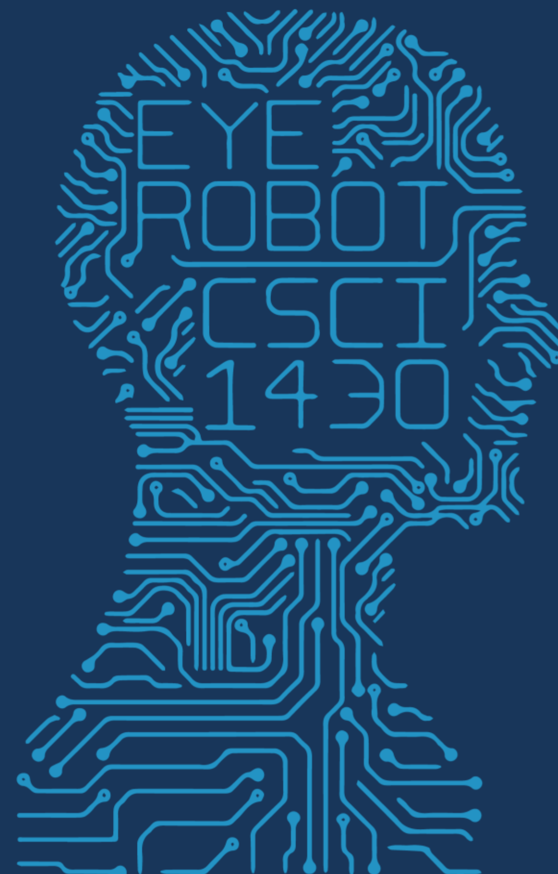
1950

Future Vision



18 March 2019

Computer Vision

BOLT 2019 Fall Trip Application is LIVE!

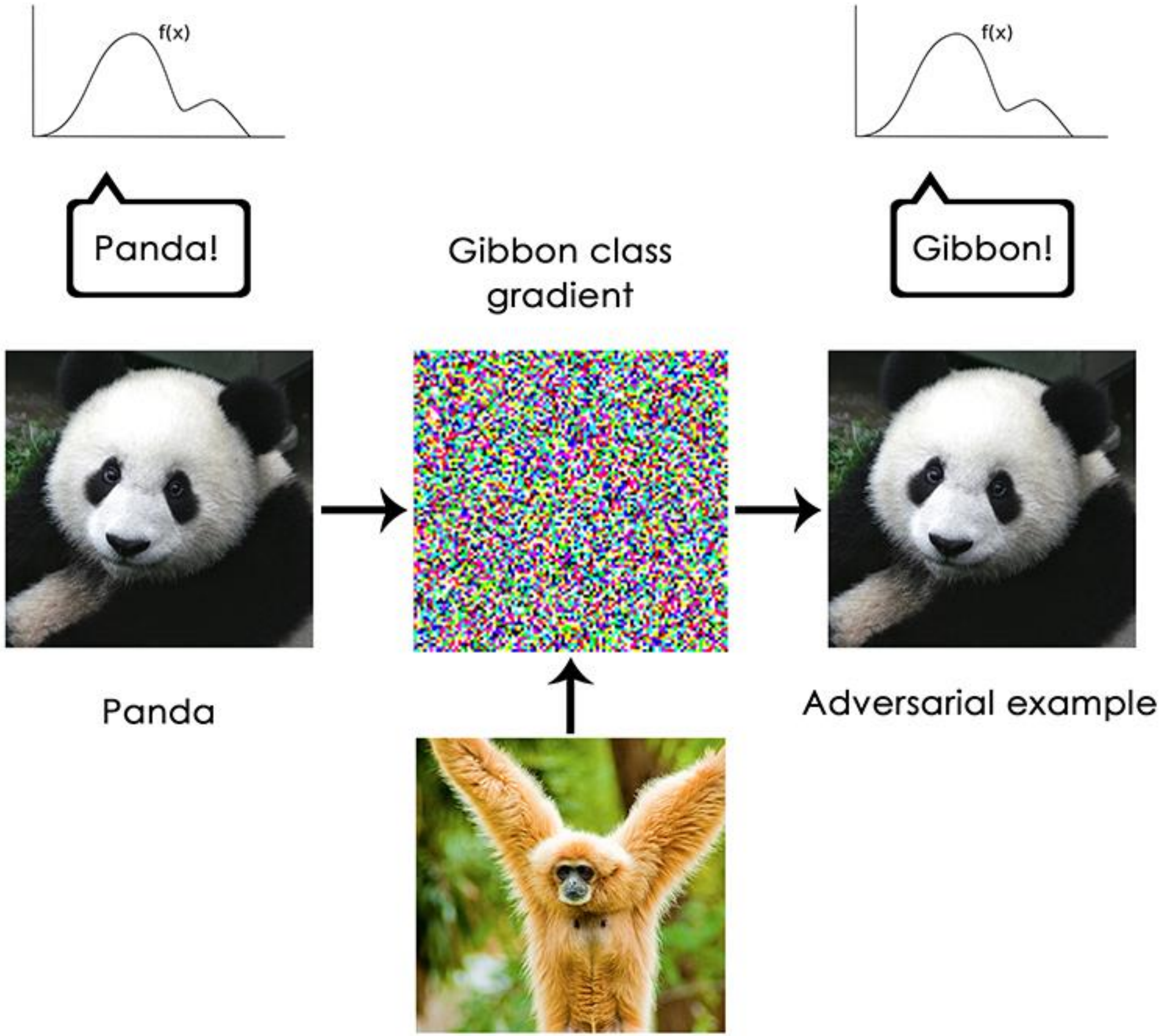
Info sessions on 4/8 @ 7PM in Friedman 102 and 4/9 @ 7PM in Salomon 003

Brown Outdoor Leadership Training is recruiting for rising sophomores, new transfers and new RUEs for our 2019 Fall Trip and Program!

BOLT commences with a 5-day backpacking trip in the White Mountains, NH right before the Fall semester starts. It continues throughout the semester with group get-togethers, leadership development workshops & community events that aim to foster community, mentorship, personal reflection & leadership in the outdoors & back at Brown.

Everyone is welcome to apply - no hiking experience needed! Financial aid, gear and boot rentals available!

For more information: email bolt@brown.edu and visit brown.edu/bolt



Francois Chollet - https://blog.keras.io/the-limitations-of-deep-learning.html

3rd November 2017

Technology

Single pixel change fools AI programs

Tiny changes can make image recognition systems think a school bus is an ostrich, find scientists.

3 hours ago | Technology

Algorithm learns to recognise natural beauty

Artificial intelligence fools security

AI used to detect breast cancer

BBC | Your account | News | Sport | Weather | Shop | Earth | Travel | More | Search

NEWS

Home | Video | World | US & Canada | UK | Business | Tech | Science | Stories | Entertainment & Arts | Health | More

Technology

Single pixel change fools AI programs

Tiny changes can make image recognition systems think a school bus is an ostrich, find scientists.

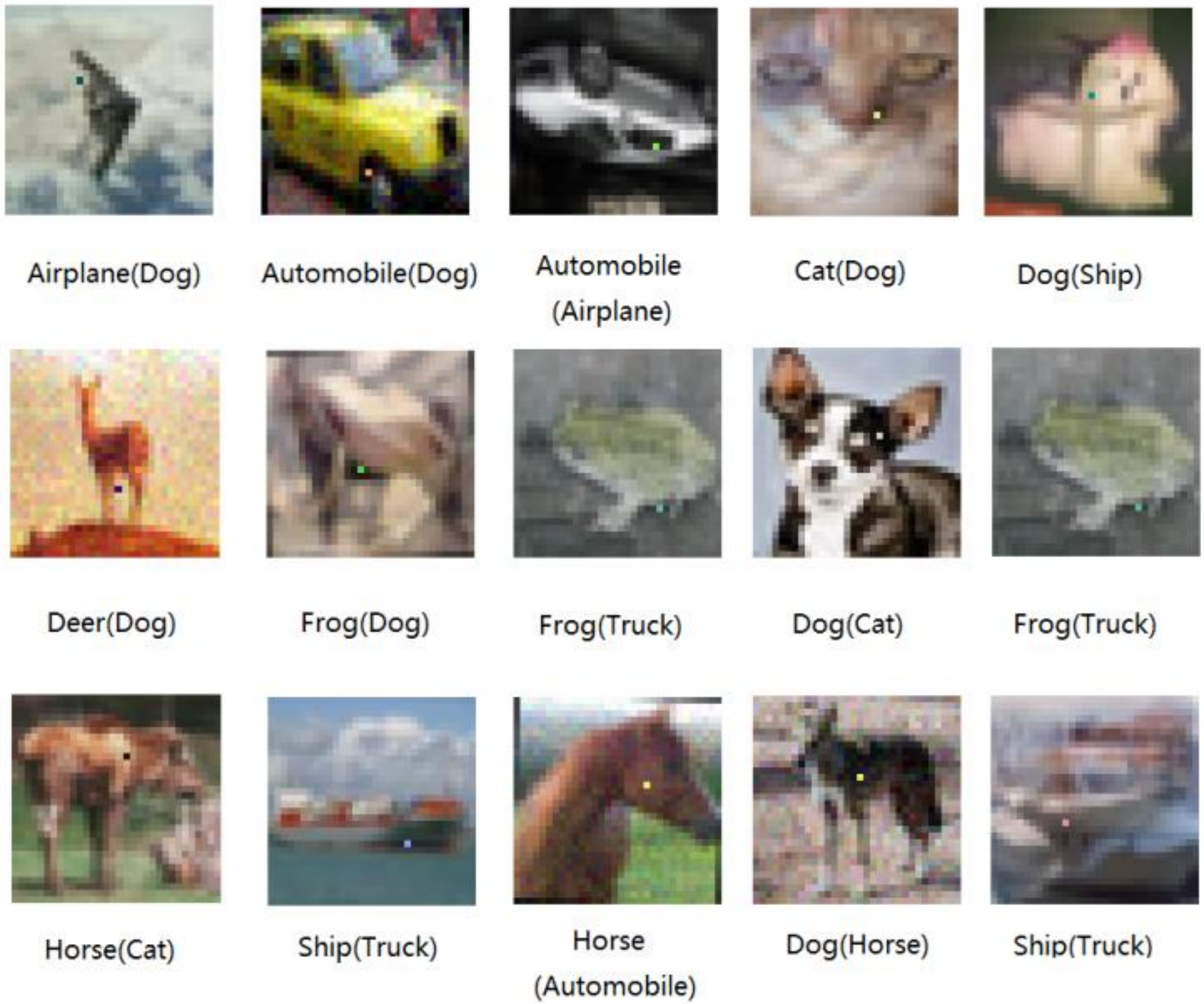
3 hours ago | Technology

Algorithm learns to recognise natural beauty

Artificial intelligence fools security

AI used to detect breast cancer

Yes, it’s a blue brain image : (

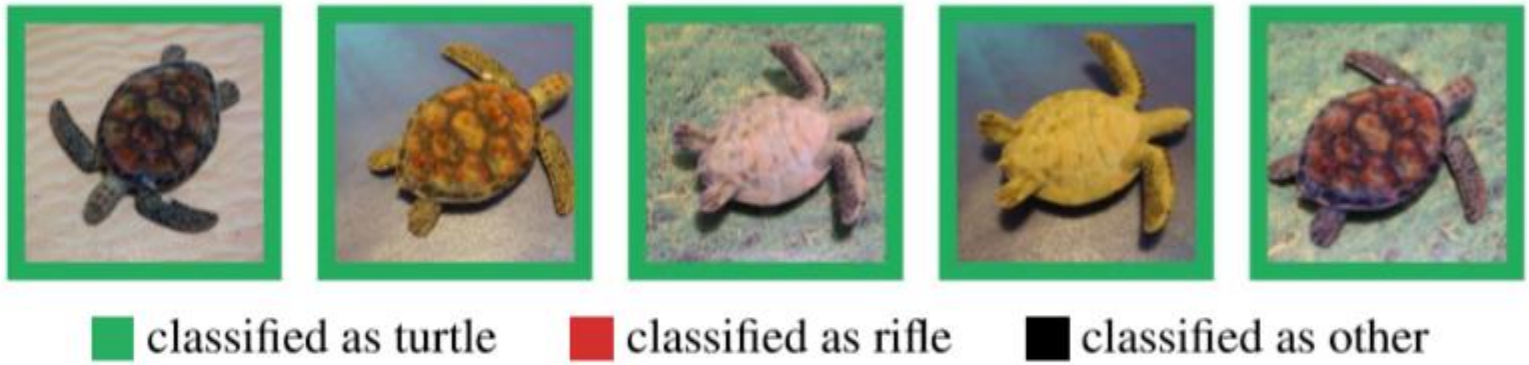


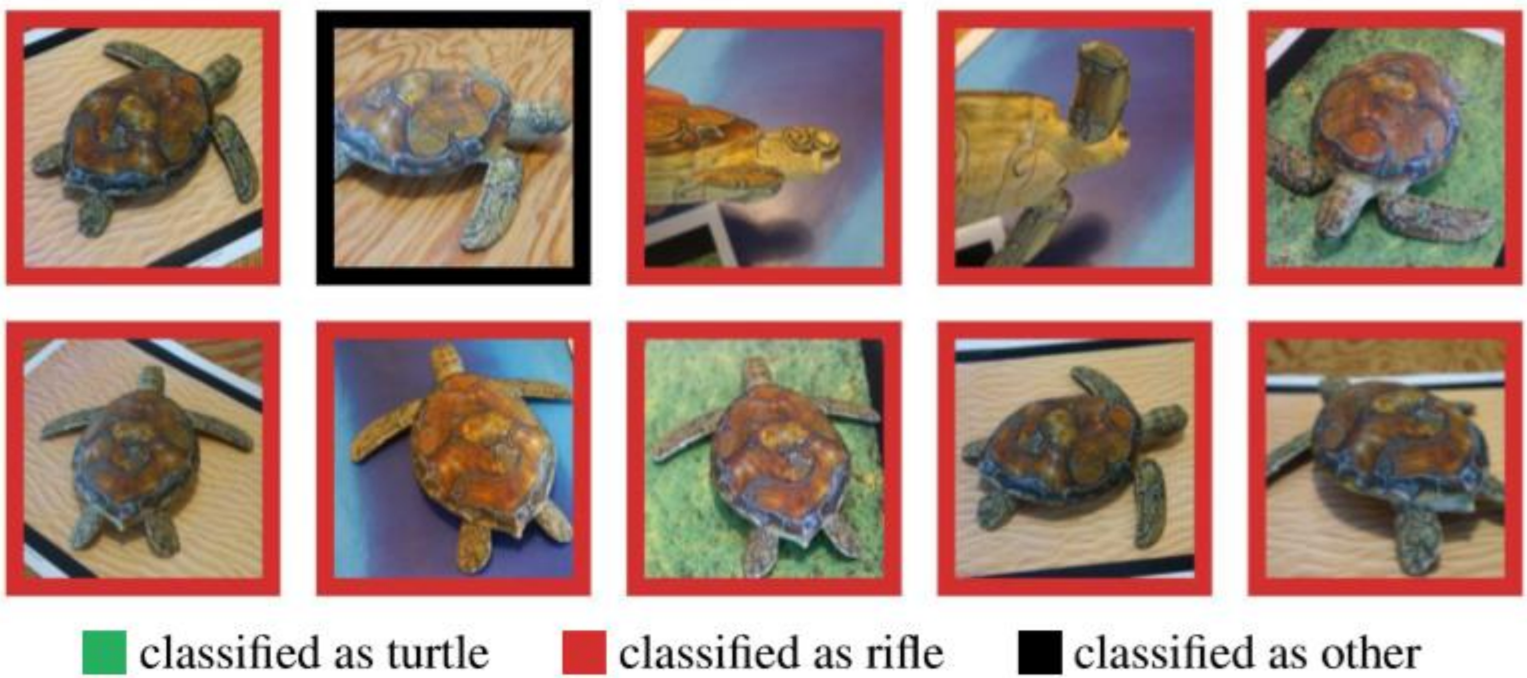
Su et al., One pixel attack for fooling deep neural networks
https://arxiv.org/abs/1710.08864

https://arxiv.org/pdf/1707.07397.pdf

Synthesizing Robust Adversarial Examples

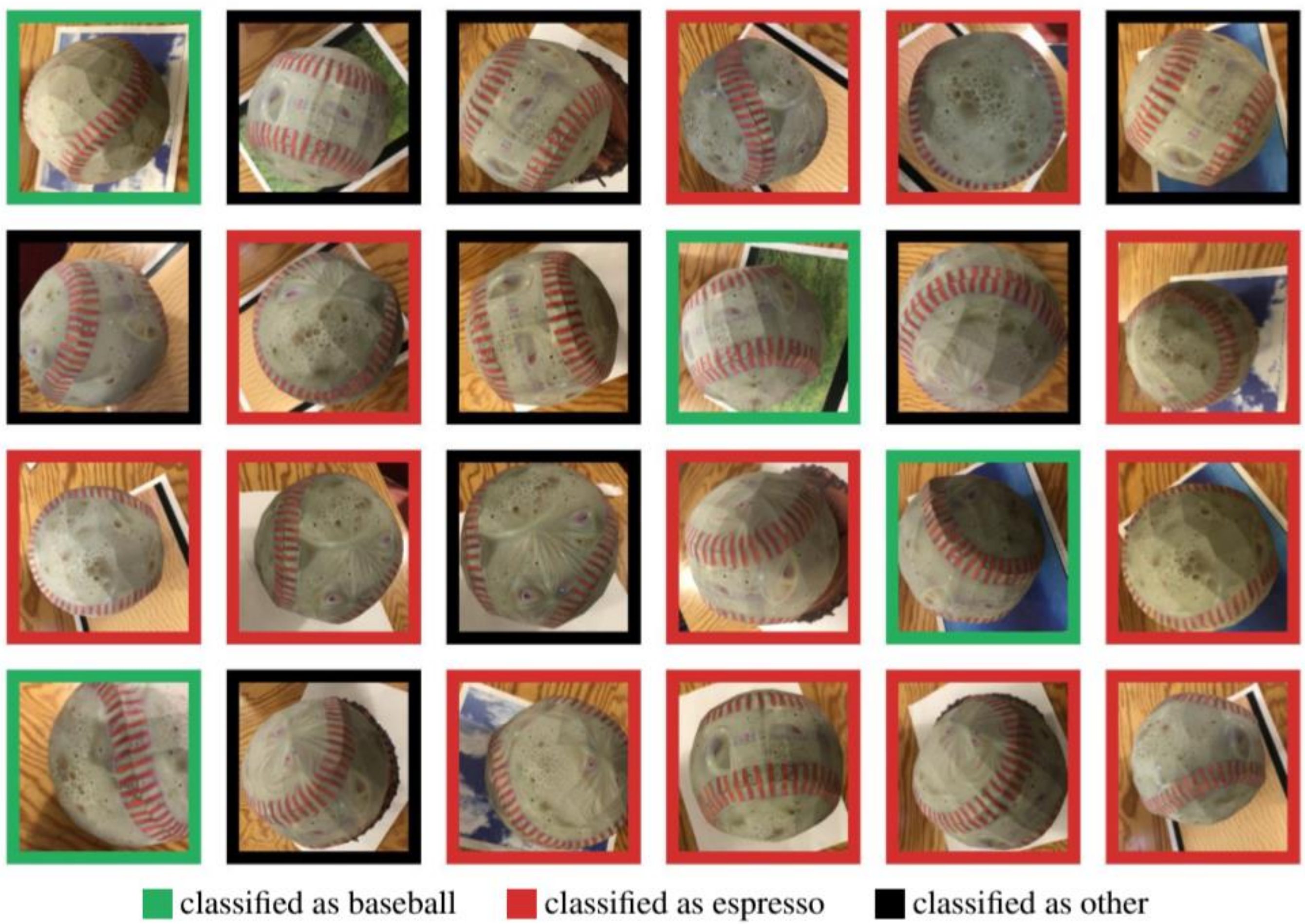
Anish Athalye*[1,2], Logan Engstrom*[1,2], Andrew Ilyas*[1,2], Kevin Kwok[2]
[1]Massachusetts Institute of Technology, [2]LabSix
{aathalye,engstrom,ailyas}@mit.edu, kevin@labsix.org

