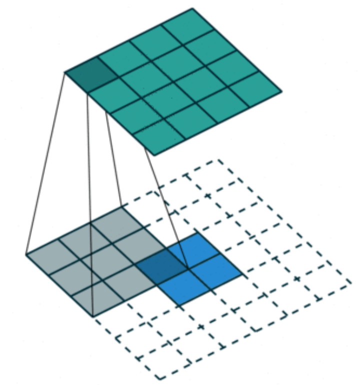
1	1	1
1	1	1
1	1	1

Input feature map

1	2
3	4

Padded input feature map

		1	2		
		3	4		

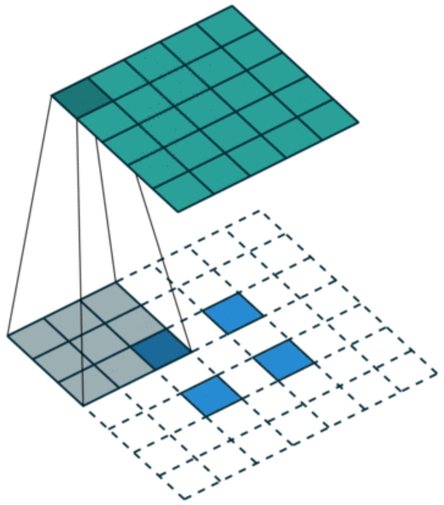


Cropped output feature map

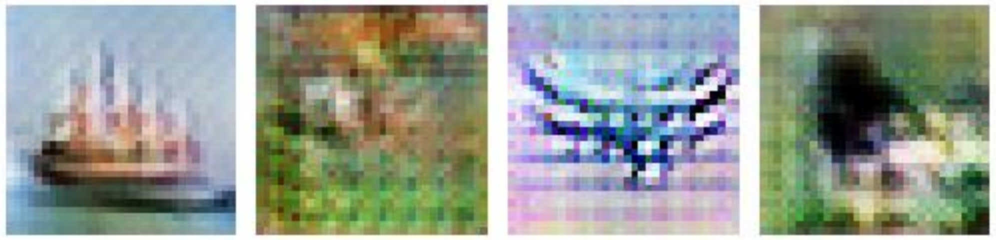
18	31	34	21
32	55	60	37
38	66	64	43
24	41	44	27

Inspired by andriys

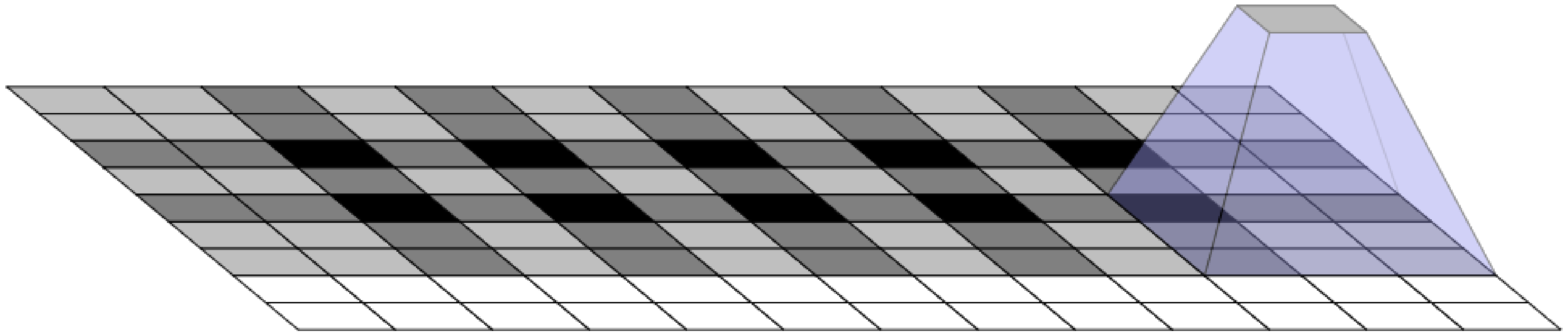
Is uneven overlap a problem?



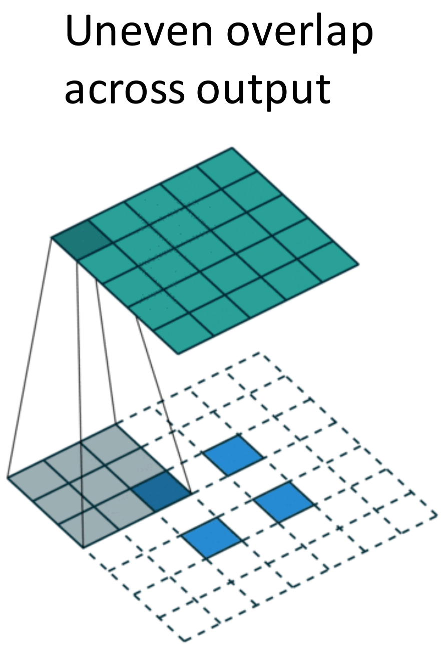
Yes = causes grid artifacts



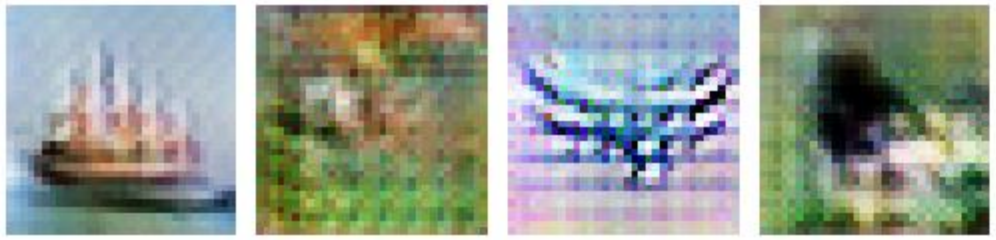
Could fix it by picking stride/kernel numbers which have no overlap…



Is uneven overlap a problem?



Yes = causes grid artifacts

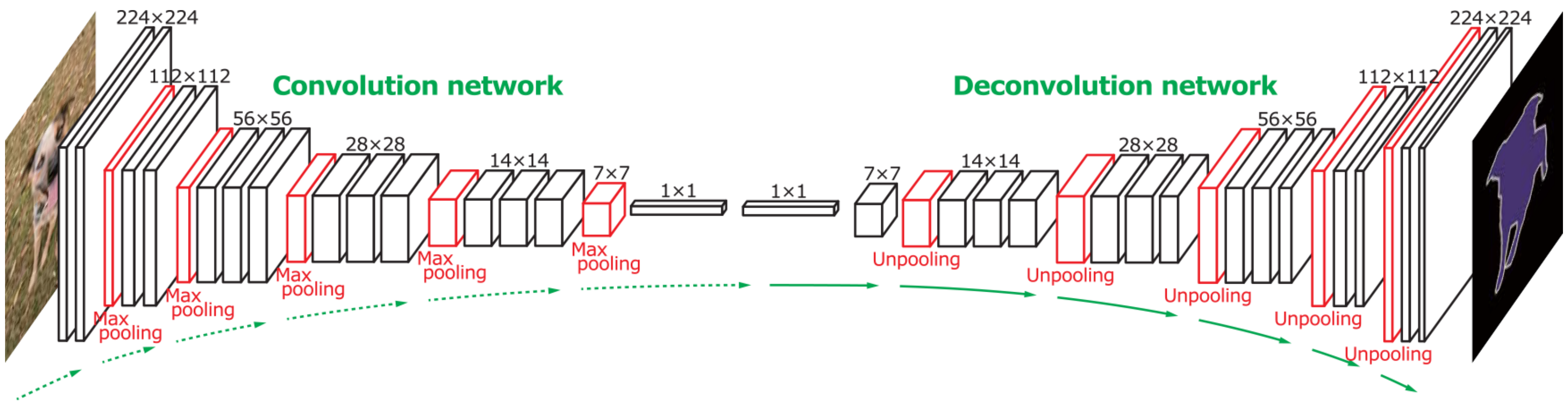


Could fix it by picking stride/kernel numbers which have no overlap…

Or…*think in frequency*!
Introduce explicit bilinear upsampling before transpose convolution; let kernels of transpose convolution learn to fill in only high-frequency detail.

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

‘Deconvolution’ networks *learn to upsample*



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

‘Transposed convolution’ is better.

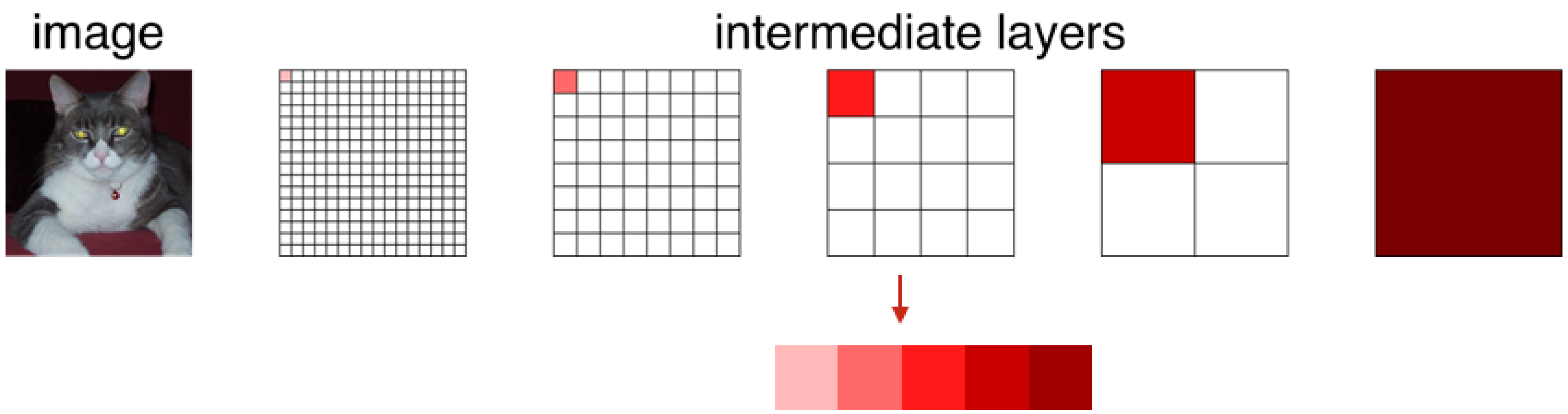
Zeiler et al., Deconvolutional Networks, CVPR 2010
Noh et al., Learning Deconvolution Network for Semantic Segmentation, ICCV 2015

But we have downsampled so far…

How do we ‘learn to create’ or ‘learn to restore’ new high frequency detail?

Spectrum of deep features

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

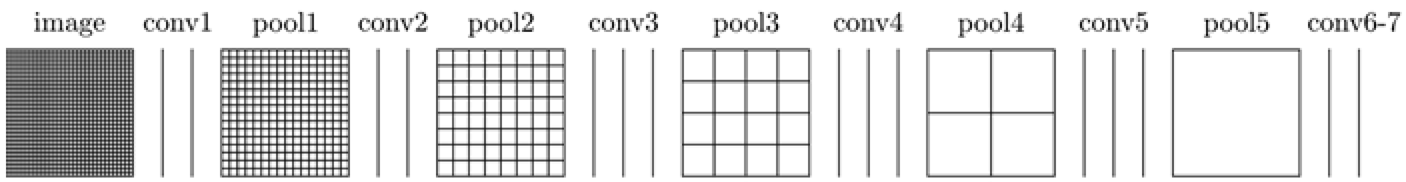


Fuse features into **deep jet**

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

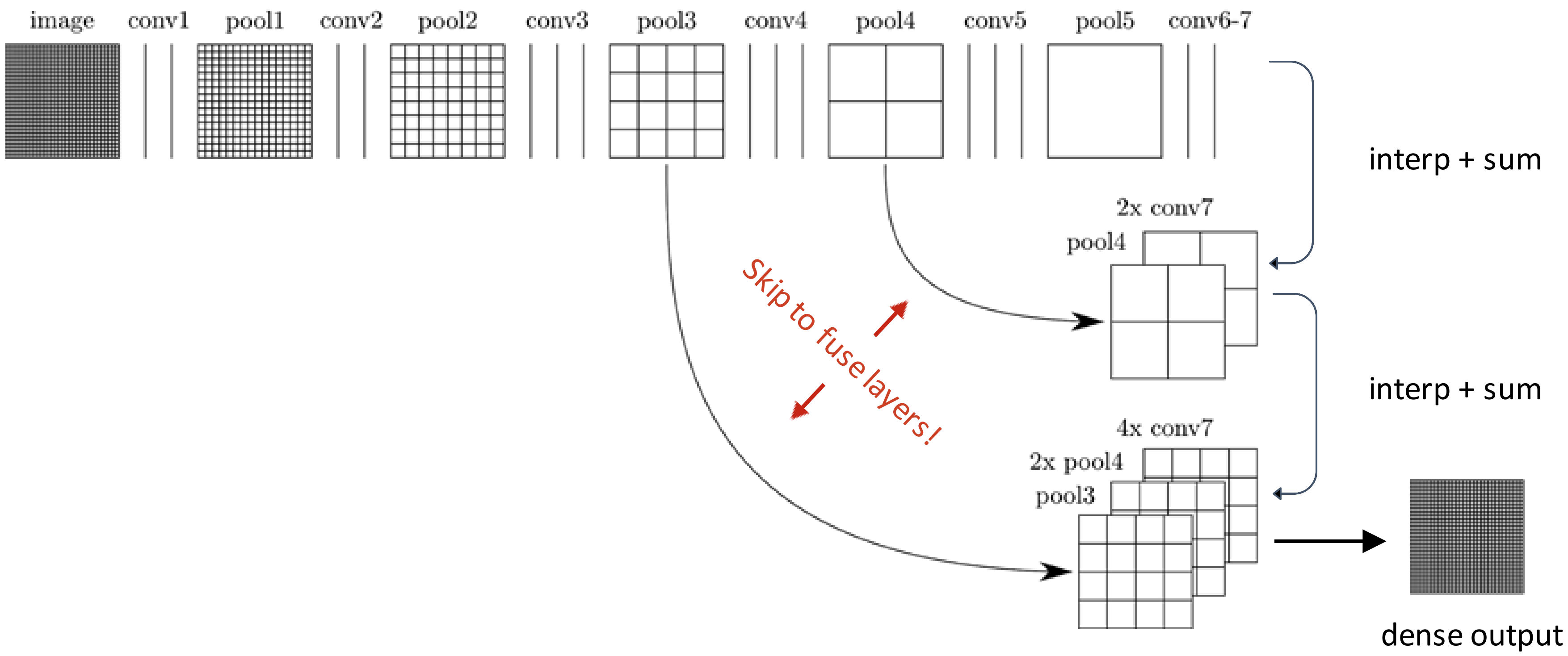
[Long et al.]