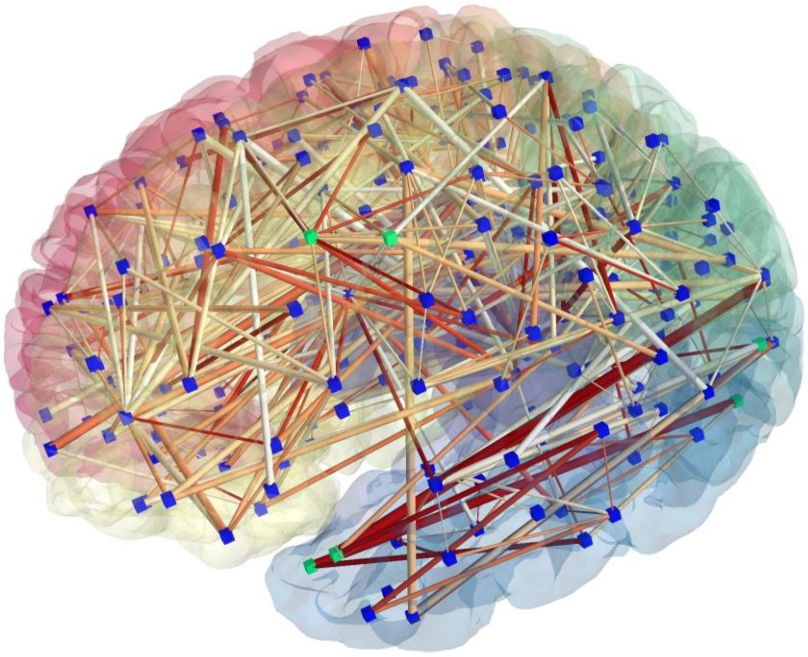





Fooling Neural Networks
in the Real World
labsix
rifle
shield, buck
revolver, si


classified as baseball
classified as espresso
classified as other

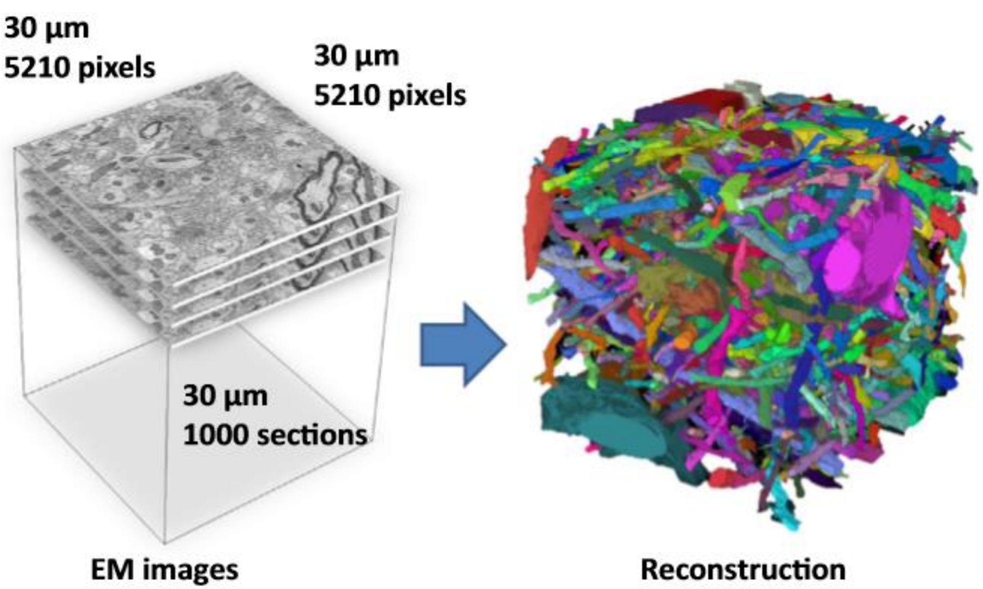

classified as baseball
classified as espresso
classified as other



Connectomics: Neural nets for neural nets

[Patric Hagmann]

Vision for understanding the brain



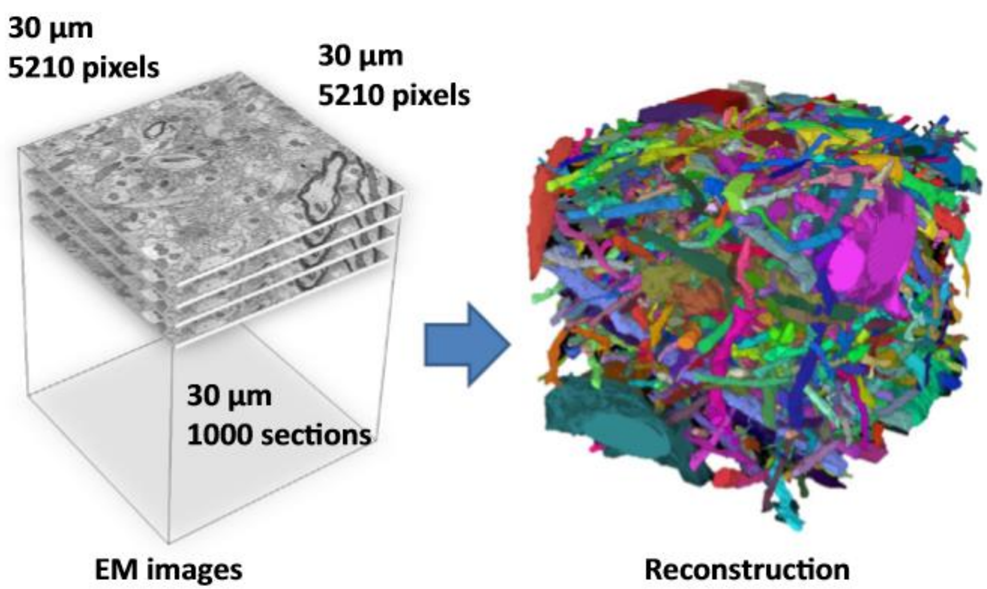
1mm cubed of brain

Image at 5-30 nanometers

How much data?

[Kaynig-Fittkau et al.]

Vision for understanding the brain



1mm cubed of brain

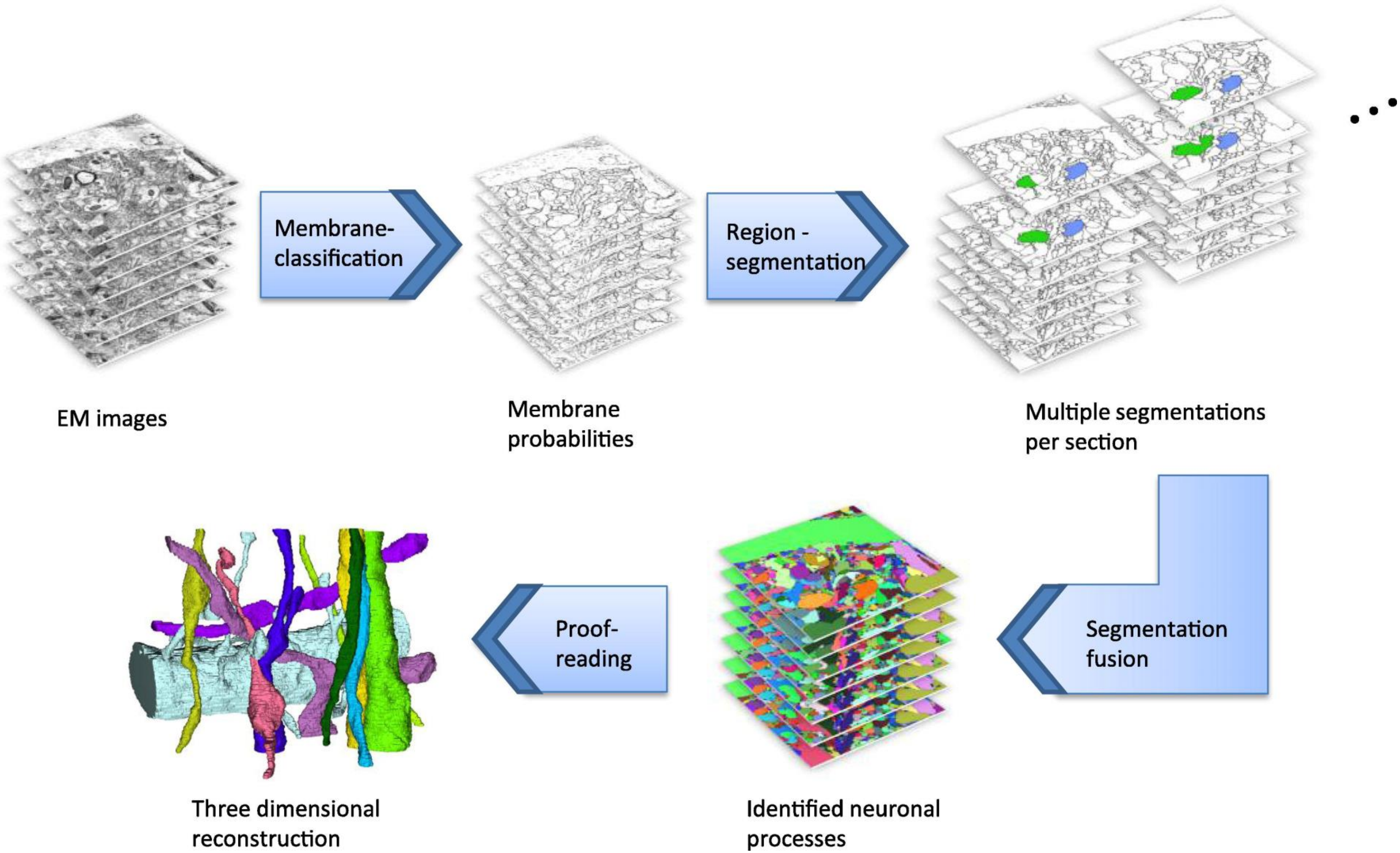
Image at 5-30 nanometers

How much data?

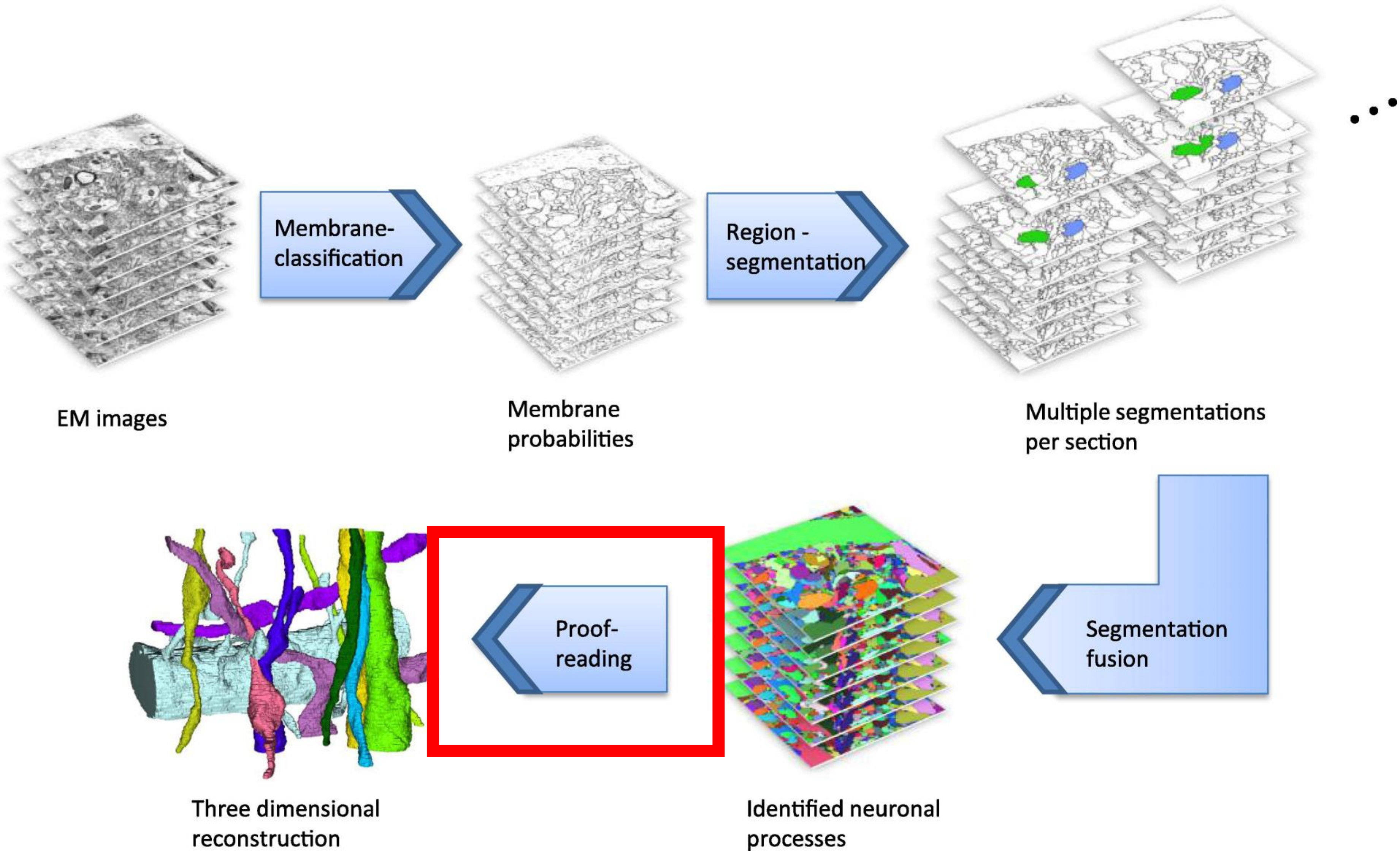
1 Petabyte –
1,000,000,000,000,000

~ All photos uploaded to Facebook per day

[Kaynig-Fittkau et al.]



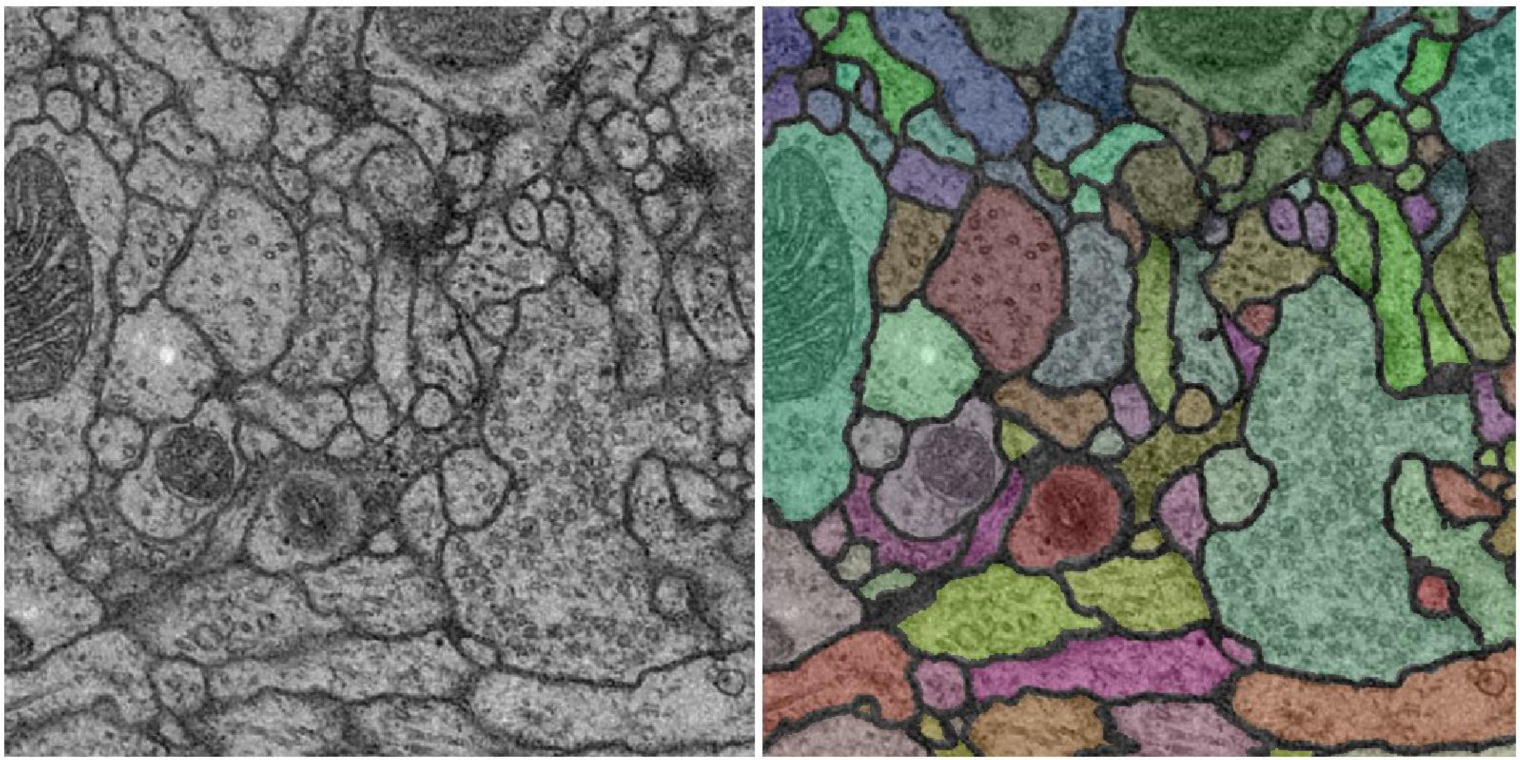
[Kaynig-Fittkau et al.]

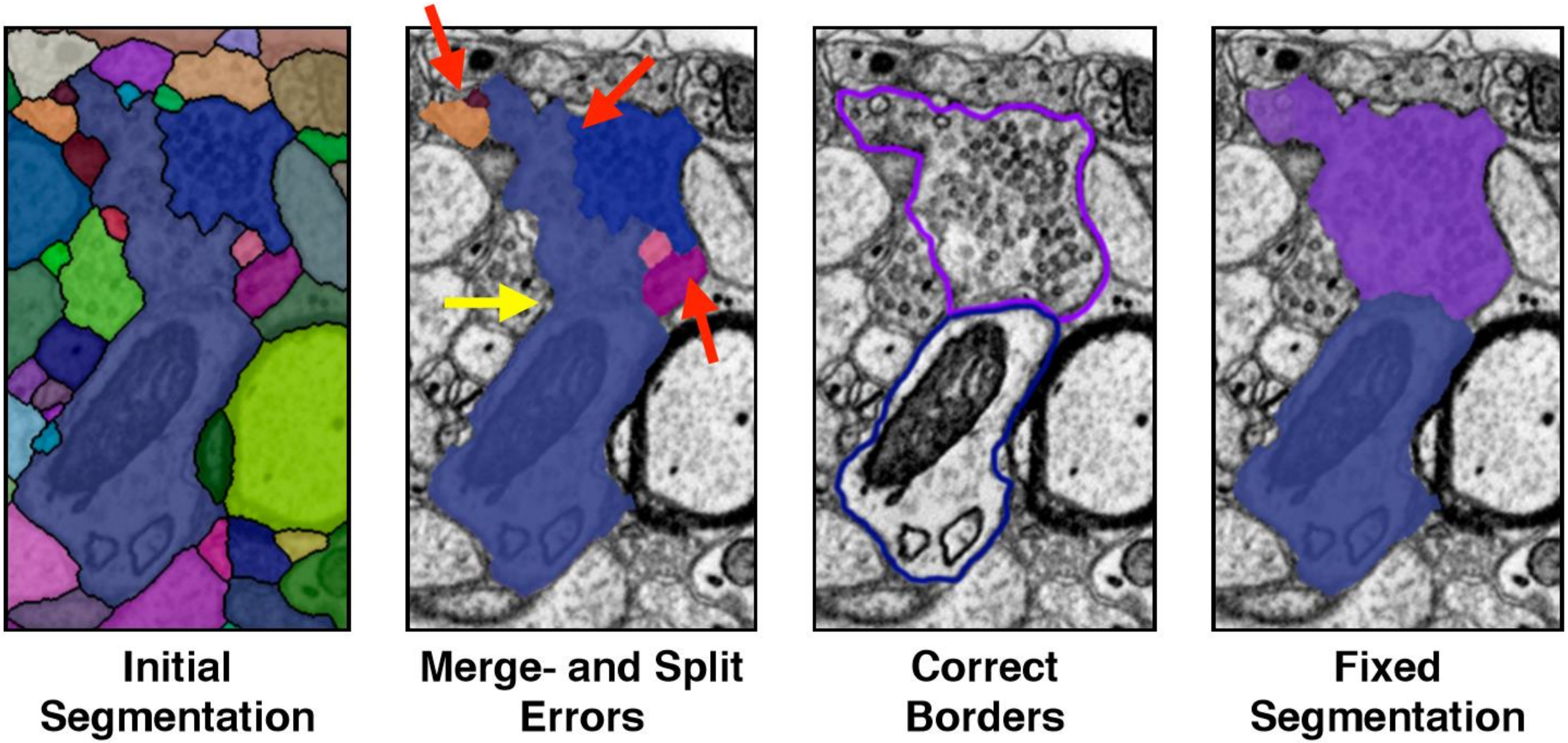


[Kaynig-Fittkau et al.]