Learning upsampling kernels with skip layer refinement



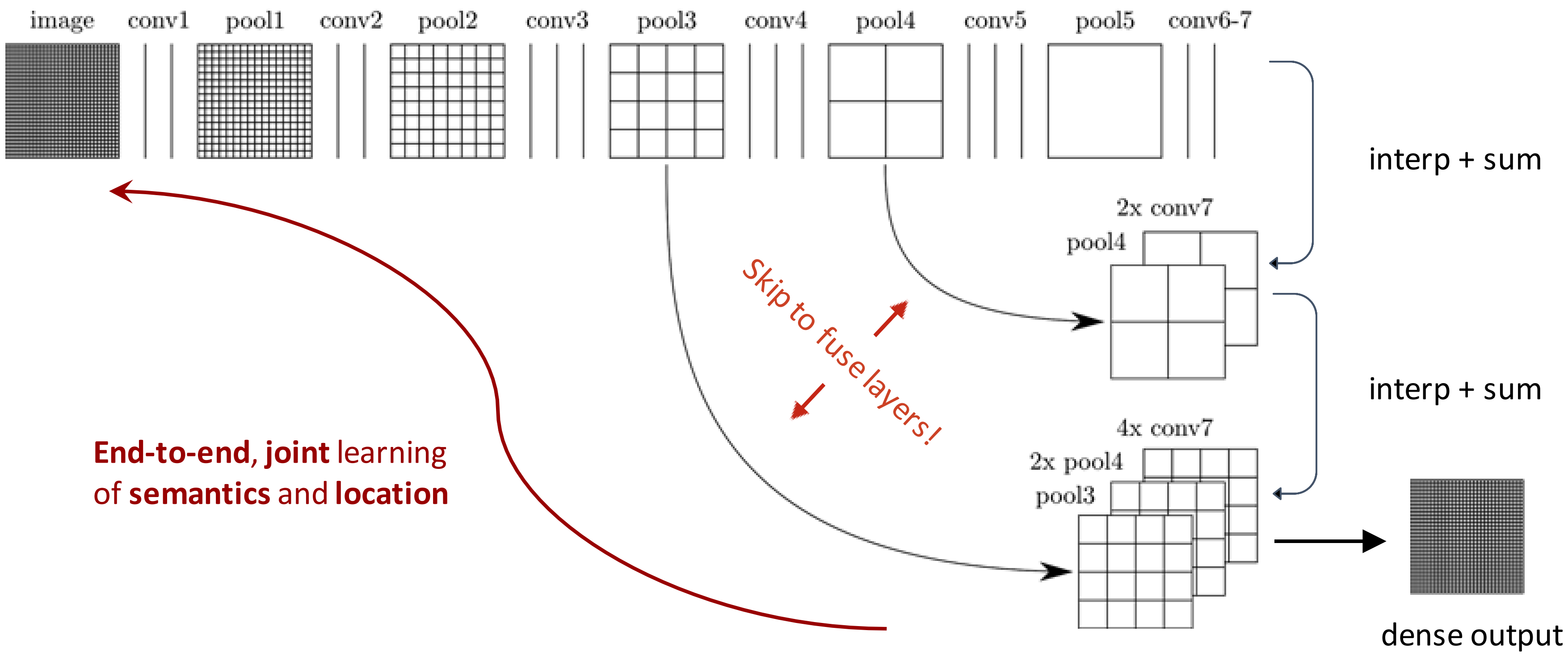
[Long et al.]

Learning upsampling kernels with skip layer refinement



[Long et al.]

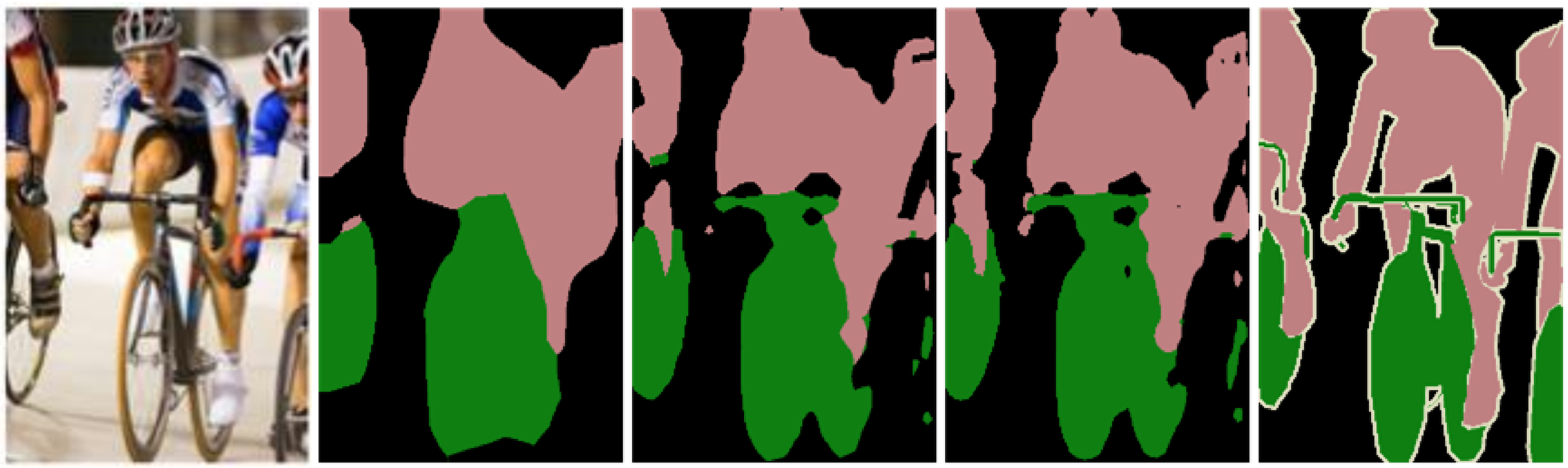
Learning upsampling kernels with skip layer refinement



[Long et al.]

Skip layer refinement

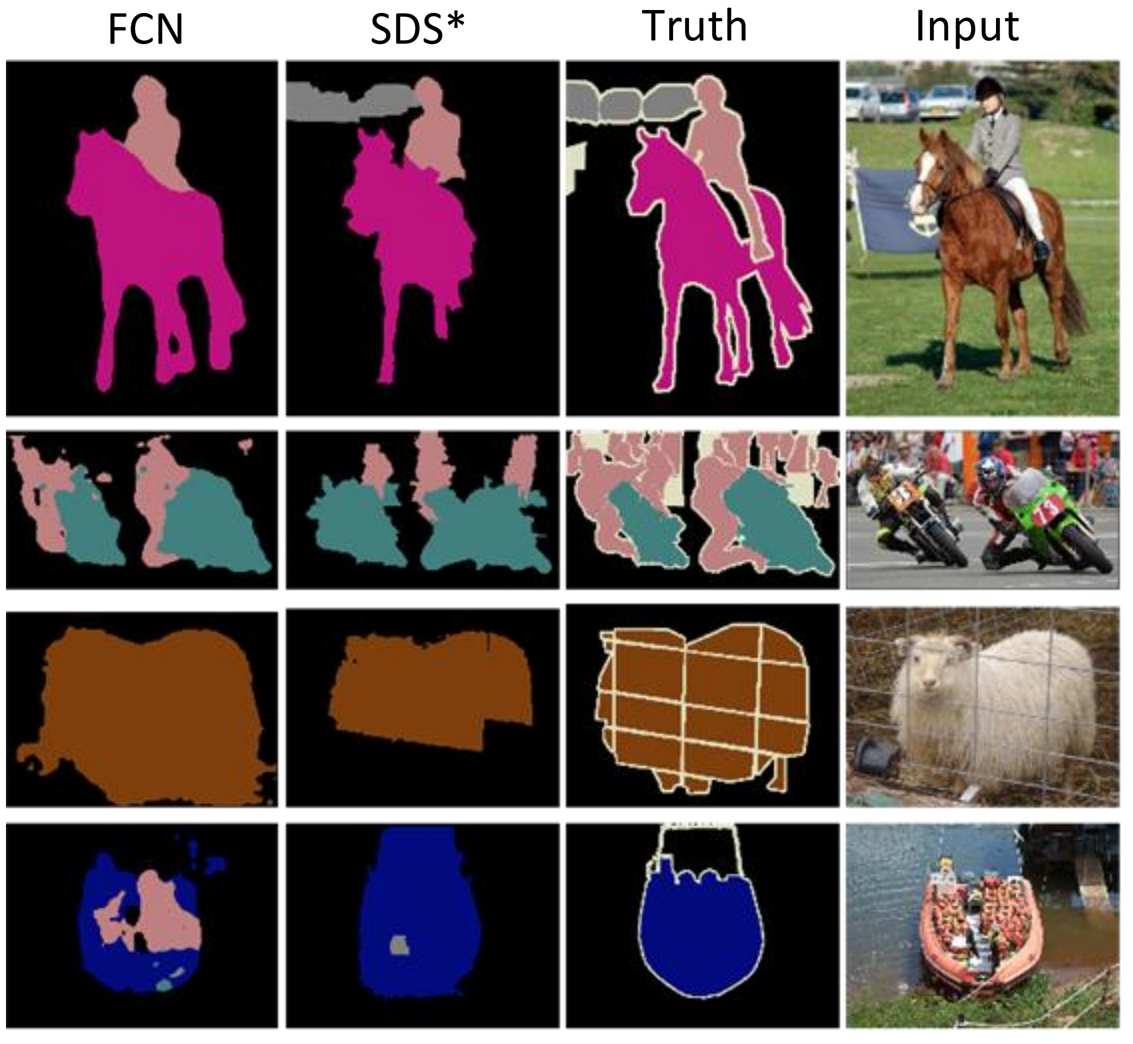
input image | stride 32 | stride 16 | stride 8 | ground truth



no skips | 1 skip | 2 skips

[Long et al.]

Results



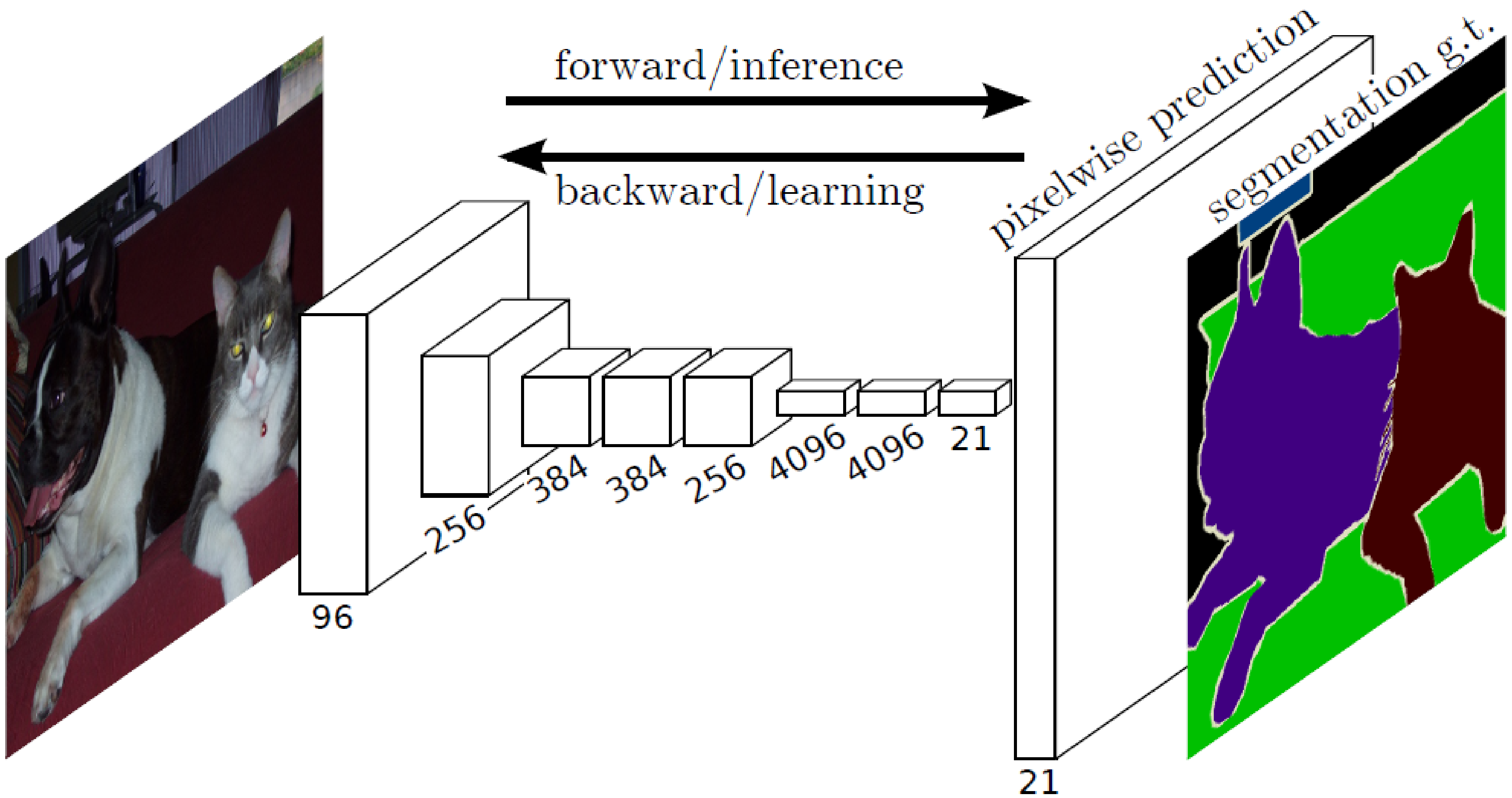
Relative to prior state-of-the-art SDS:

- 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

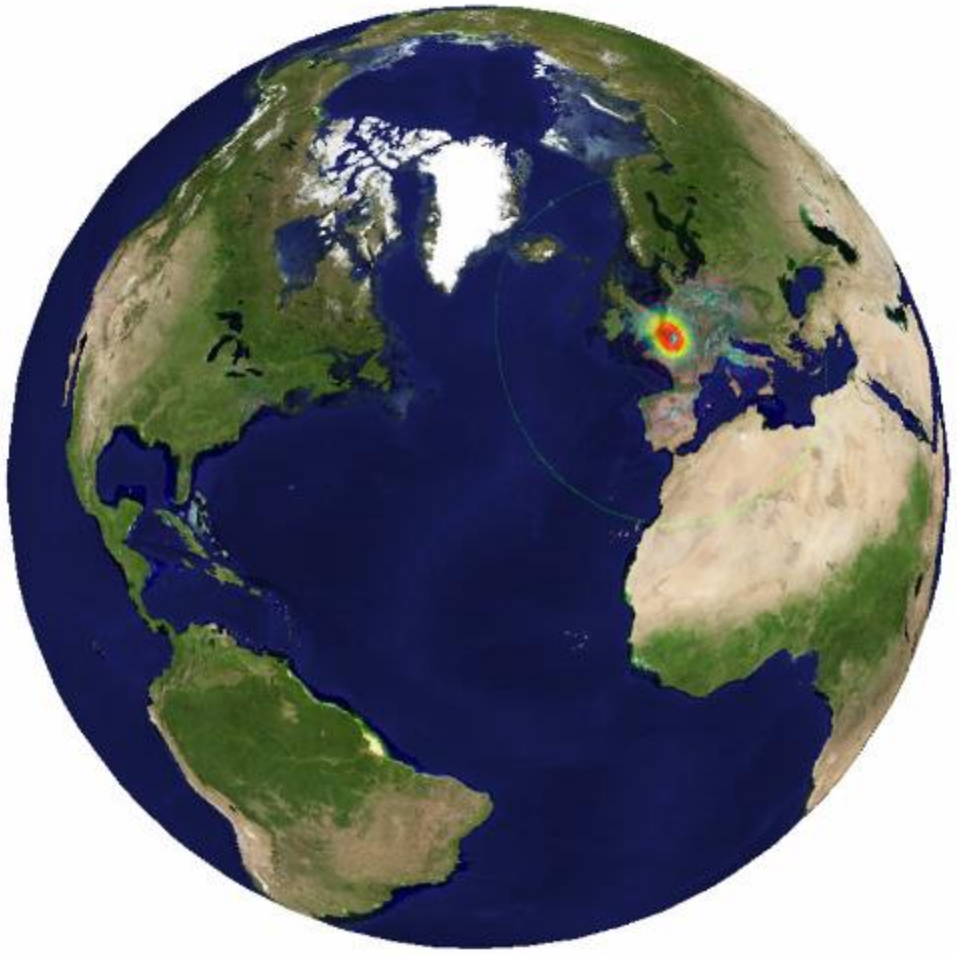
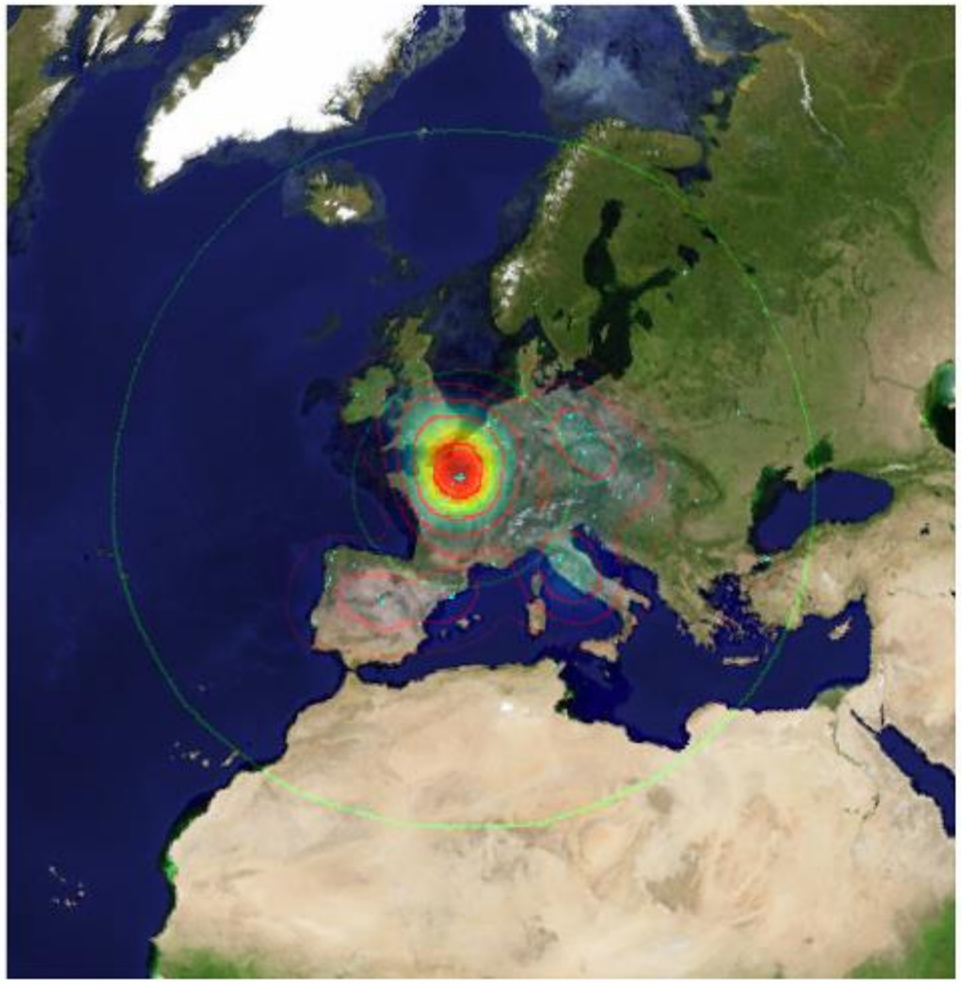
How much can an image tell about its geographic location?