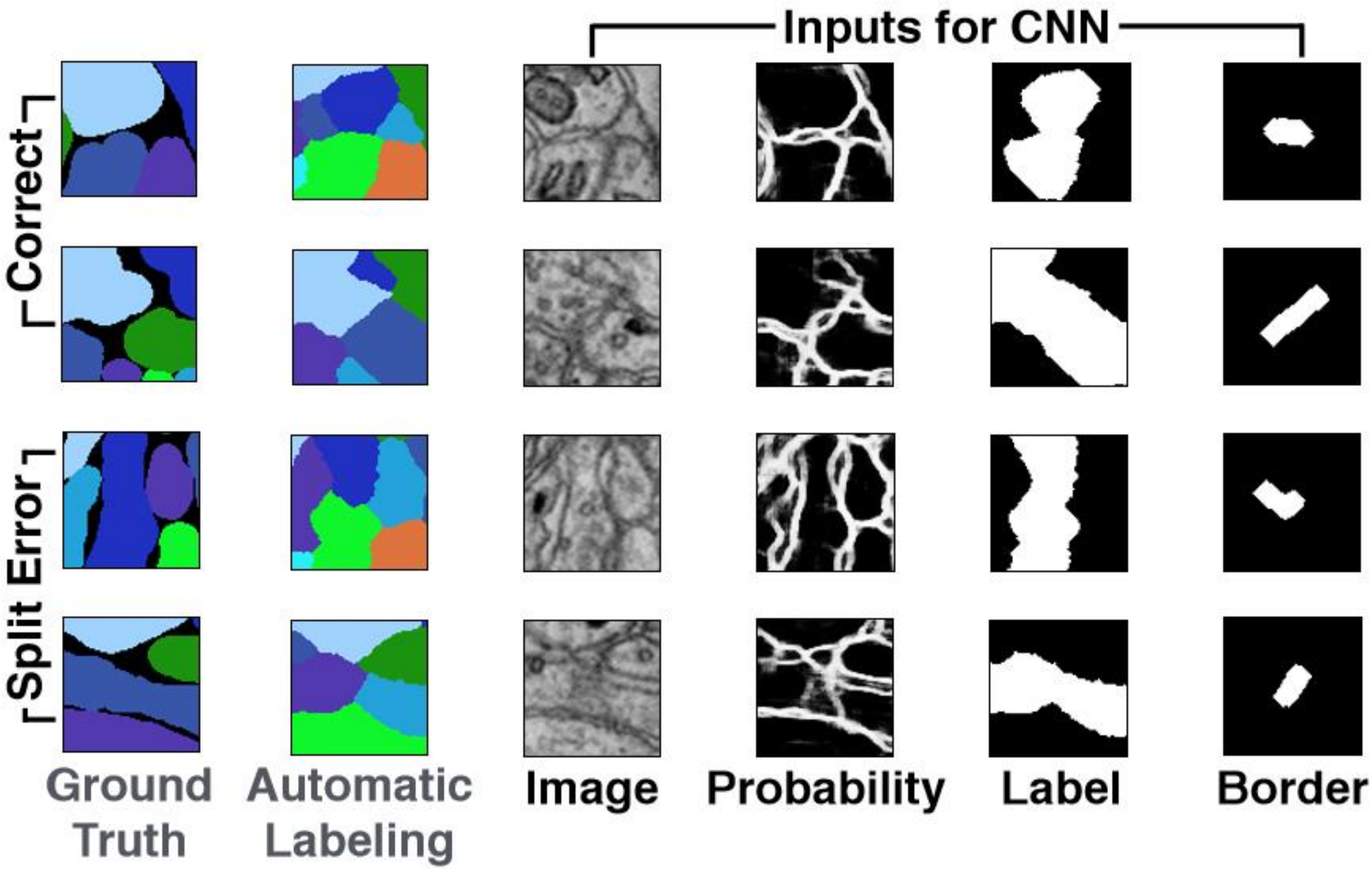
Vision for understanding the brain

Instance segmentation (but the instances sometimes look quite different!)




Initial Segmentation
Merge- and Split Errors
Correct Borders
Fixed Segmentation

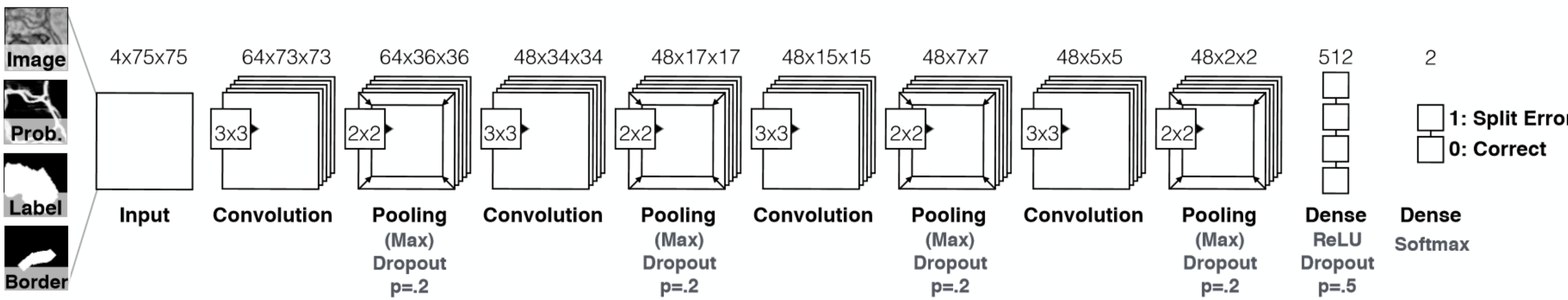
[Haehn et al.]



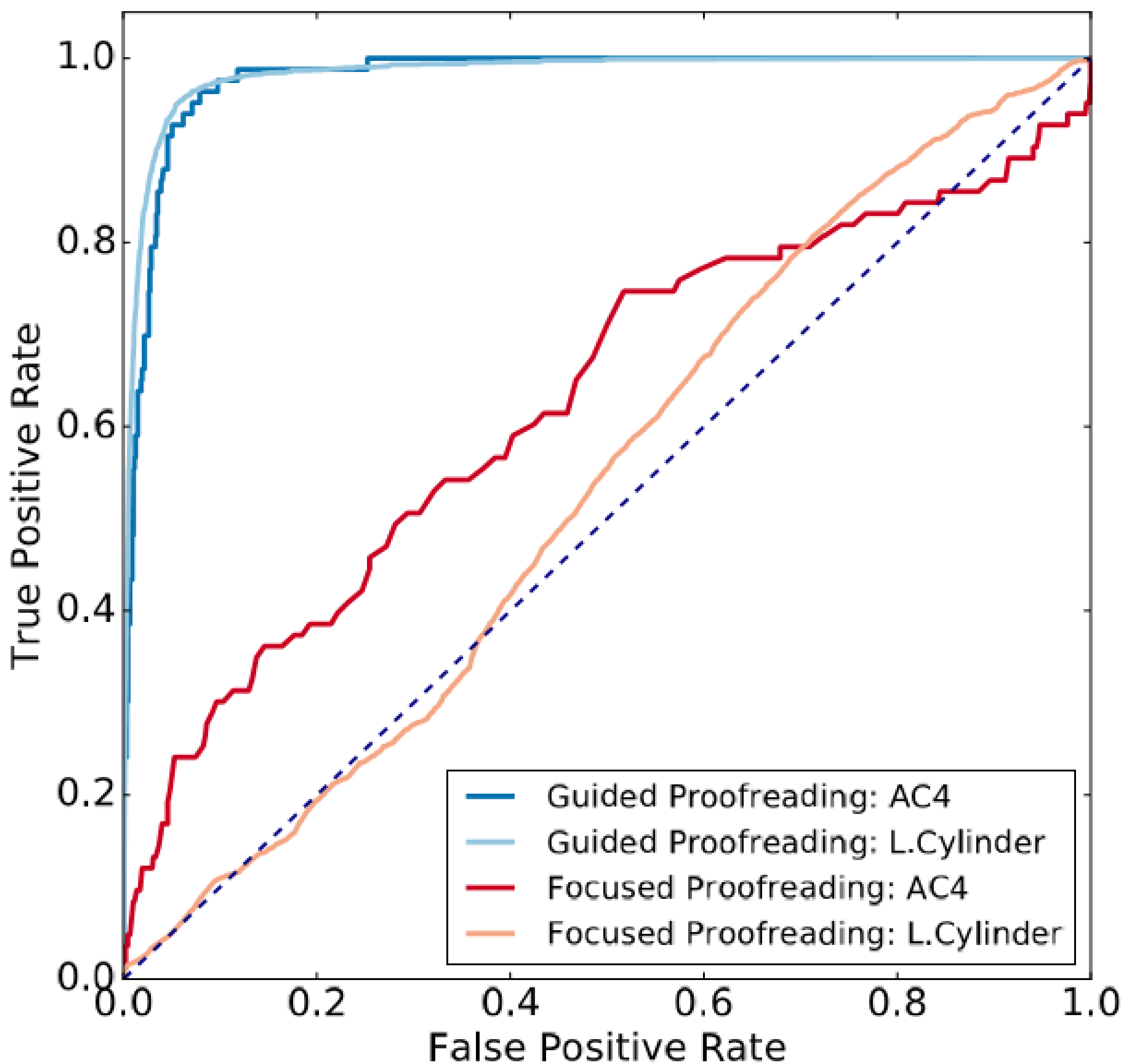
Inputs for CNN
Correct
Split Error
Ground Truth
Automatic Labeling
Image
Probability
Label
Border

[Haehn et al.]

Network Architecture



[Haehn et al.]


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
True Positive Rate
False Positive Rate
Guided Proofreading: AC4
Guided Proofreading: L.Cylinder
Focused Proofreading: AC4
Focused Proofreading: L.Cylinder

[Haehn et al.]

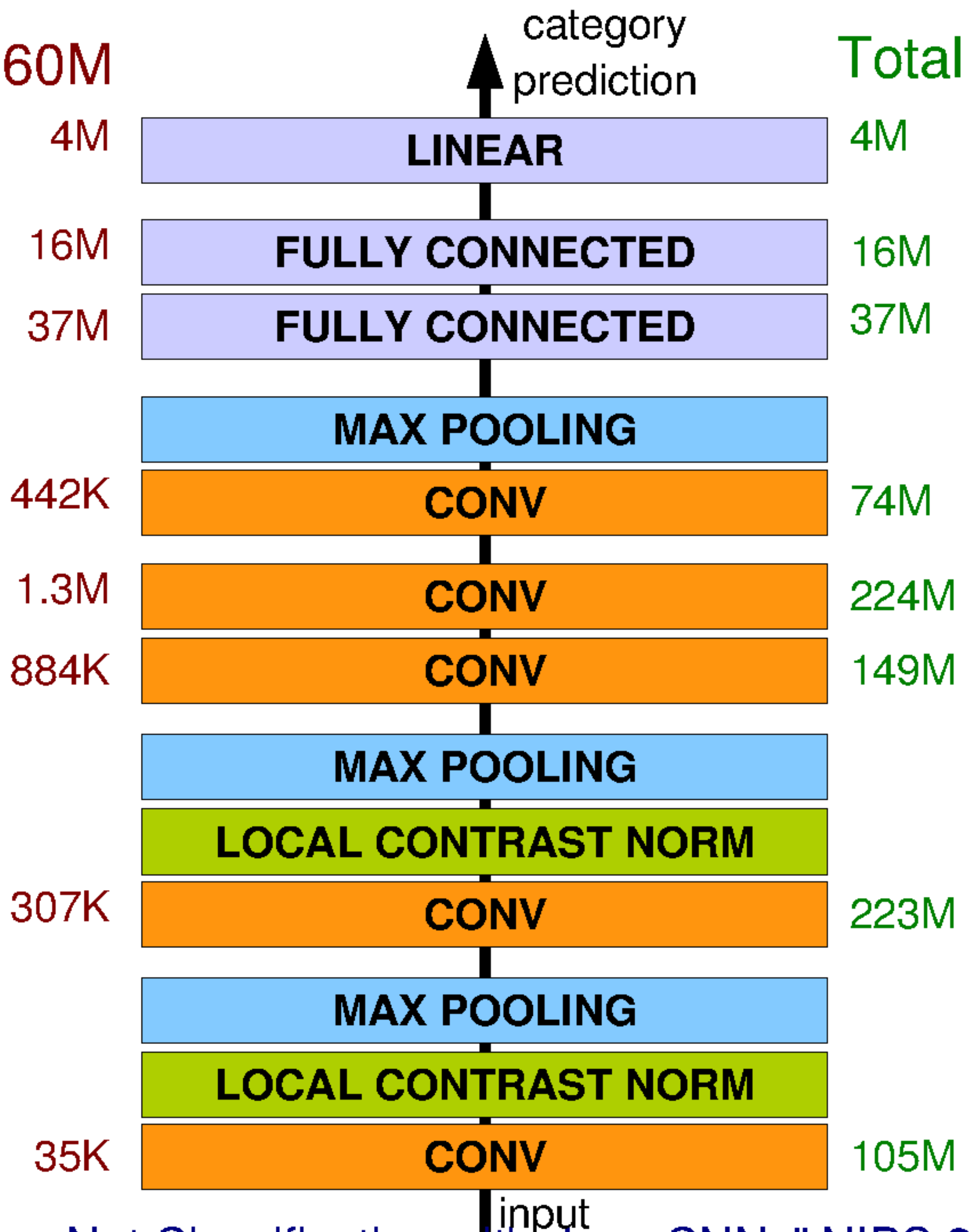
Big space of designs!

But we still don’t even know how many layers we need.

Architecture for Classification

Total nr. params: 60M

Total nr. flops: 832M



Krizhevsky et al. “ImageNet Classification with deep CNNs” NIPS 2012

WE NEED TO GO
DEEPER


NOT SURE IF
WE NEED TO GO DEEPER
imgflip.com


WE NEED TO GO
DEEPER

Beyond AlexNet

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan & Andrew Zisserman 2015

These are the pre-trained "VGG" networks that you use in project 4

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

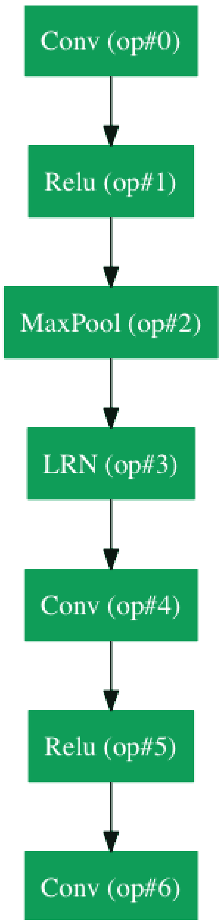
Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Table 4: **ConvNet performance at multiple test scales.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
B	256	224,256,288	28.2	9.6
C	256	224,256,288	27.7	9.2
	384	352,384,416	27.8	9.2
	[256; 512]	256,384,512	26.3	8.2
D	256	224,256,288	26.6	8.6
	384	352,384,416	26.5	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**
E	256	224,256,288	26.9	8.7
	384	352,384,416	26.7	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**

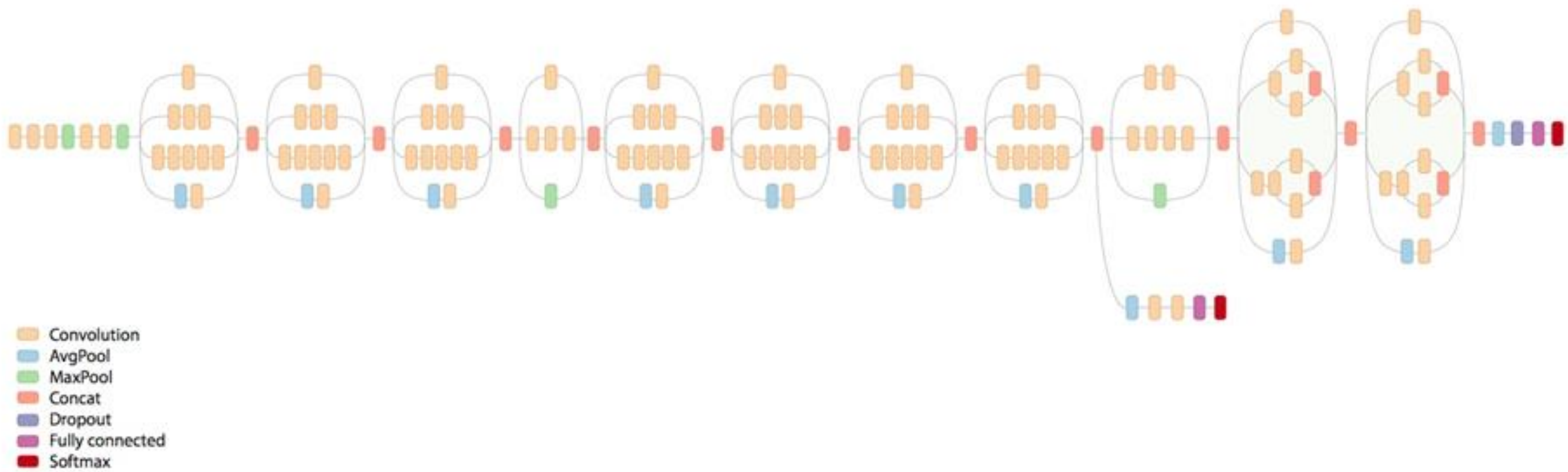
Google LeNet (2014)



22 layers

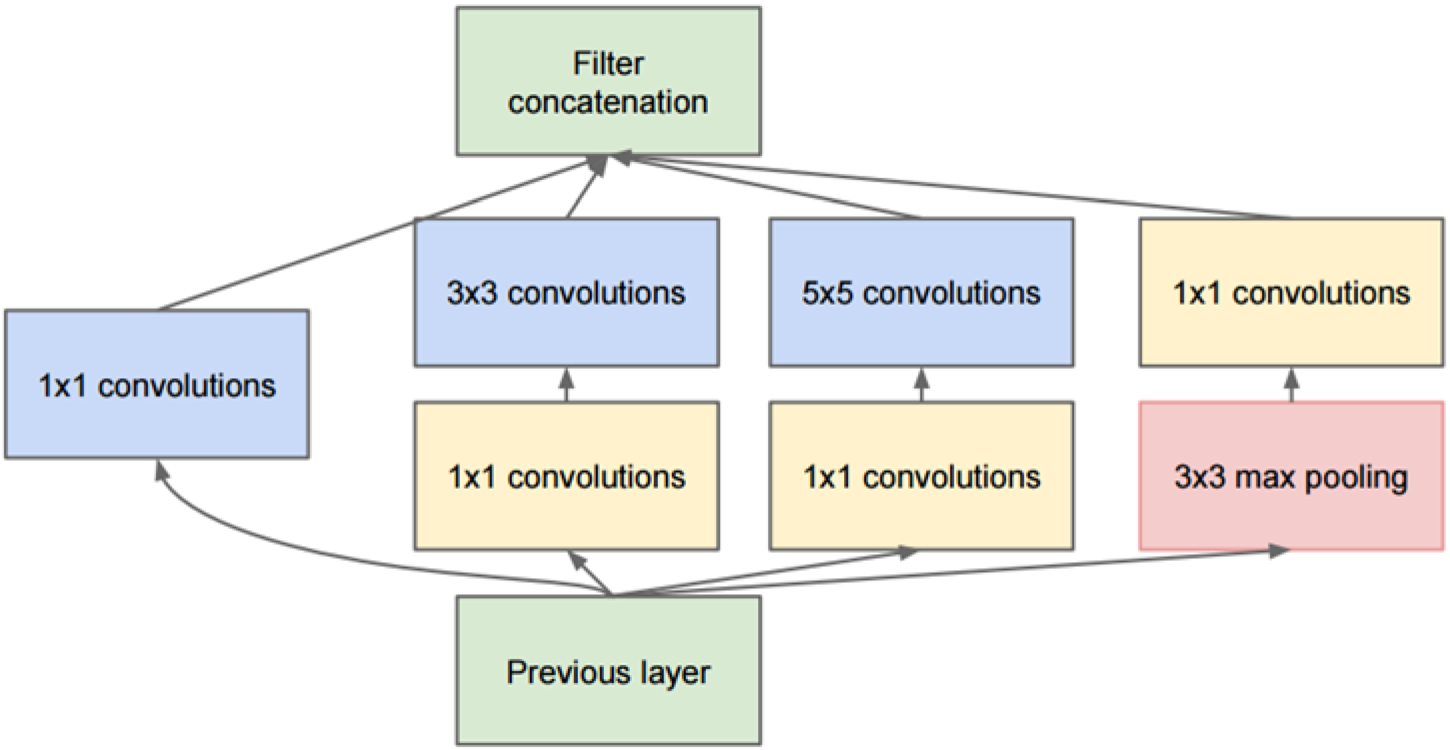
6.67% error
ImageNet top 5

Inception!



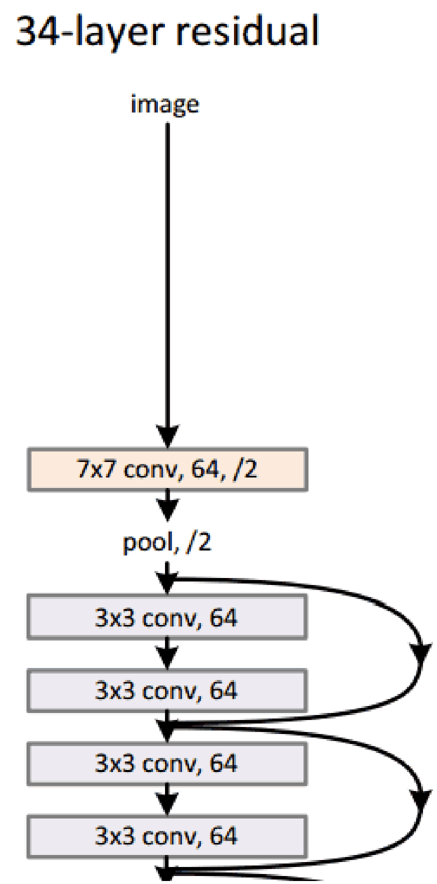
Another view of GoogLeNet's architecture.

Parallel layers



Full Inception module

ResNet (He et al., 2015)

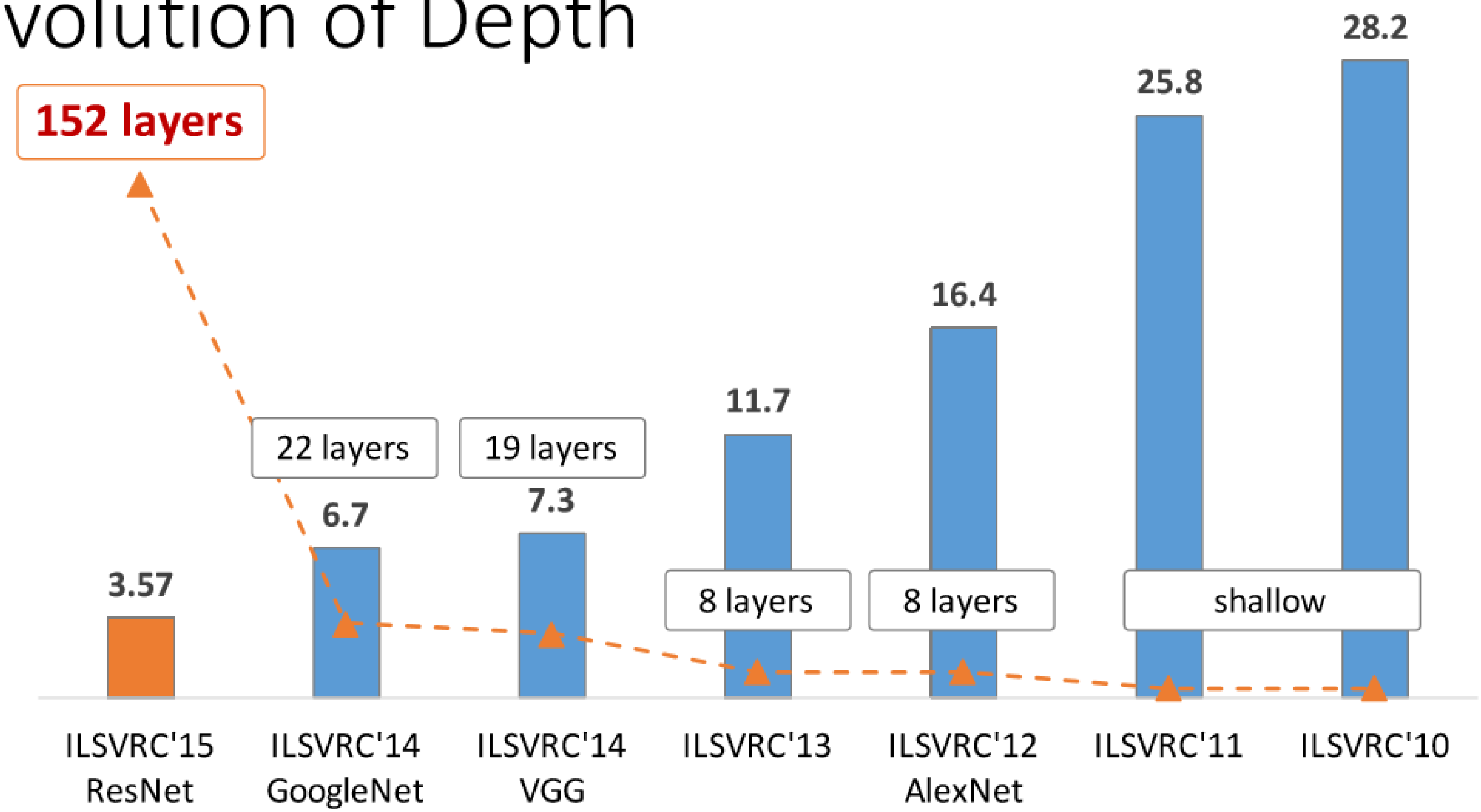


ResNet won ILSVRC 2015 with a top-5 error rate of 3.6%

Depending on their skill and expertise, humans generally hover around a 5-10% error.

~~Superhuman performance!~~
But the task is arguably not well defined.

Revolution of Depth



ImageNet Classification top-5 error (%)

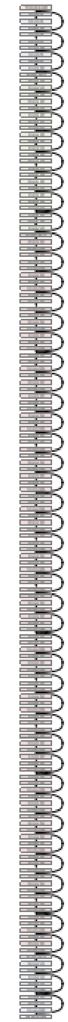
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

VGG, 19 layers
(ILSVRC 2014)

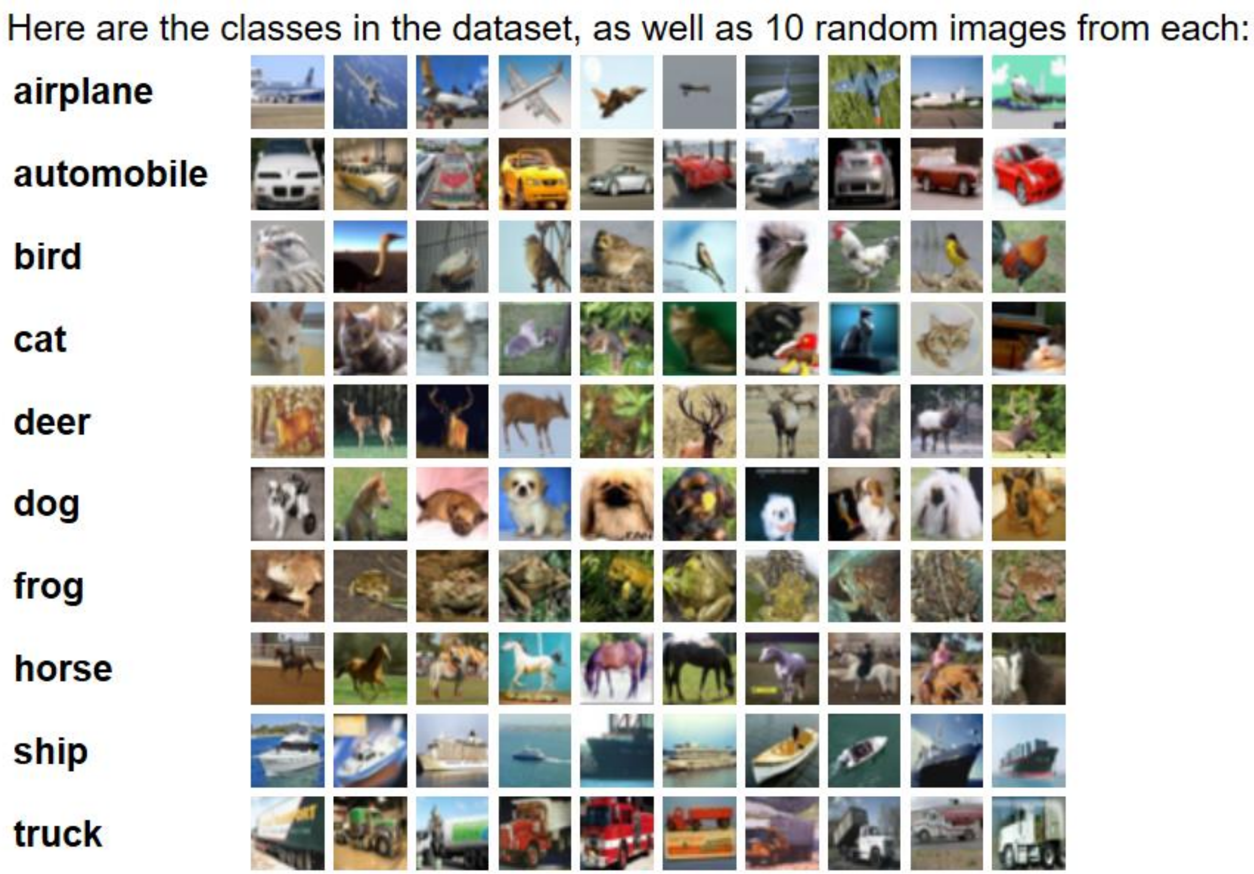
ResNet, 152 layers
(ILSVRC 2015)



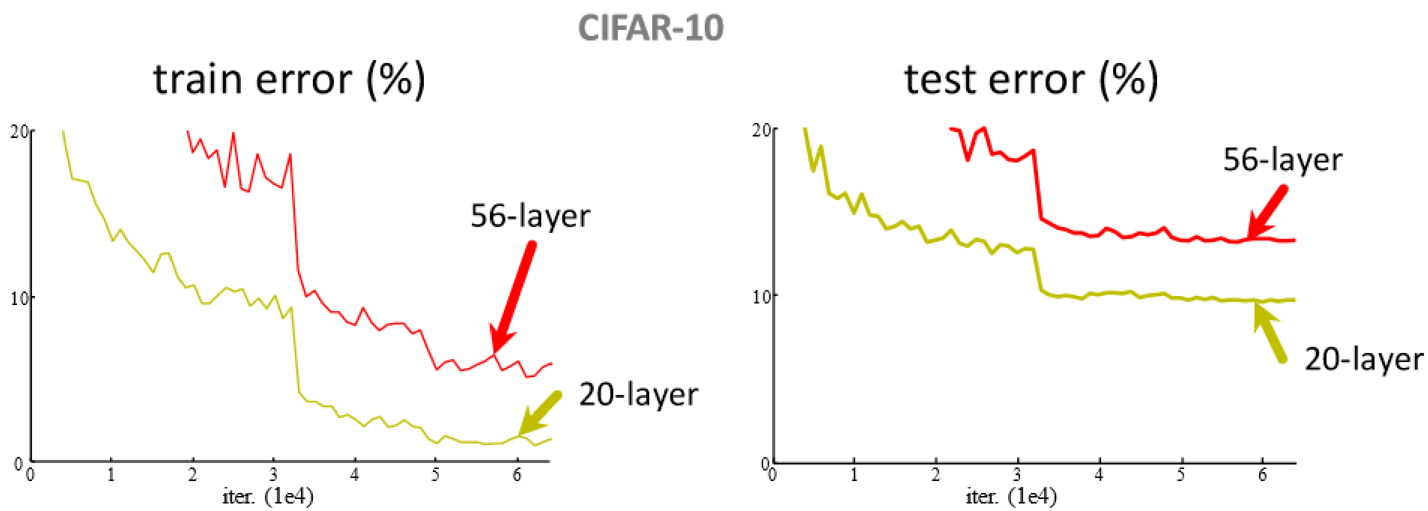
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

CIFAR-10

- 60,000 32x32 color images, 10 classes



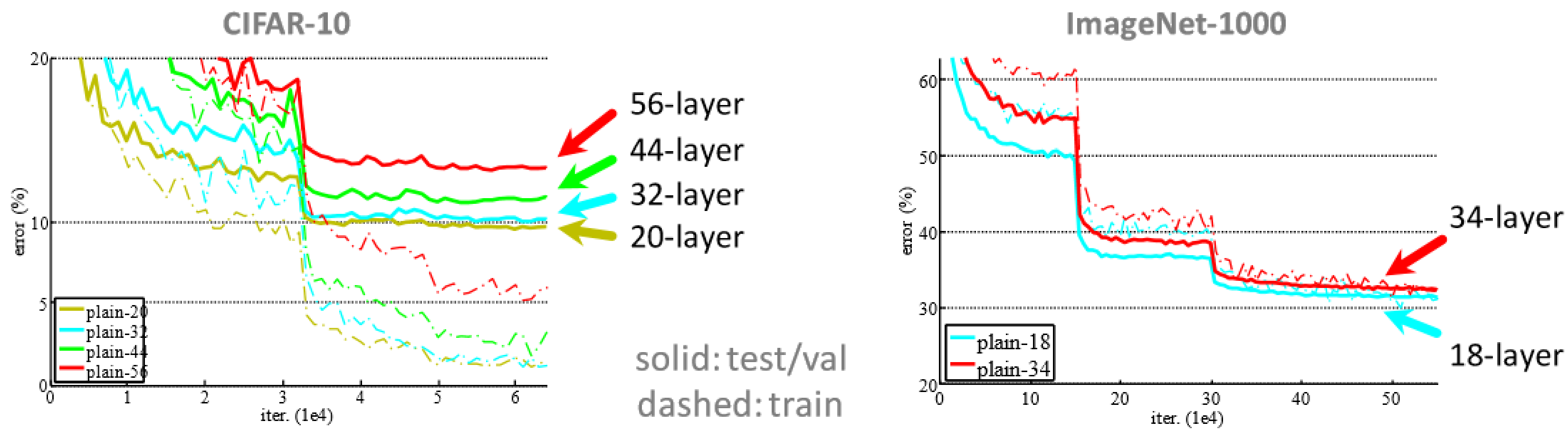
Simply stacking layers?



- *Plain* nets: stacking 3x3 conv layers…
- 56-layer net has **higher training error** and test error than 20-layer net

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Simply stacking layers?



- "Overly deep" plain nets have **higher training error**
- A general phenomenon, observed in many datasets

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Vanishing/exploding gradient problem

Backpropagation:

- Compute gradient update for every neuron which was involved in the output across layers

Involves chaining partial derivates over many layers!

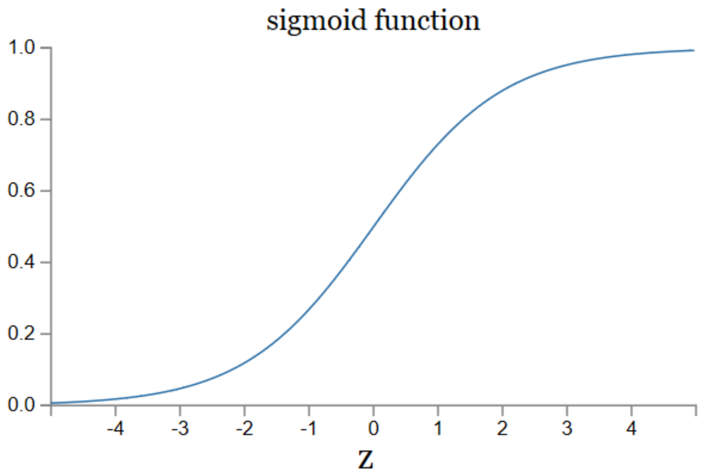
- If derivative < 1, gradient gets smaller and smaller as we go deeper and deeper -> *vanishing gradients!*
- If derivative > 1, gradient gets larger and larger as we go deeper and deeper -> *exploding gradients!*

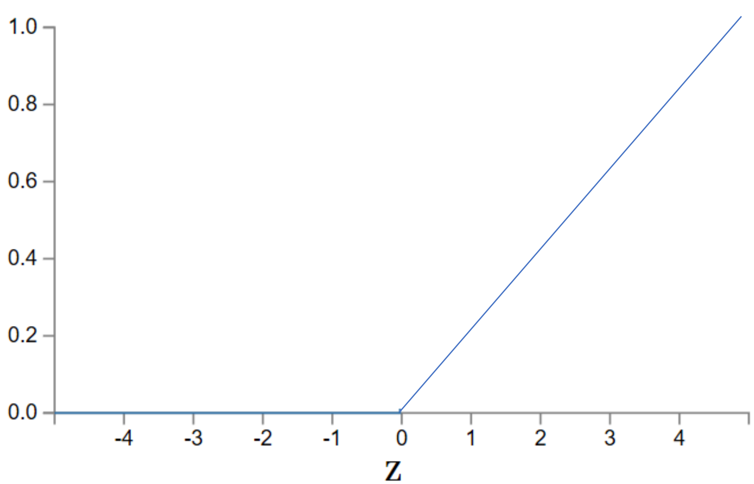
Vanishing/exploding gradient re: activation

- If derivative < 1, gradient gets smaller and smaller as we go deeper and deeper -> *vanishing gradients!*
- If derivative > 1, gradient gets larger and larger as we go deeper and deeper -> *exploding gradients!*

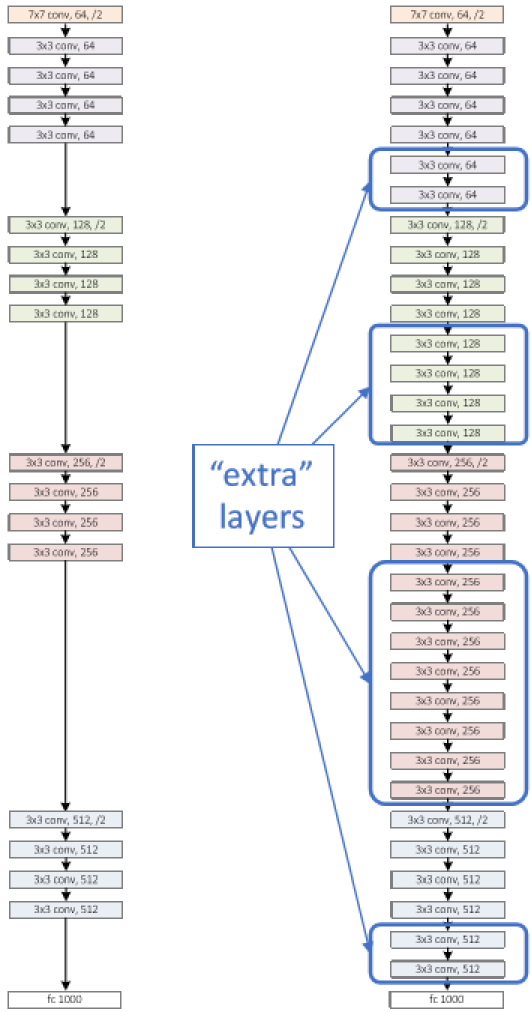
Sigmoid $\sigma(z) \equiv \frac{1}{1+e^{-z}}.$

ReLU $\sigma(z) = \max(0, x)$





a shallower model (18 layers)



a deeper counterpart (34 layers)