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

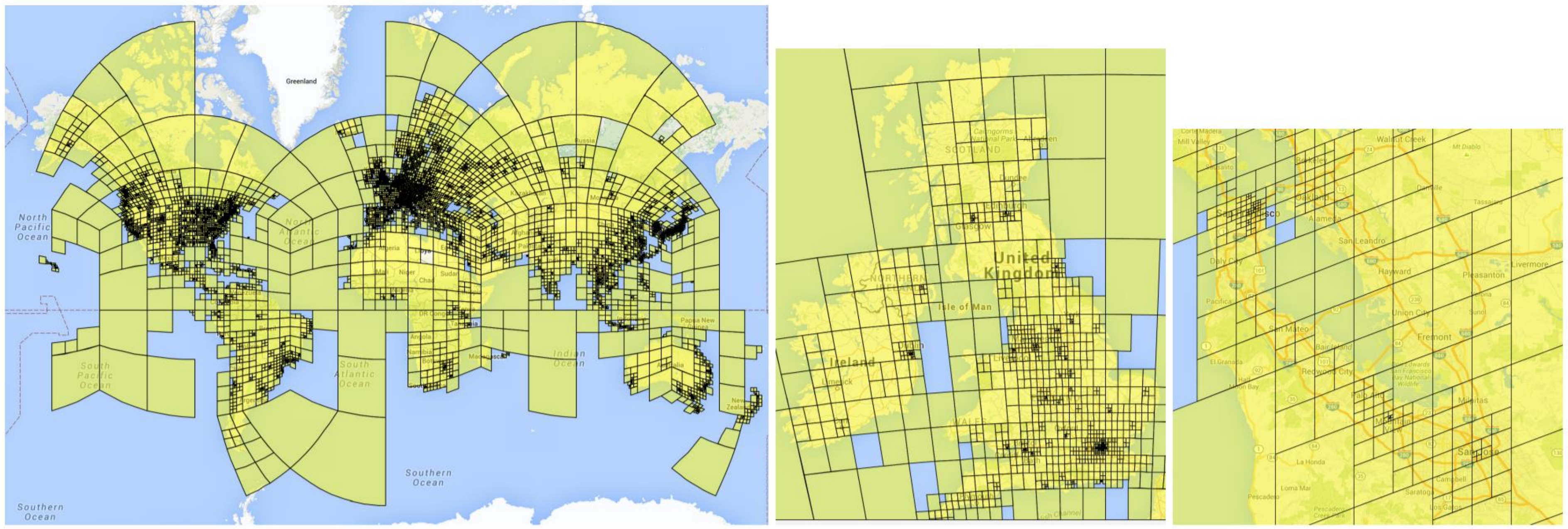


Figure 2. Left: Adaptive partitioning of the world into 26,263 S2 cells. Right: Detail views of Great Britain and Ireland and the San

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)

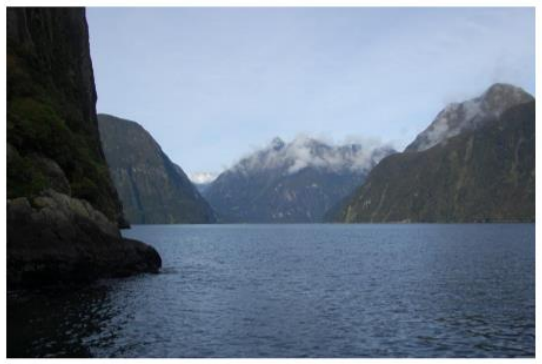
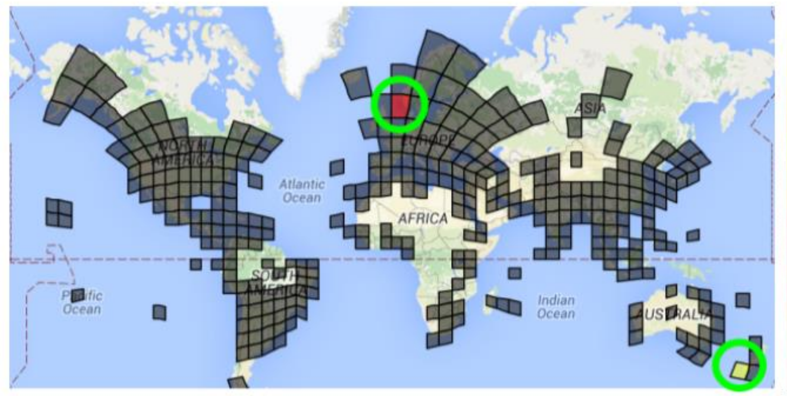


Photo CC-BY-NC by edwin.11



(b)

Photo CC-BY-NC by jonathanfh

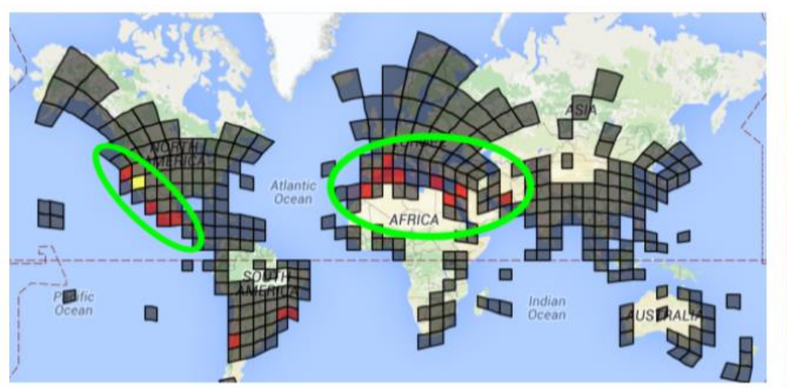



Photo by jamie.lovelock / CC BY NC Photo by MongoosePhotography / CC BY NC



Photo by Mister-E / CC BY NC Photo by dalangalma / CC BY NC Photo by siamjack / CC BY NC

Namibia / Botswana



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

Kauai, Hawaii

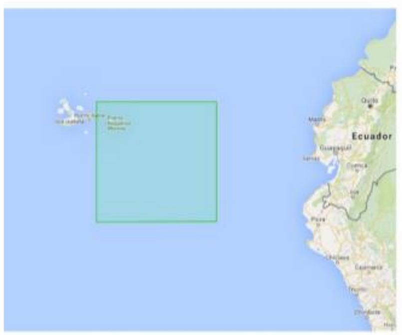




Photo by p.j.k. / CC BY NC Photo by victor408 / CC BY NC Photo by Domen Jakus / CC BY NC


Photo by cvanholder / CC BY NC Photo by rwoan / CC BY NC

Galapagos Islands

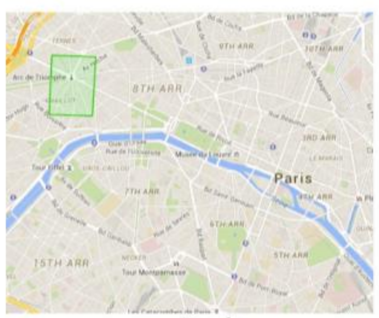




Photo by feliven / CC BY NC Photo by fred_v / CC BY NC Photo by Turansa Tours / CC BY NC


Photo by JA_FS / CC BY NC Photo by CedEm photographies / CC BY NC

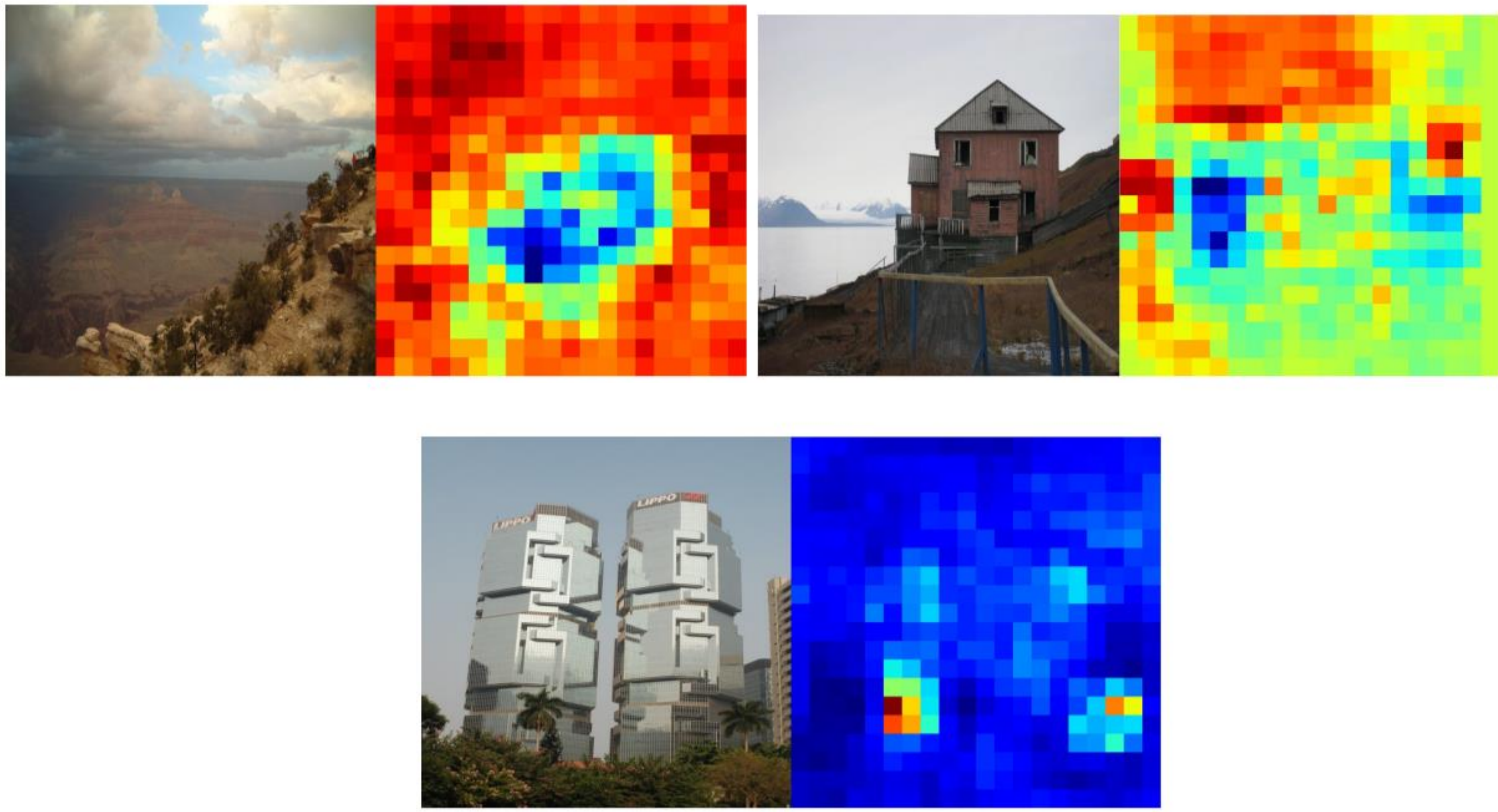
Paris

PlaNet vs im2gps (2008, 2009)

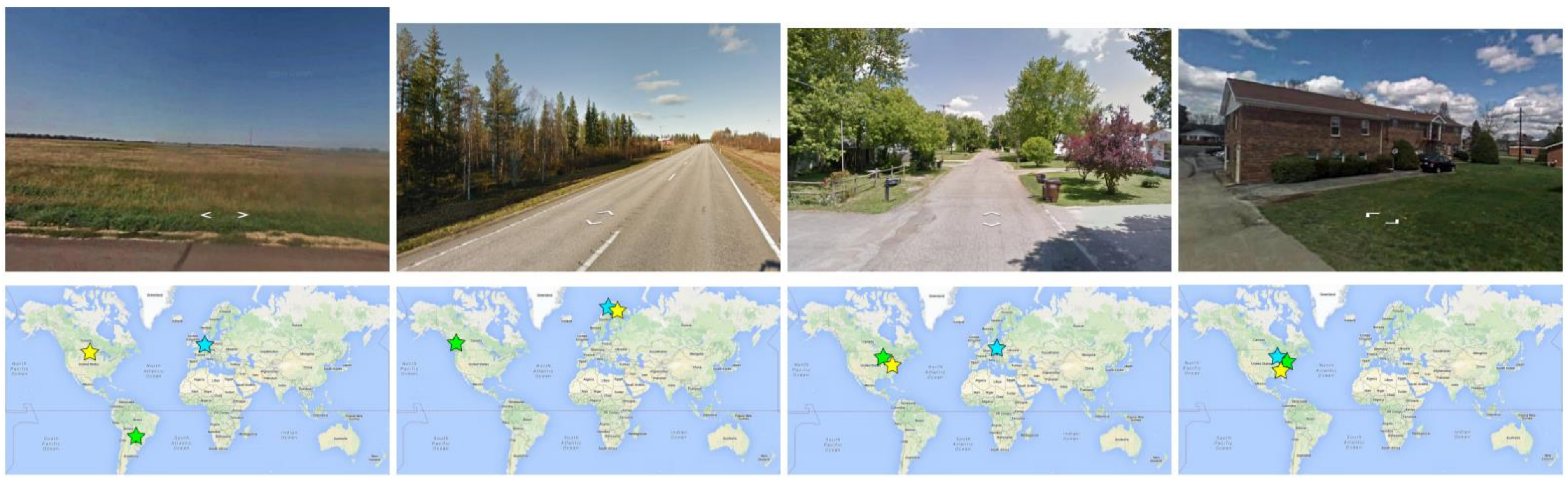
Method	**Street 1 km**	**City 25 km**	**Region 200 km**	**Country 750 km**	**Continent 2500 km**
Im2GPS (orig) [17]		12.0%	15.0%	23.0%	47.0%
Im2GPS (new) [18]	2.5%	21.9%	32.1%	35.4%	51.9%
PlaNet	**8.4%**	**24.5%**	**37.6%**	**53.6%**	**71.3%**