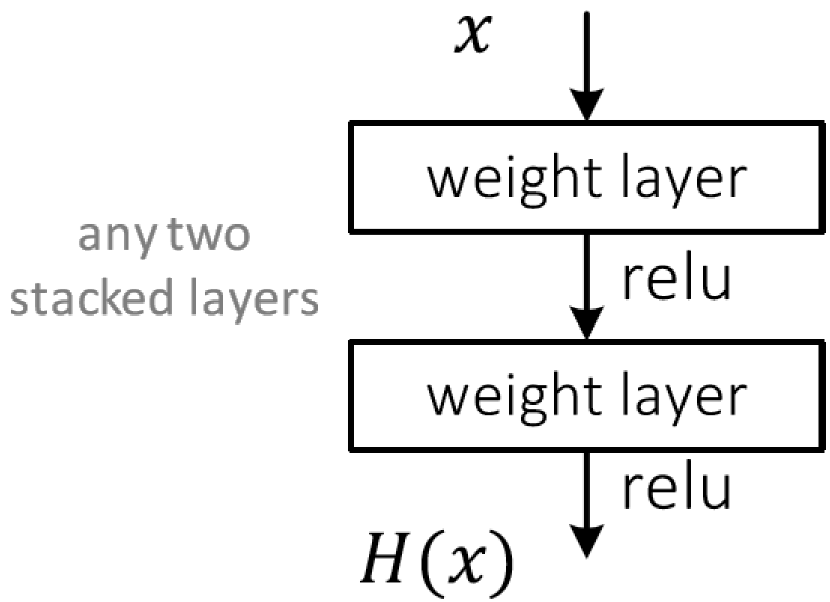
- Richer solution space
- A deeper model should not have **higher training error**

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

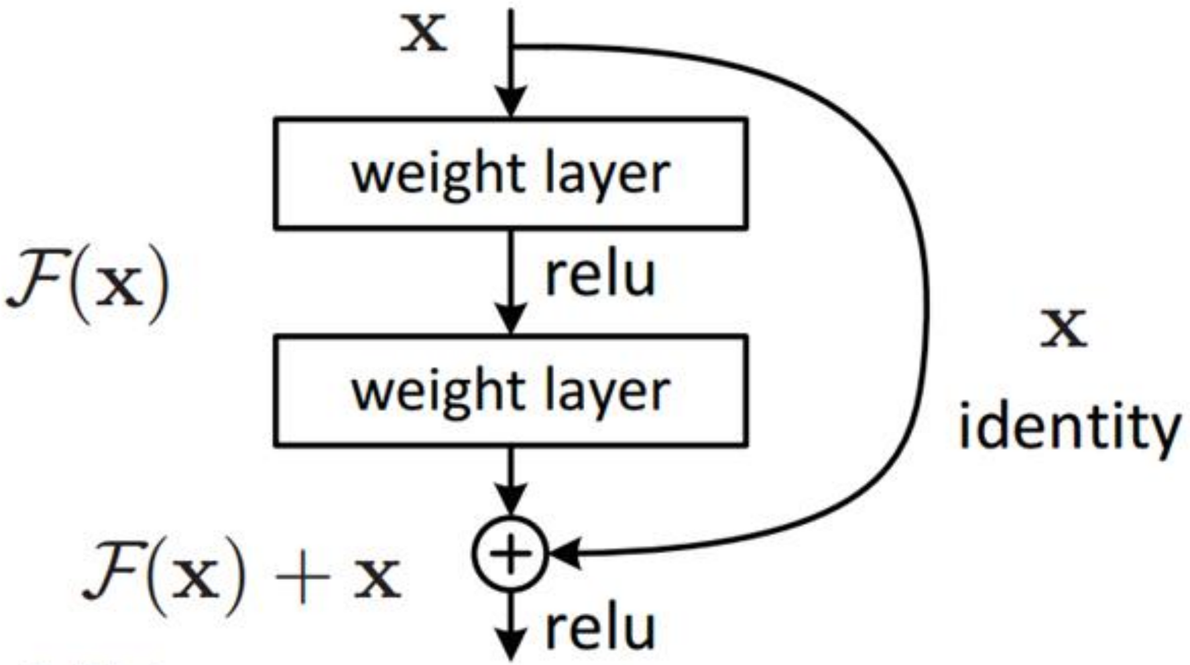
Regular net

$H(x)$ is any desired mapping,

hope the 2 weight layers fit $H(x)$



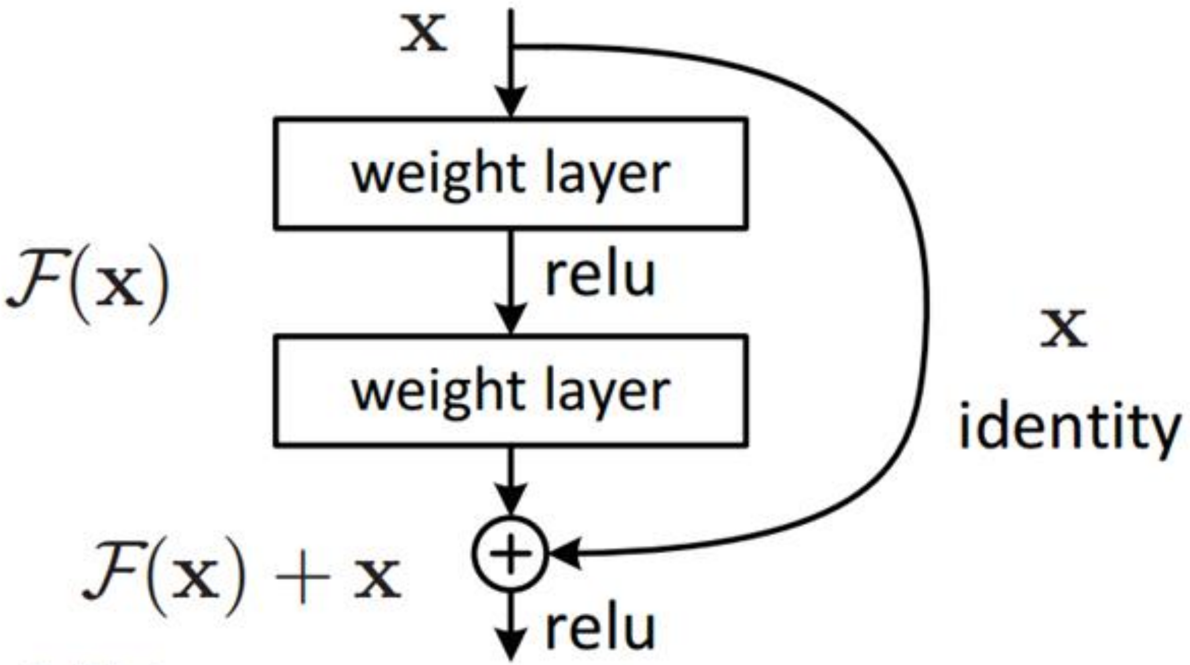
Residual Unit



A residual block

Residual Unit

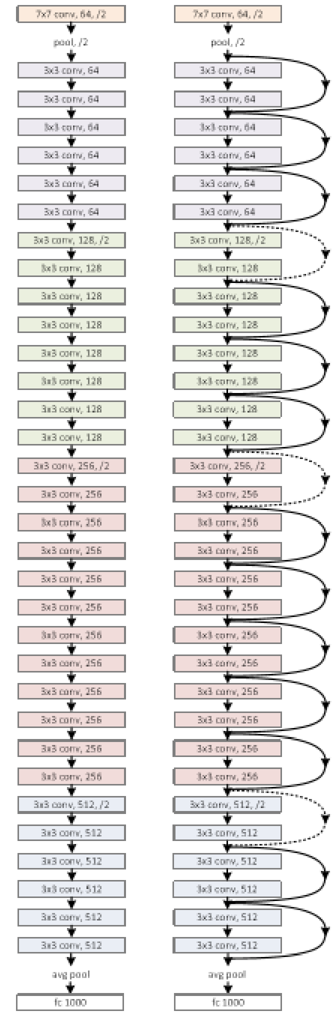
The inputs of a lower layer is made available to a node in a higher layer.



A residual block

Network “Design”

plain net

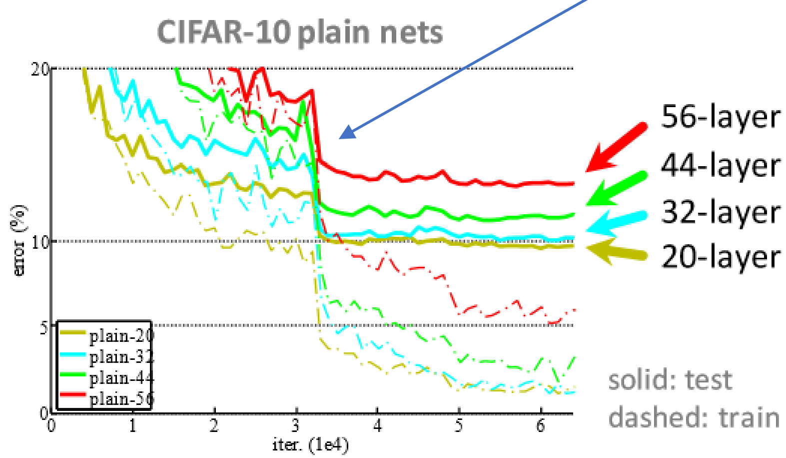


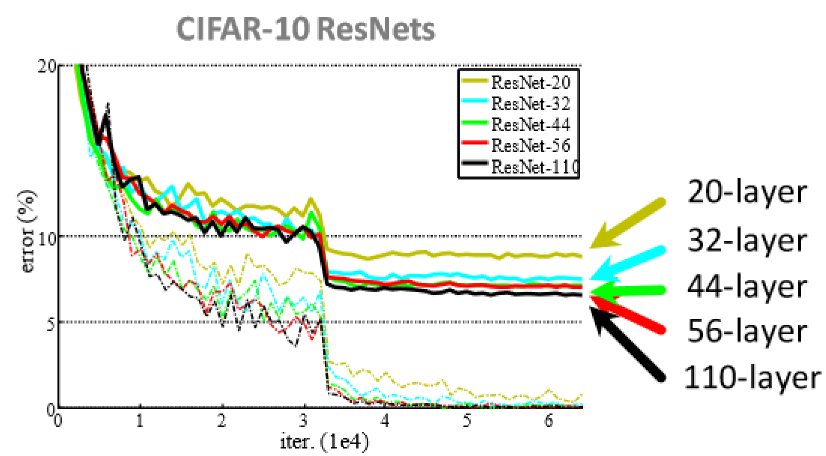
ResNet

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

Why so steep?

CIFAR-10 experiments

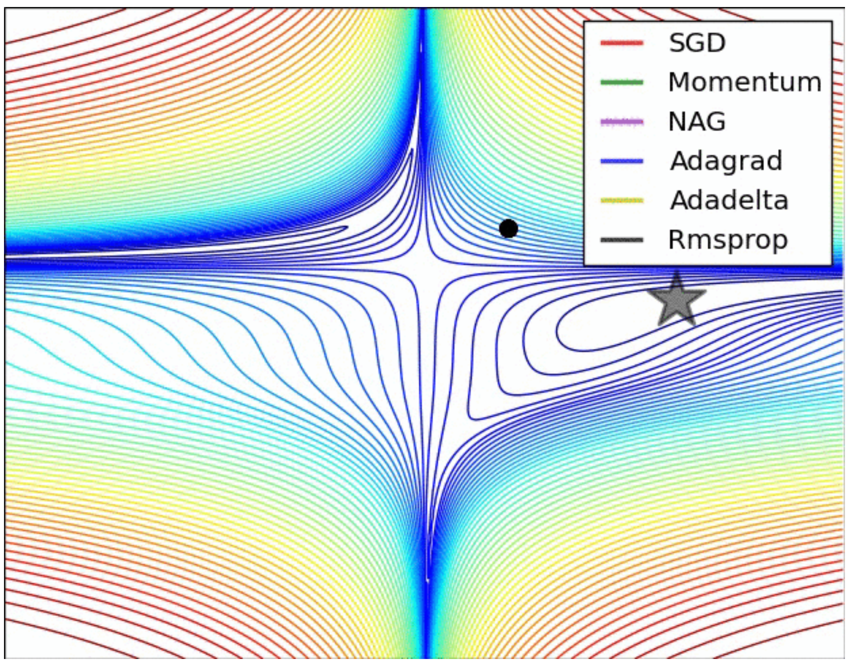


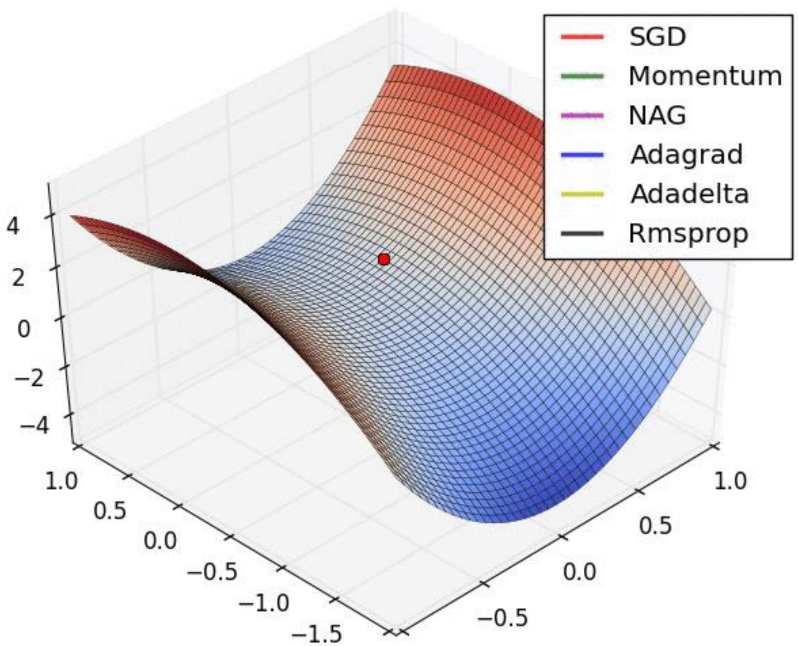


- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

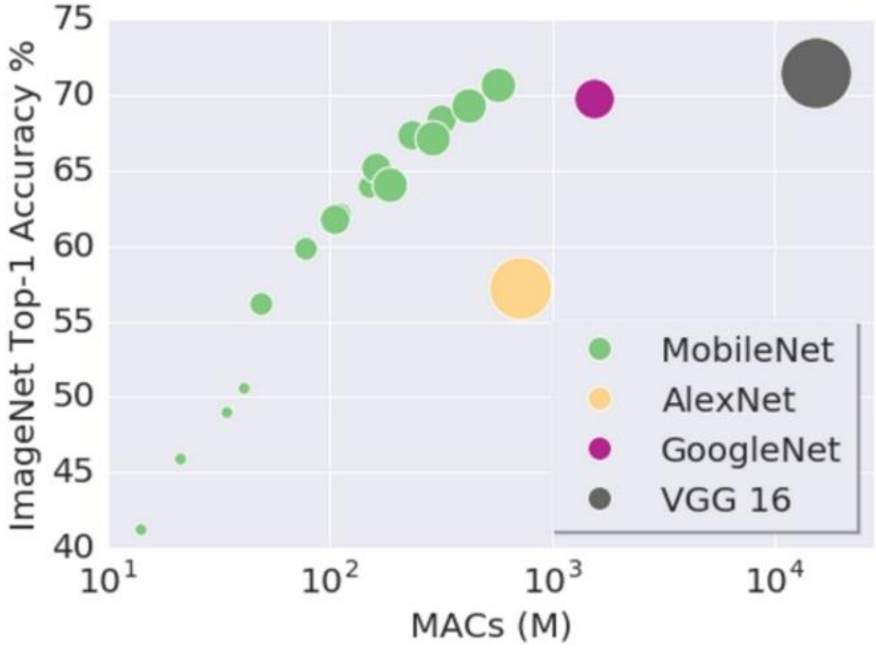
Flat regions in energy landscape





James, do we *have* to go deeper?

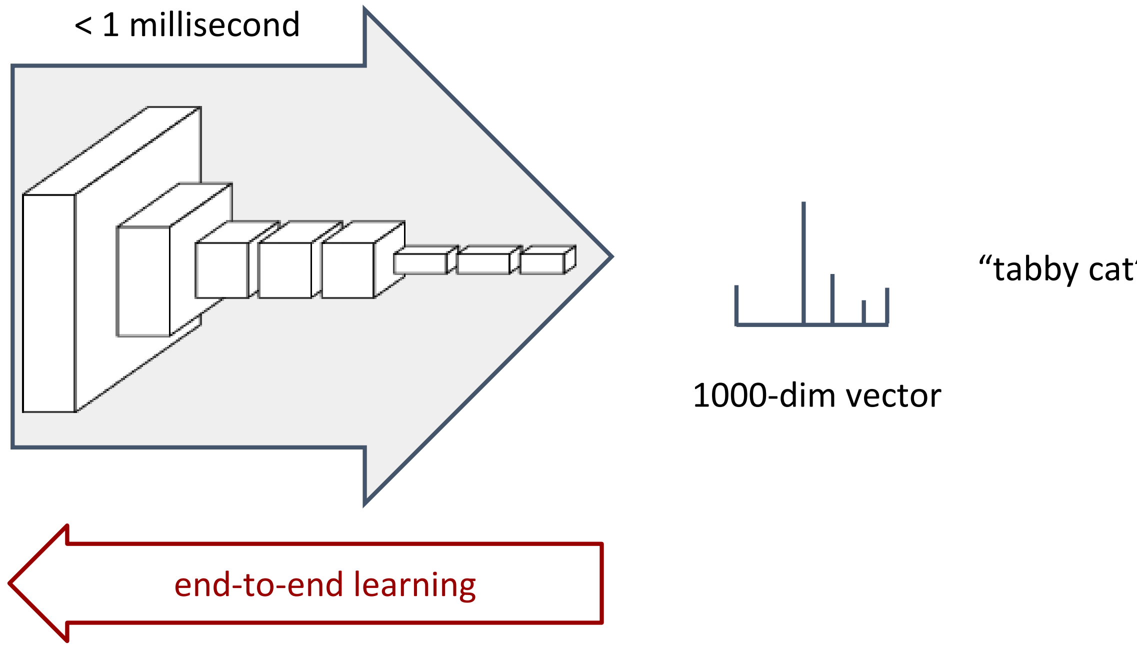
Compute vs. parameters / multiply-adds



Hmm…efficient nets…
might be useful for final project ???

https://www.infoq.com/news/2017/06/google-mobilenets-tensorflow
https://arxiv.org/abs/1704.04861

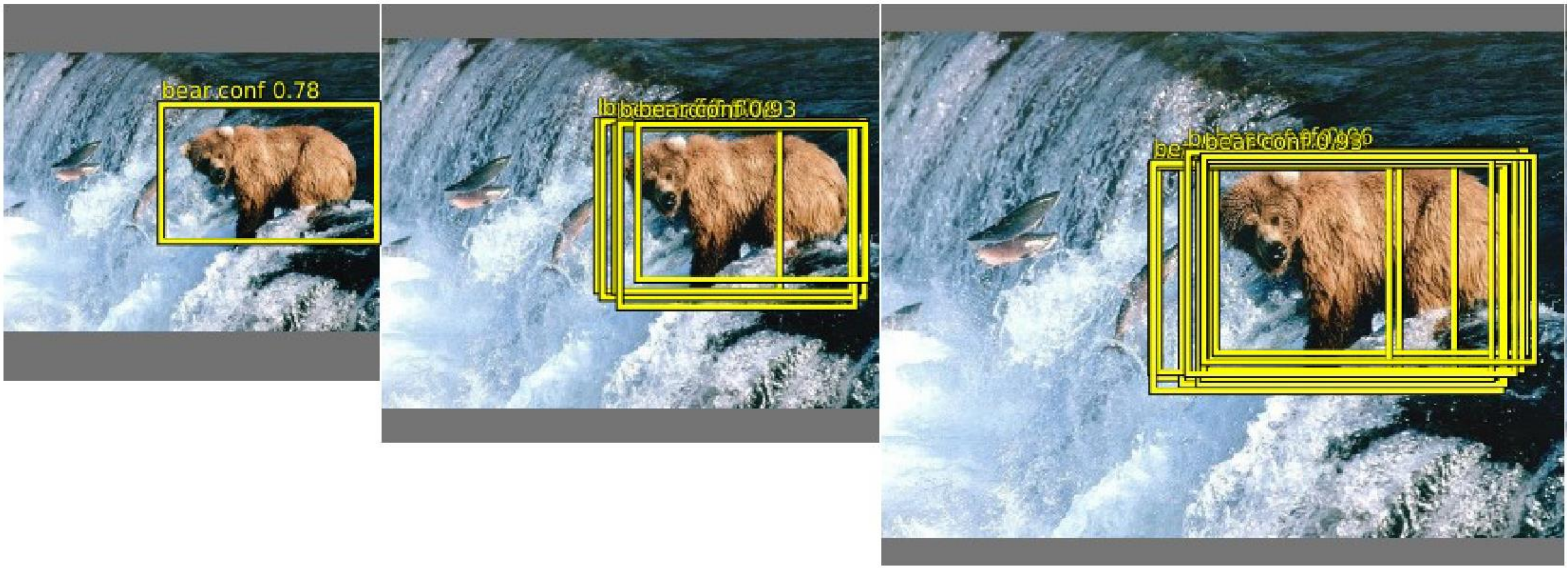
ConvNets perform classification



[Slides from Long, Shelhamer, and Darrell]