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

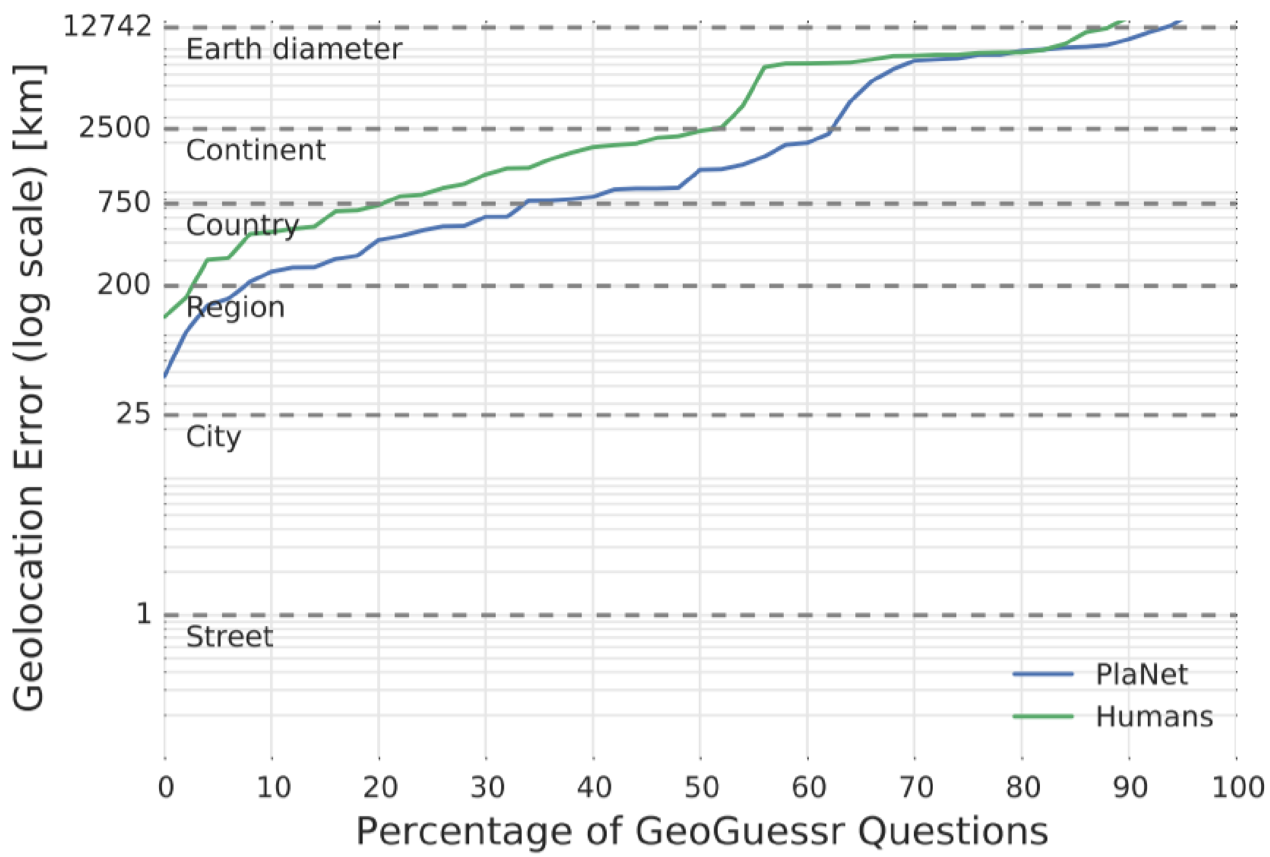
Spatial support for decision



PlaNet vs Humans



PlaNet vs. Humans



PlaNet summary

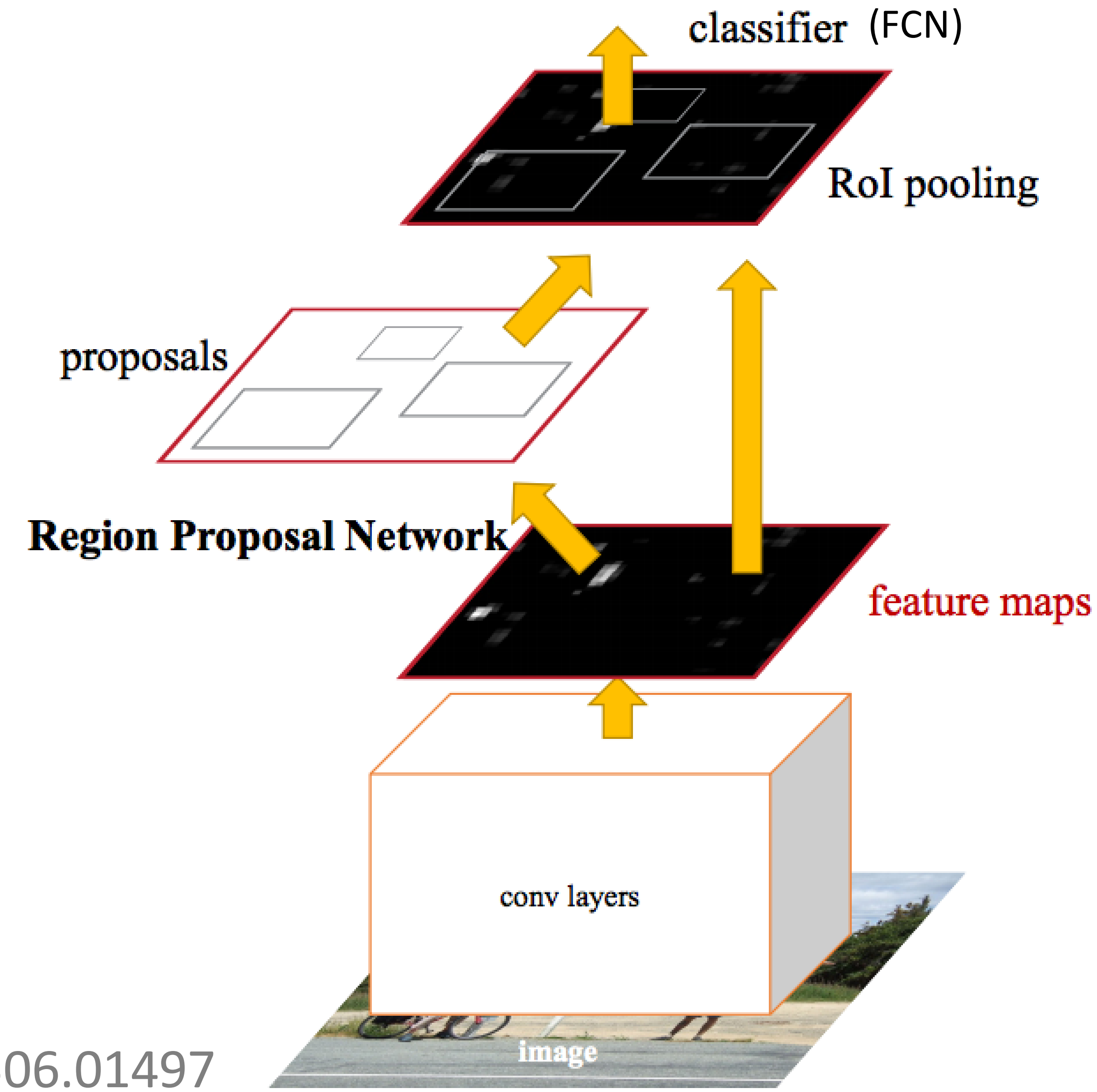
- Very fast geolocalization method by categorization.
- Uses far more training data than previous work (im2gps)
- Better than humans!