CONV NETS: EXAMPLES

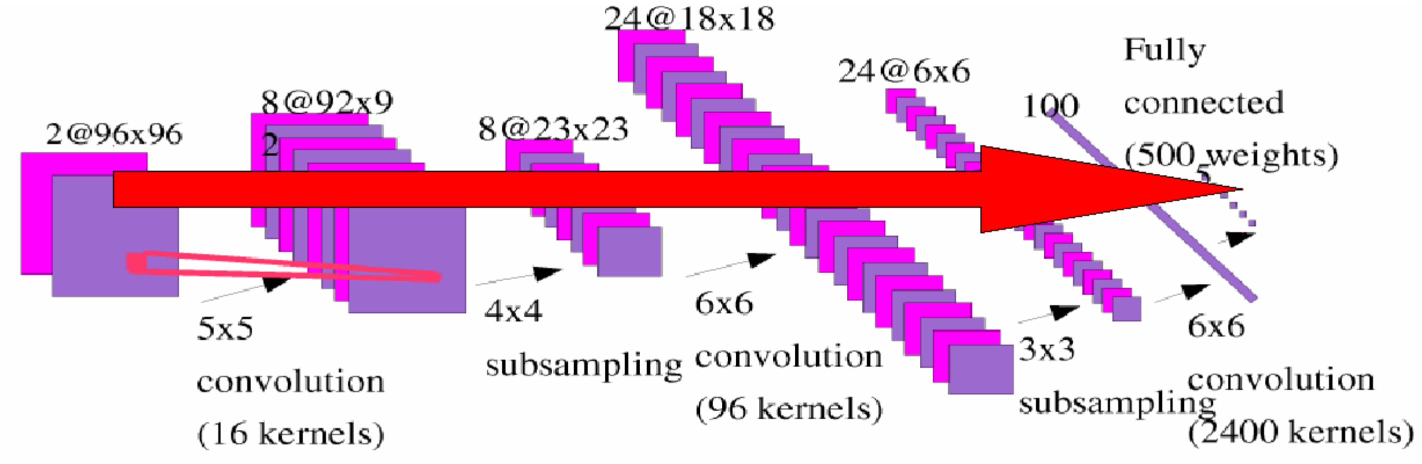
- **Object detection**



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

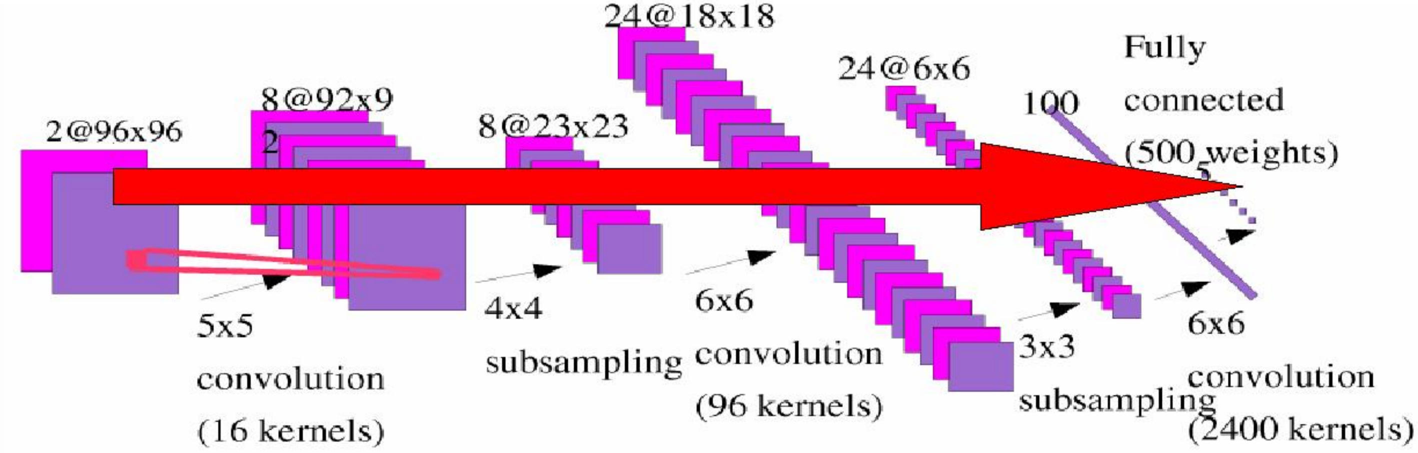
ConvNets: Test

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

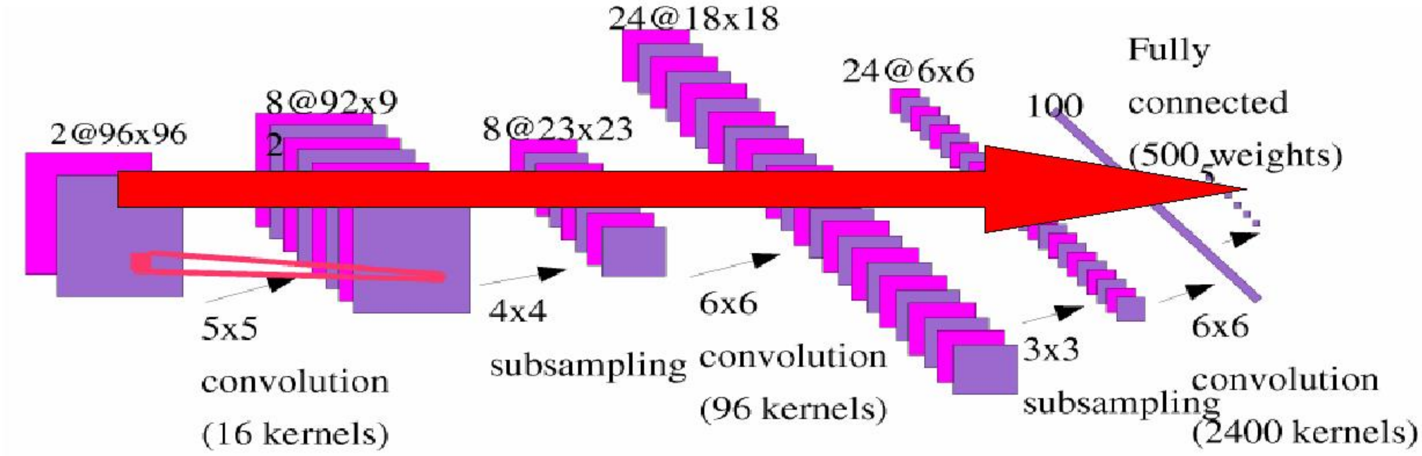
ConvNet

Traditional methods use inefficient sliding windows.

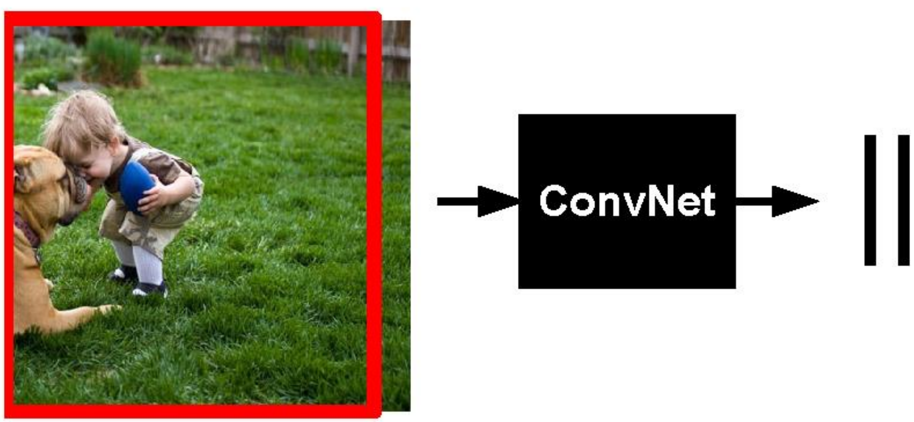
Ranzato

ConvNets: Test

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



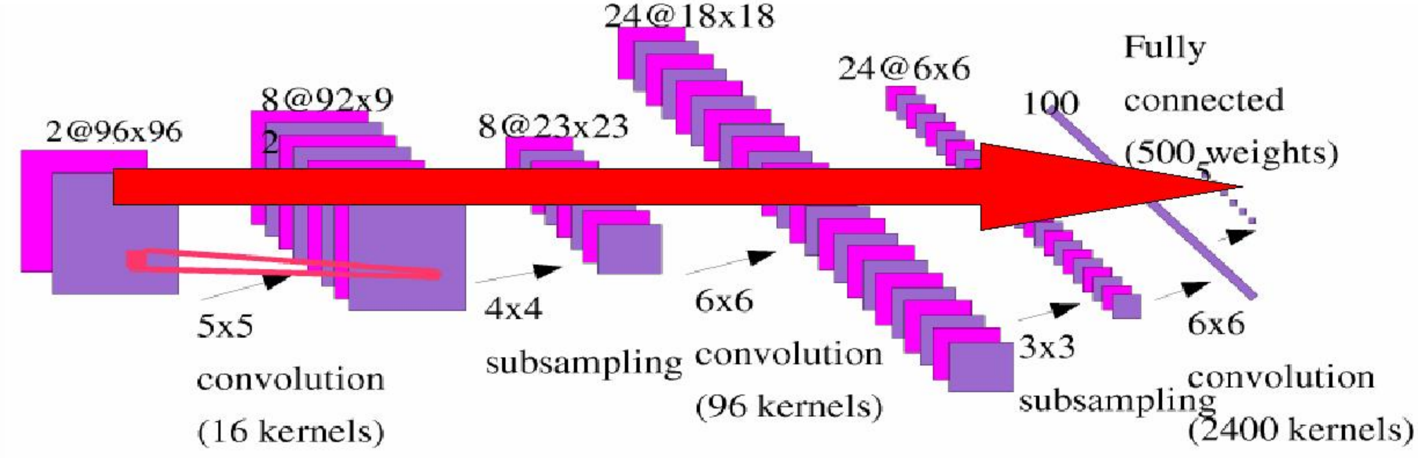
Naturally, convnet can process larger images



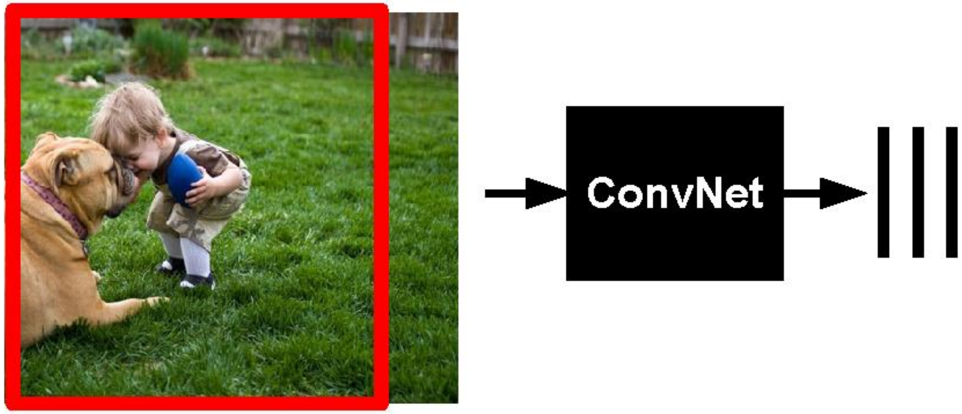
Traditional methods use inefficient sliding windows.

ConvNets: Test

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



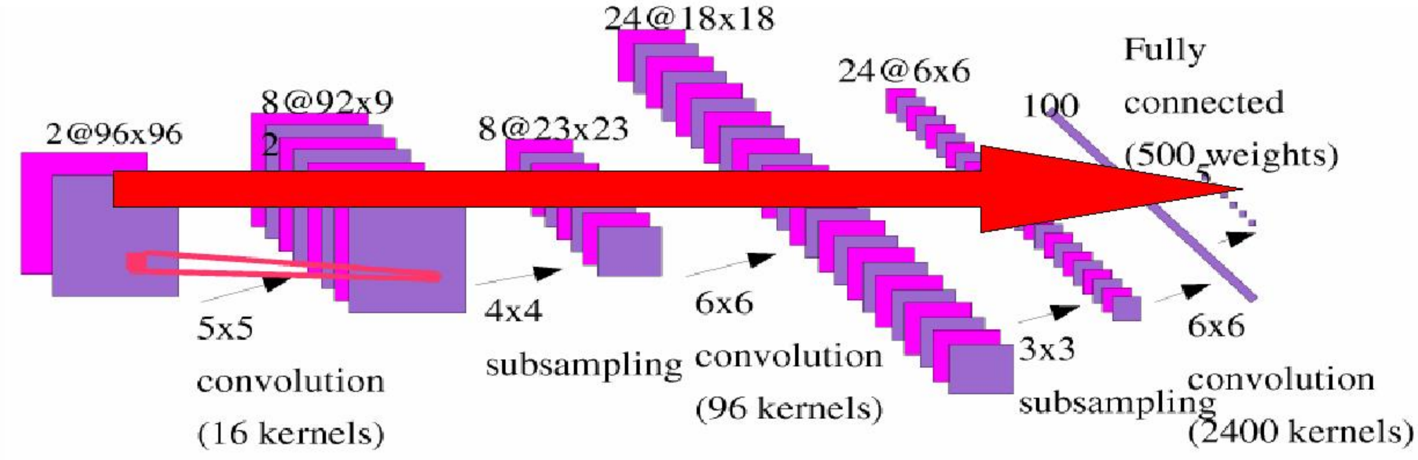
Naturally, convnet can process larger images



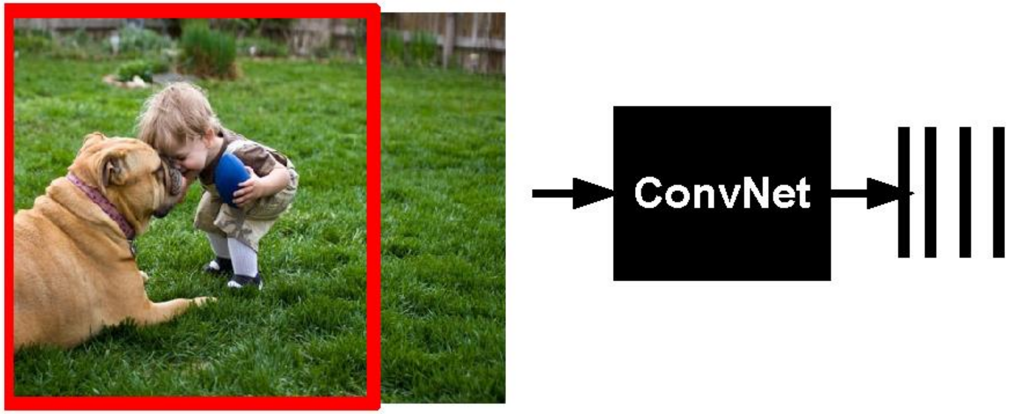
Traditional methods use inefficient sliding windows.

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

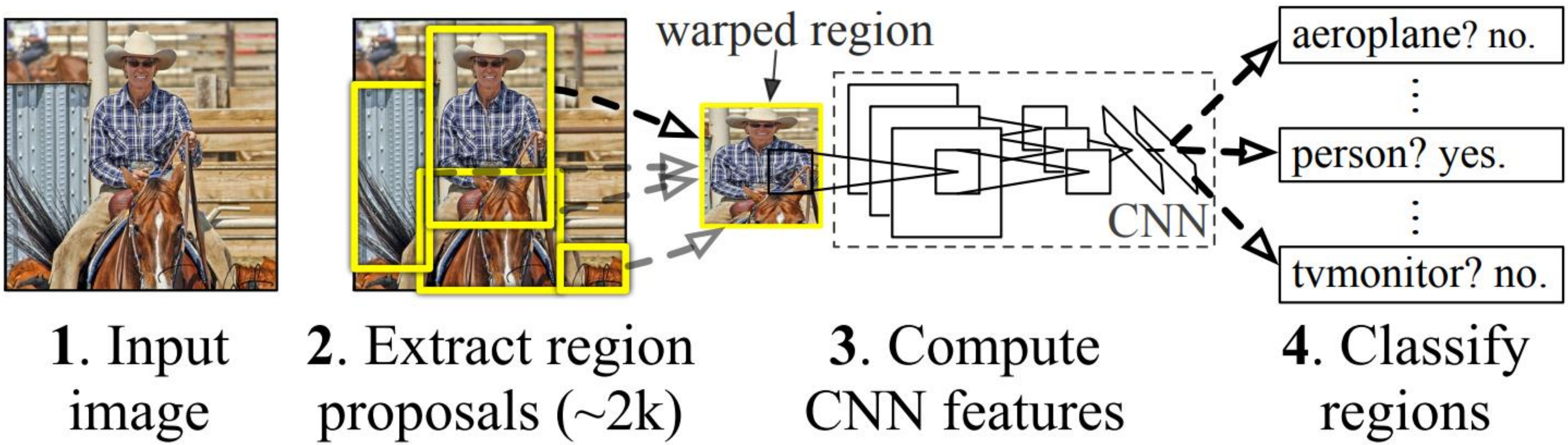


Figure: Girshick et al.

Stage 2: Efficient region proposals?

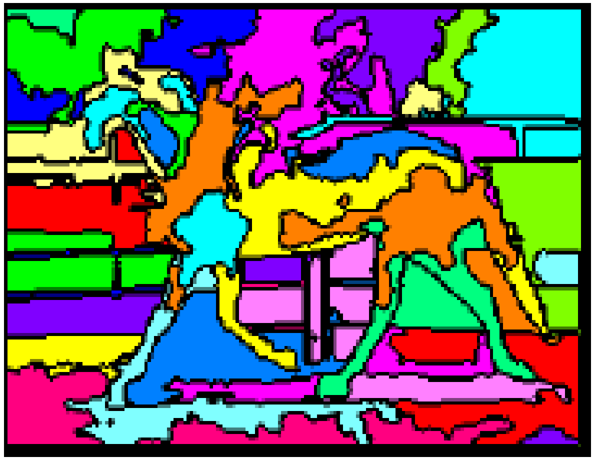
- Brute force on 1000x1000 = 250 billion rectangles
  - Testing the CNN over each one is too expensive

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

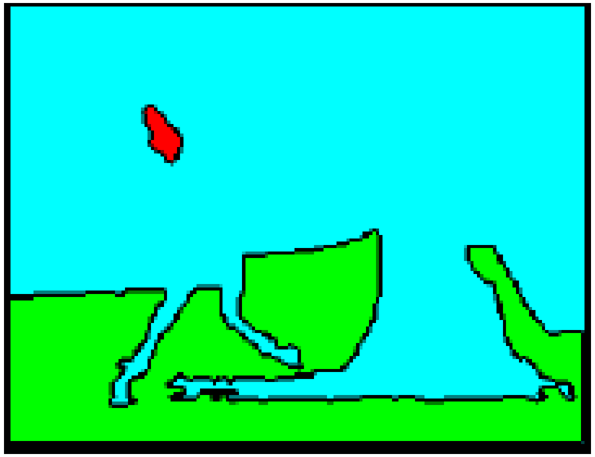
Uijlings et al., 2012, Selection Search

Thanks to Song Cao

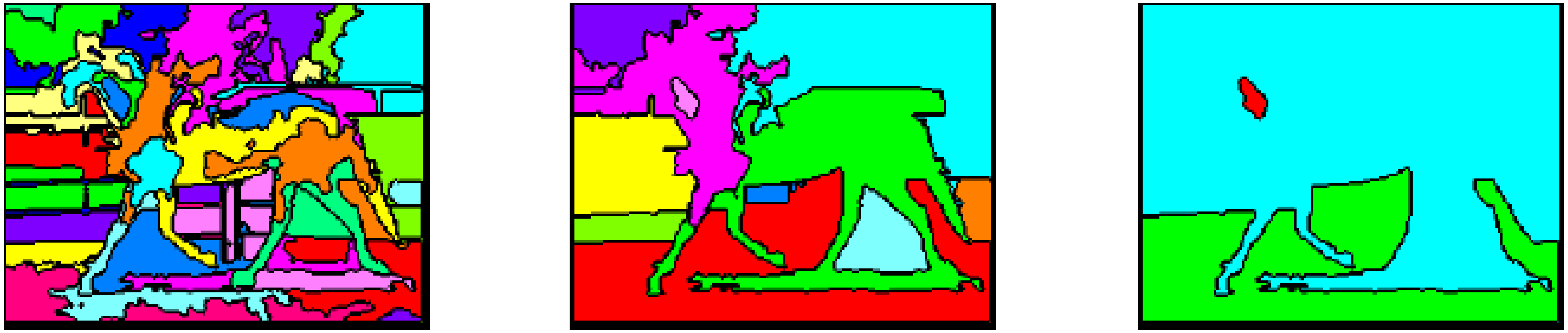
Remember clustering for segmentation?



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)

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

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

(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

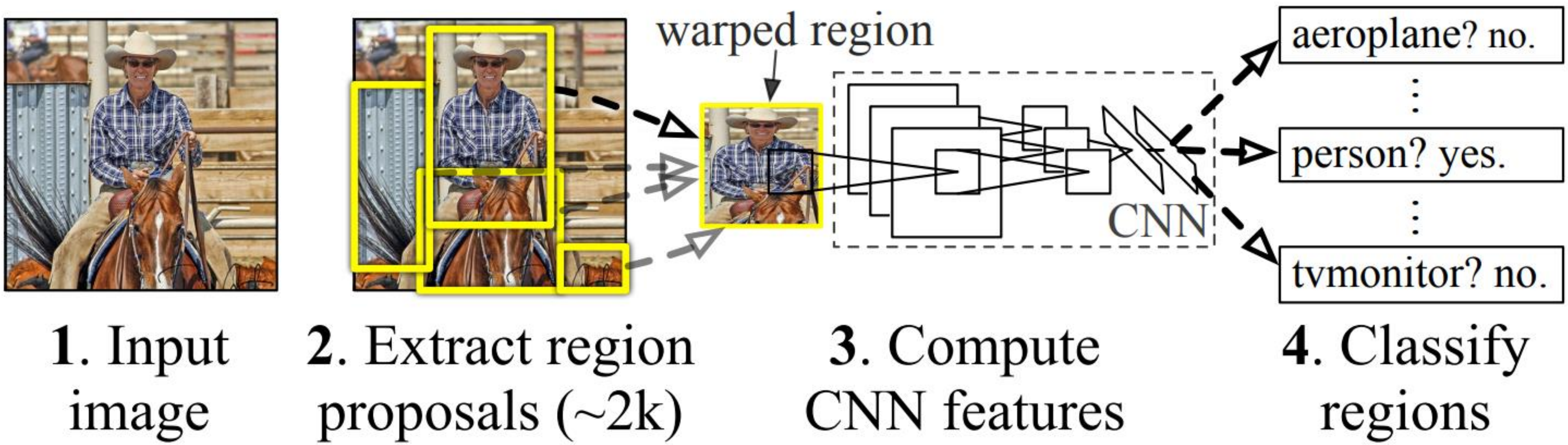
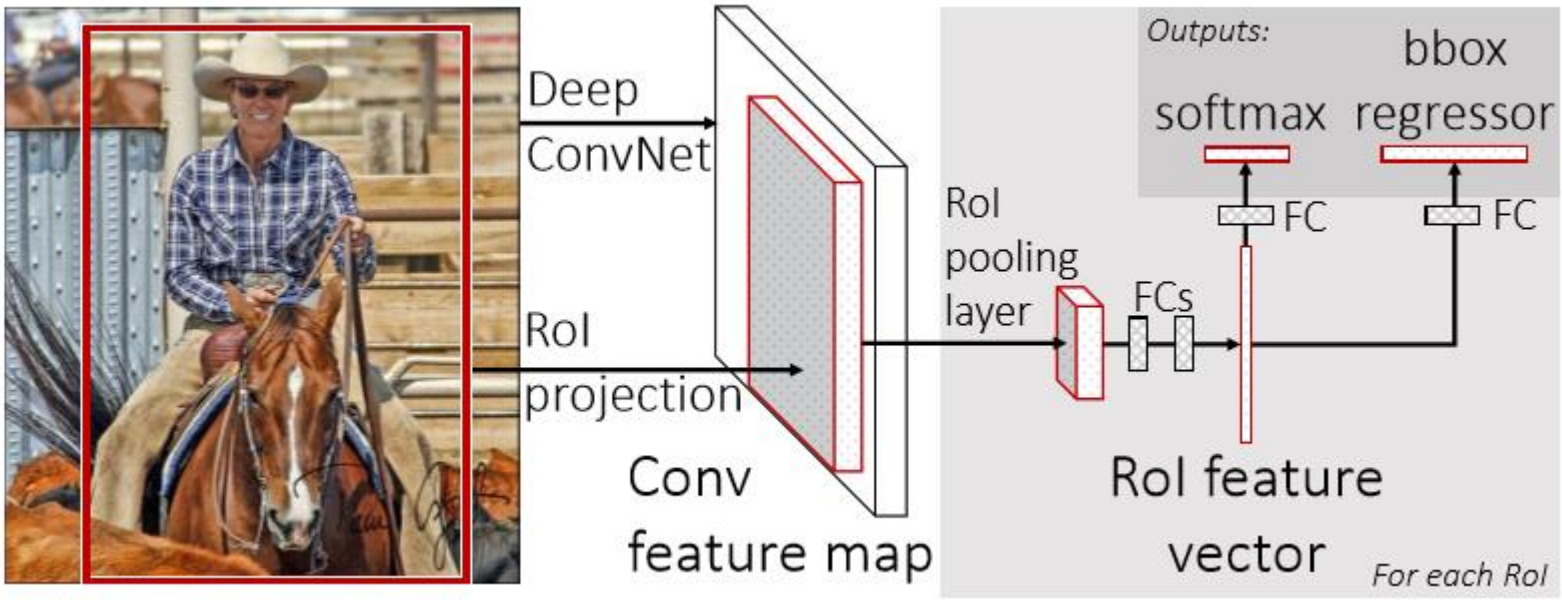


Figure: Girshick et al.

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

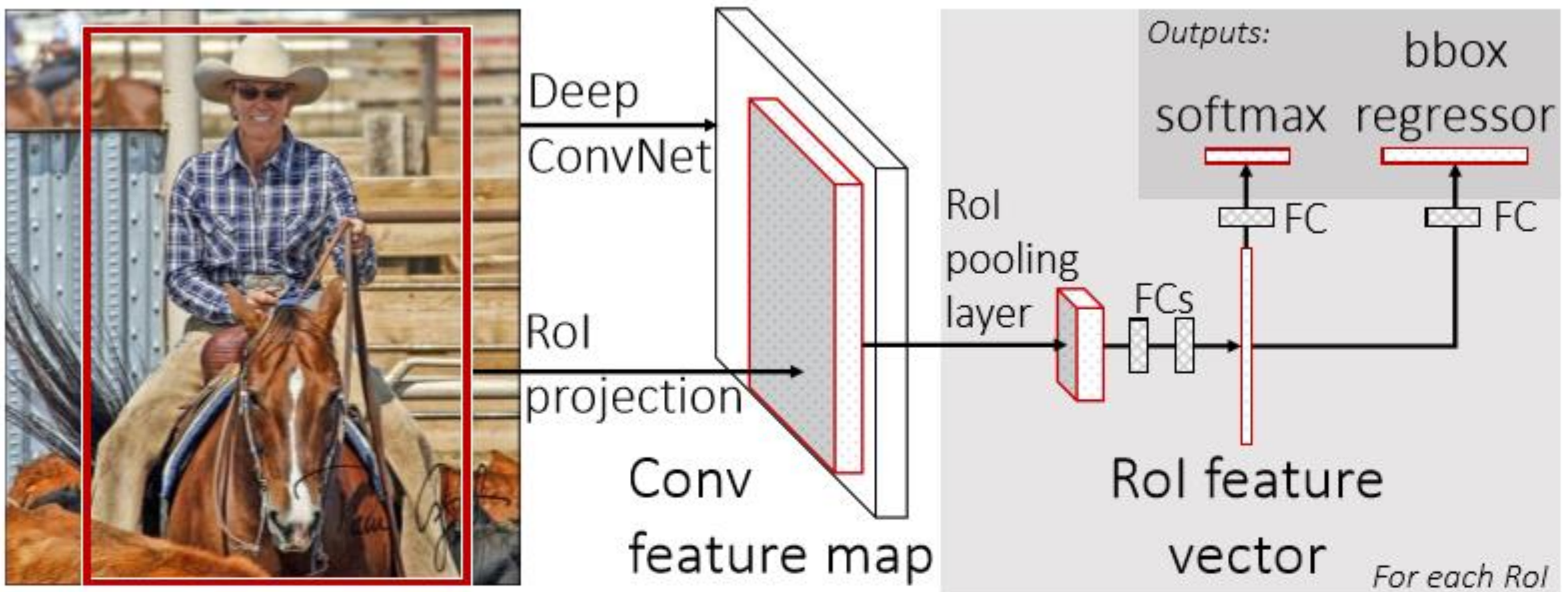
Multi-task loss



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.

Martian lava field, NASA, Wikipedia

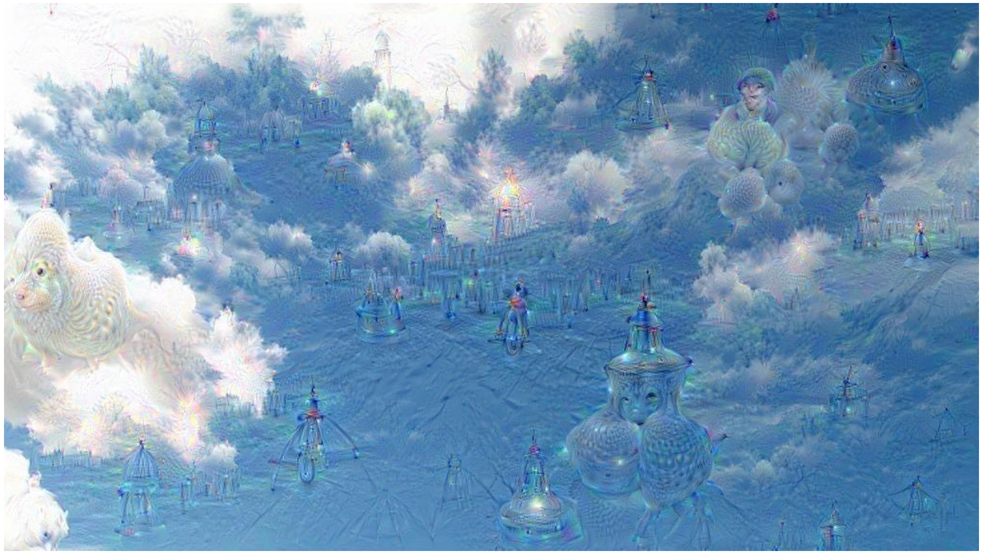
Old Man of the Mountain, Franconia, New Hampshire

Pareidolia

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

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

Pareidolia

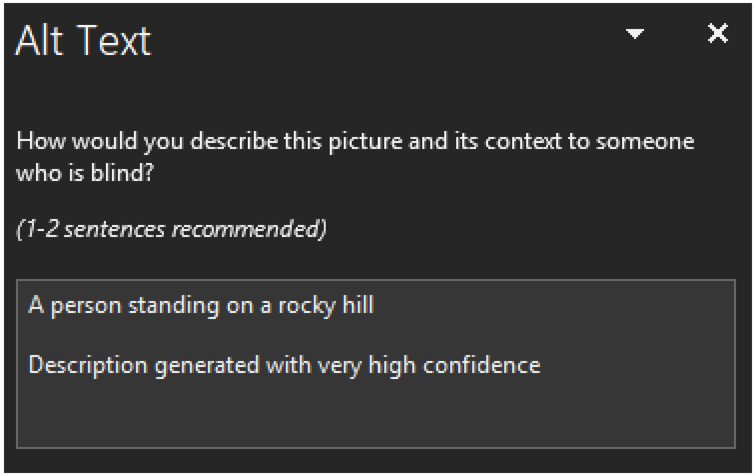


Seeing things which aren't really there…

DeepDream as reinforcement pareidolia

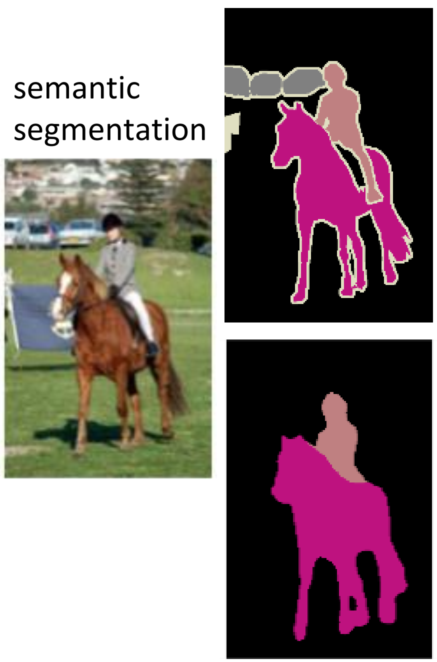
Powerpoint Alt-text Generator

Vision-based caption generator

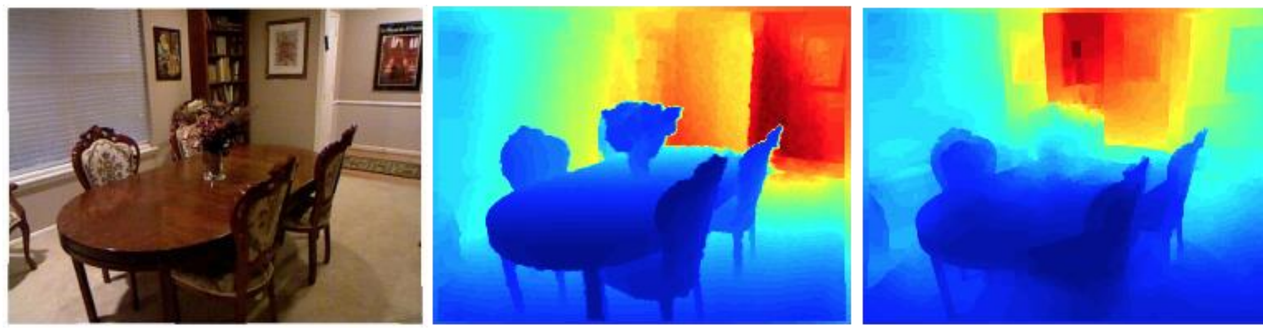


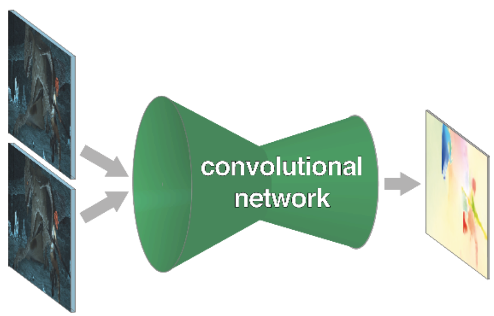
What if we want pixels out?

semantic segmentation



monocular depth estimation Eigen & Fergus 2015





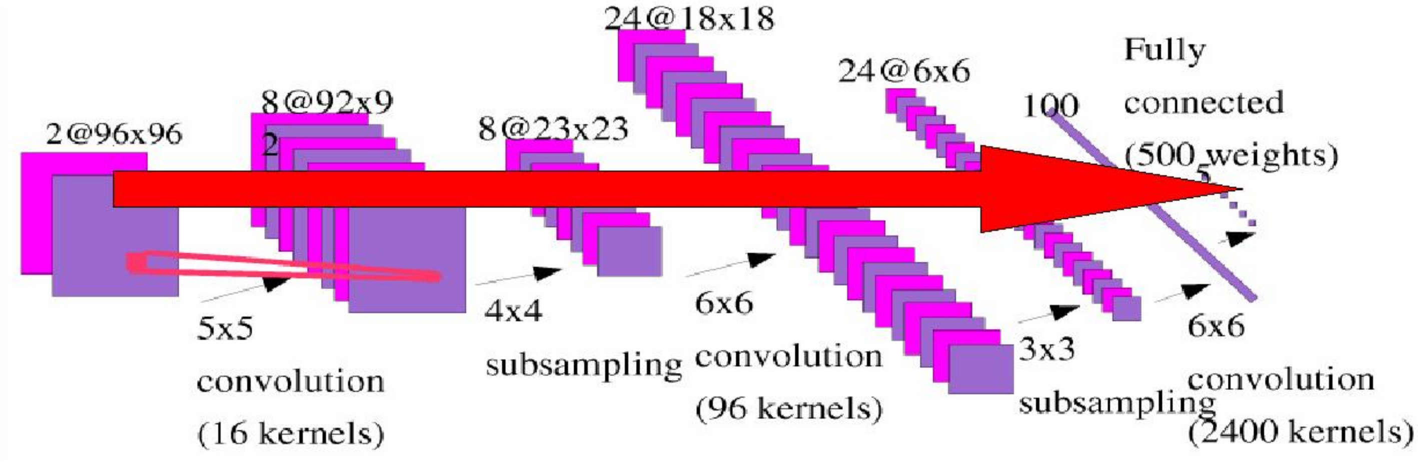
optical flow Fischer et al. 2015

boundary prediction Xie & Tu 2015

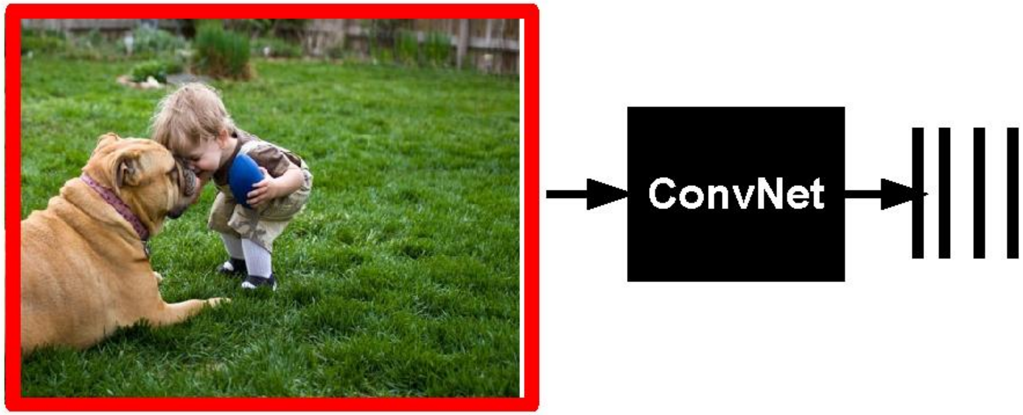
[Long et al.]

ConvNets: Test

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



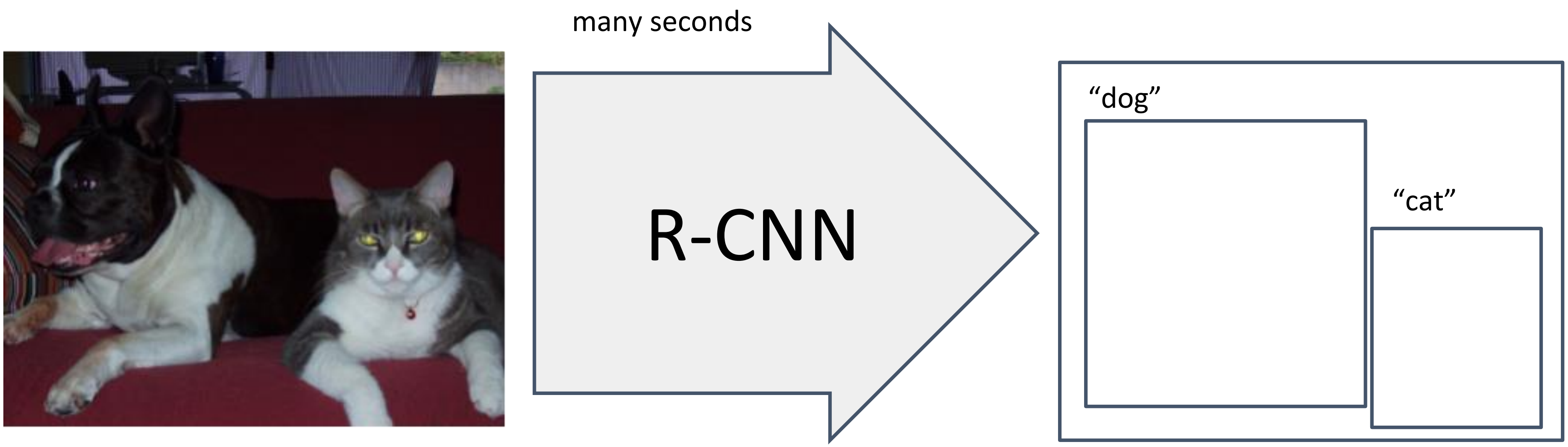
Naturally, convnet can process larger images at little cost.