Even more: Faster R-CNN

'Region Proposal Network' uses CNN feature maps.

Then, FCN on top to classify.

End to end object detection.



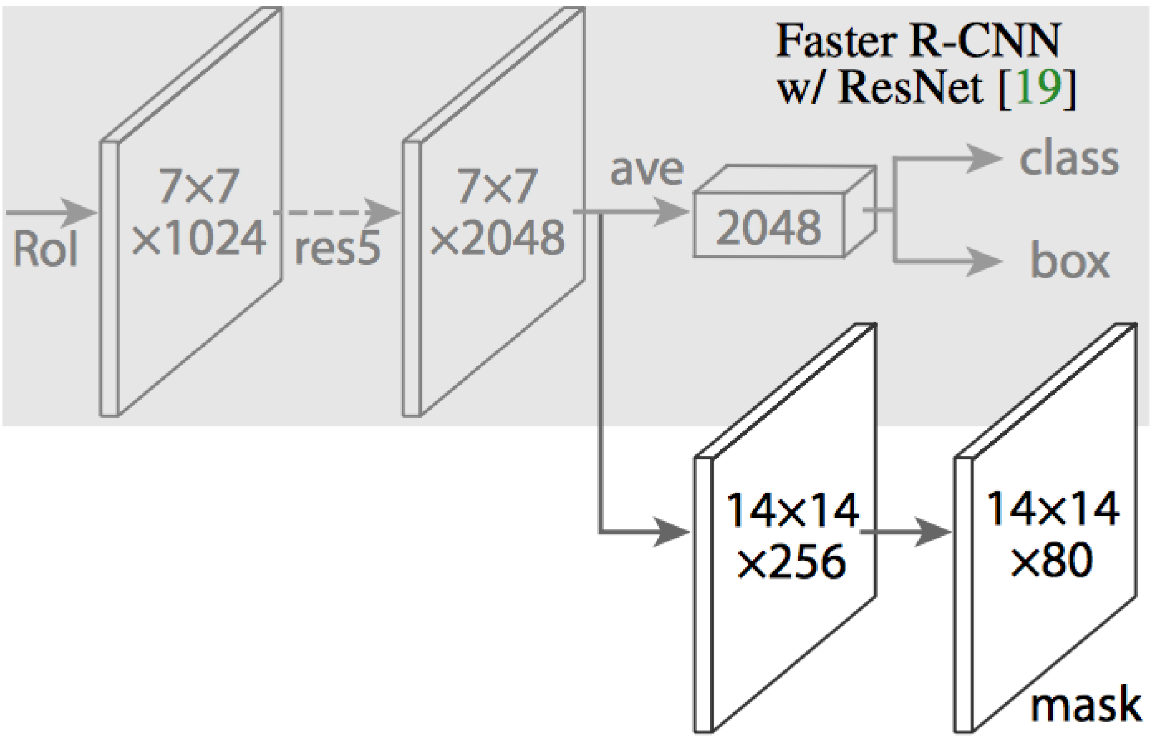
Ren et al. 2016
https://arxiv.org/abs/1506.01497

Even more! Mask R-CNN

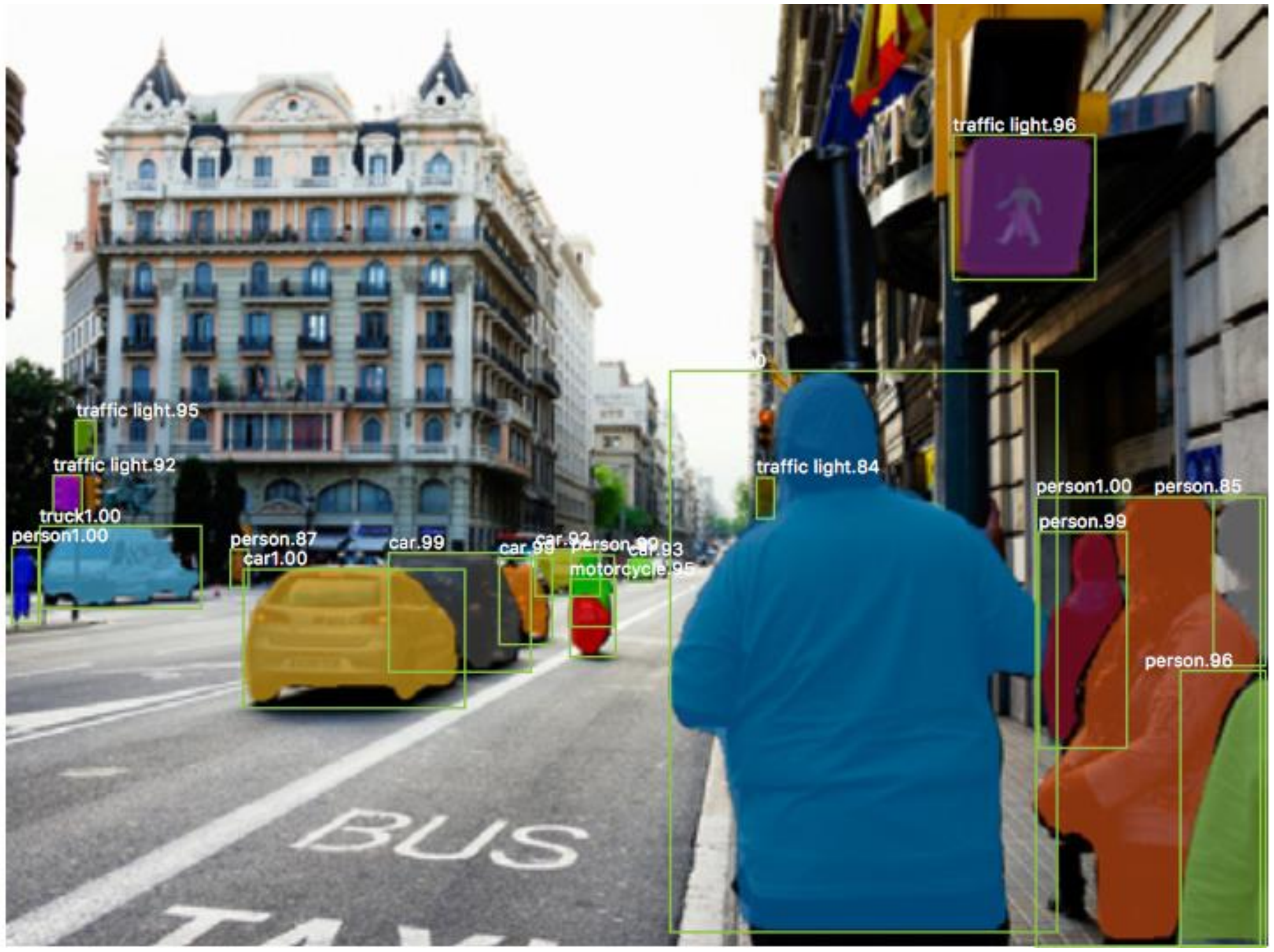
Extending Faster R-CNN for Pixel Level Segmentation

He et al. - https://arxiv.org/abs/1703.06870

Add new training data: segmentation masks



Second output which is segmentation mask



traffic light.96
traffic light.95
traffic light.92
traffic light.84
truck1.00
person1.00
person.87
car1.00
car.99
motorcycle.95
person1.00
person.85
person.99
person.96