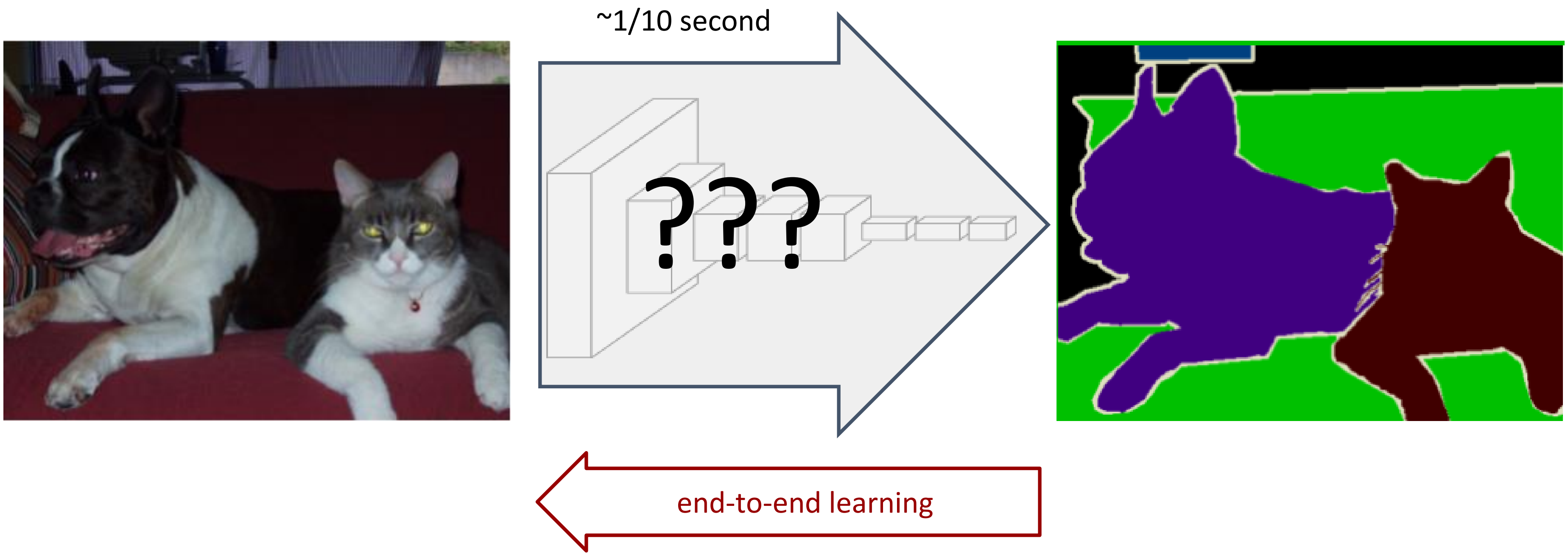
ConvNet: unrolls convolutions over bigger images and produces outputs at several locations.

Ranzato

R-CNN does detection

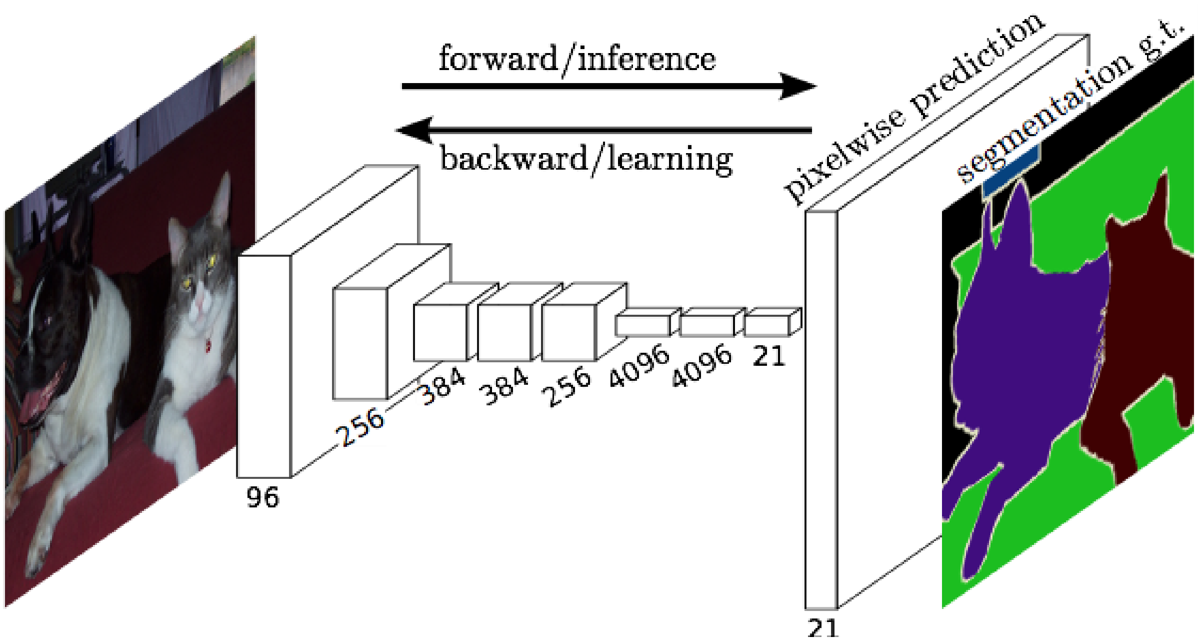


[Long et al.]



[Long et al.]

Fully Convolutional Networks for Semantic Segmentation

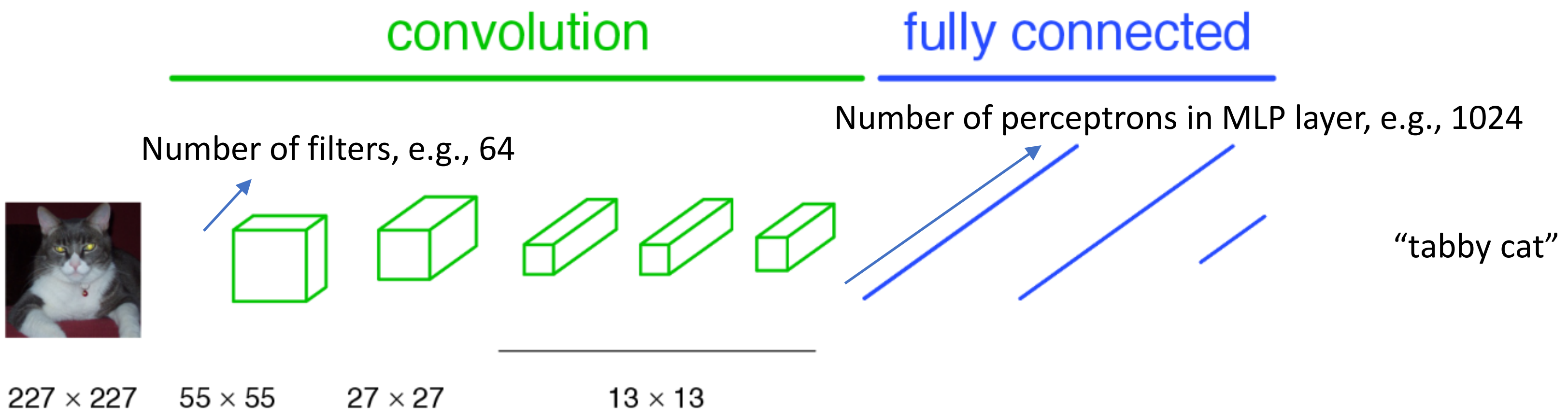


Jonathan Long* Evan Shelhamer* Trevor Darrell
UC Berkeley

[CVPR 2015]

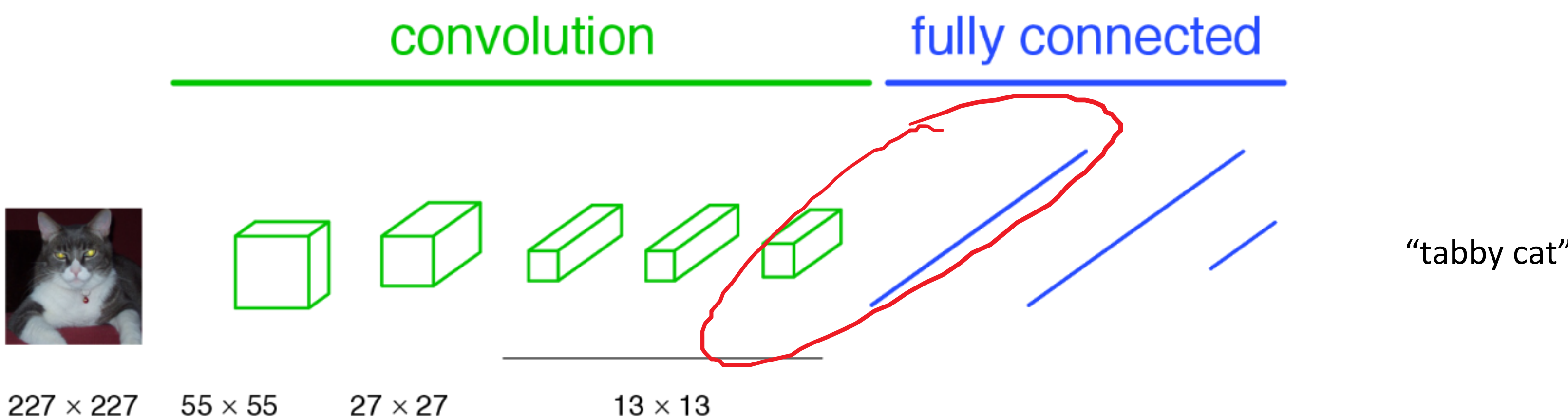
Slides from Long, Shelhamer, and Darrell

A classification network…



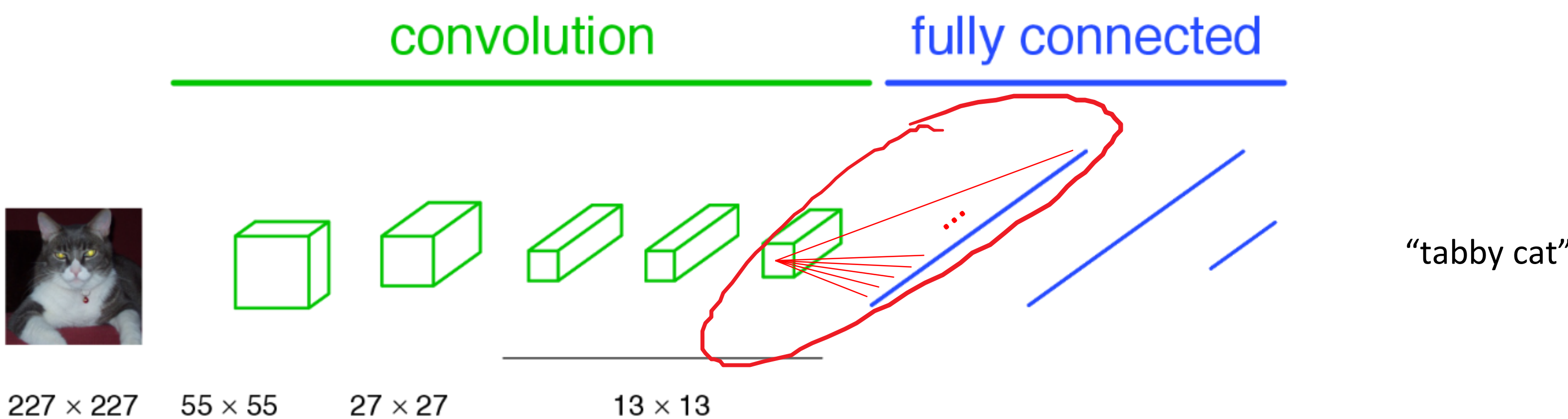
[Long et al.]

A classification network…



[Long et al.]

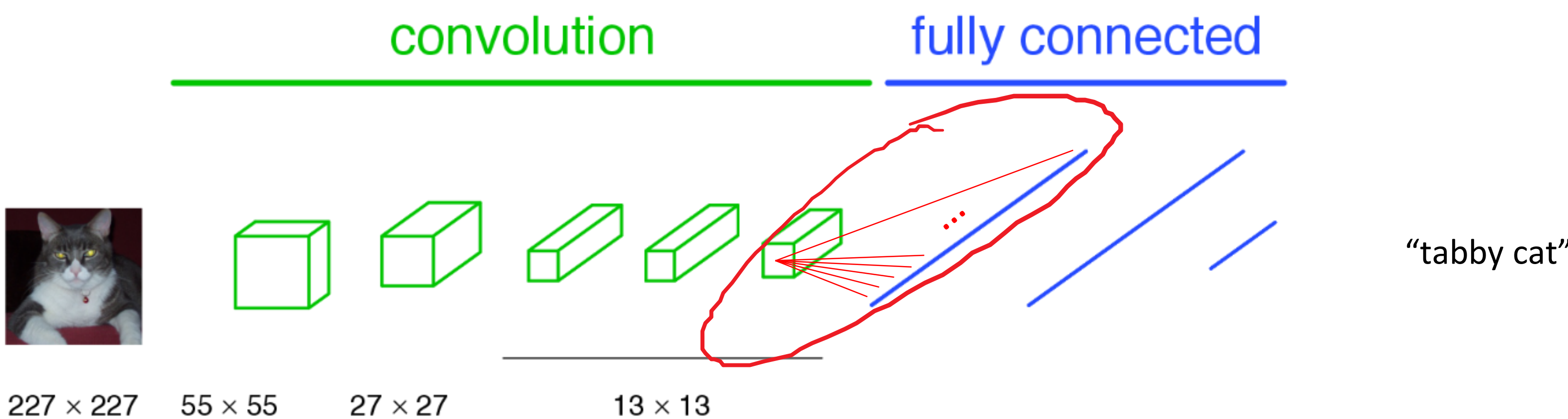
A classification network…



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.

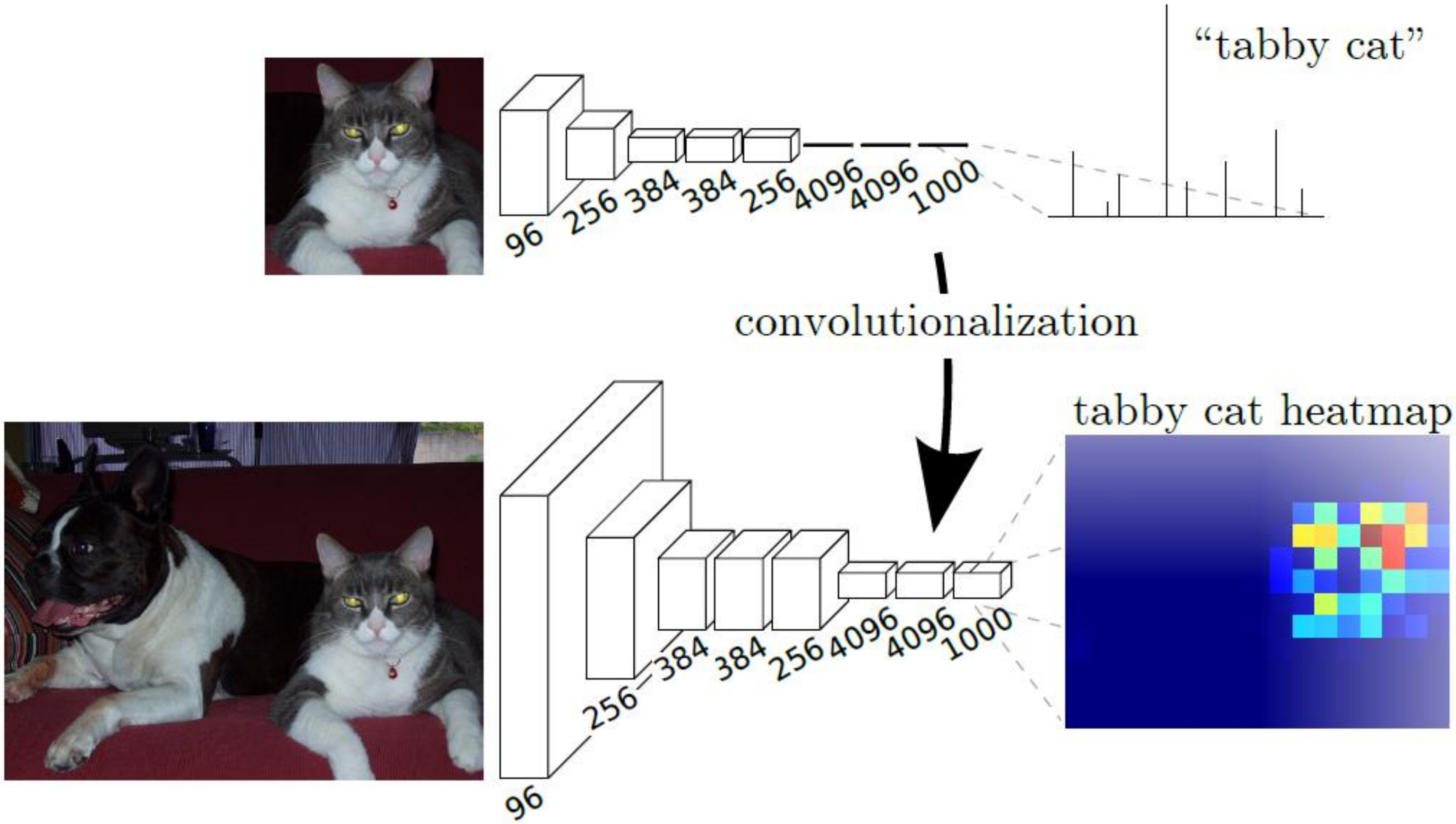
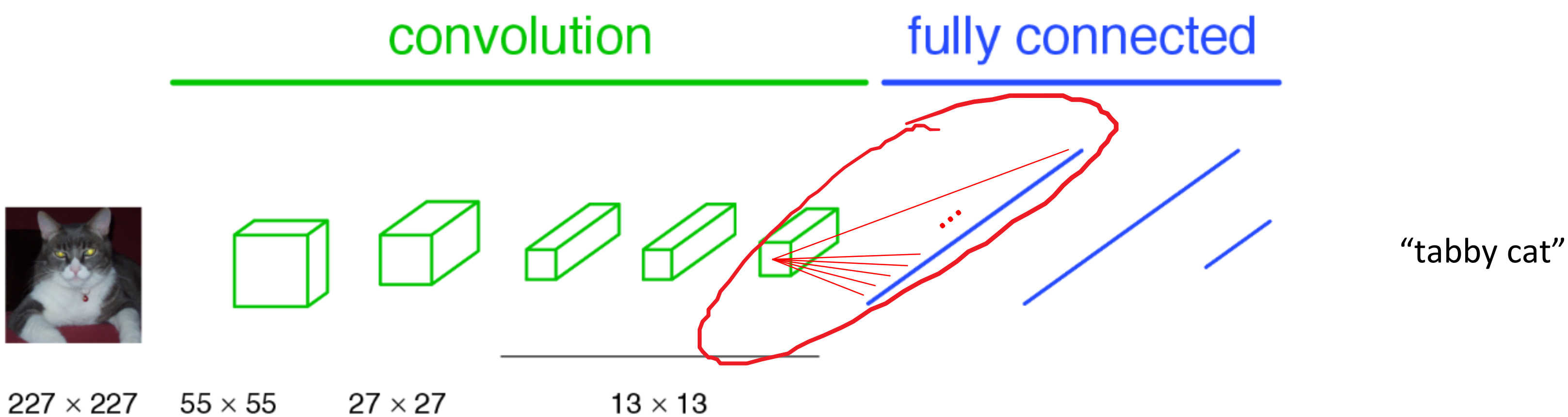


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

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.]

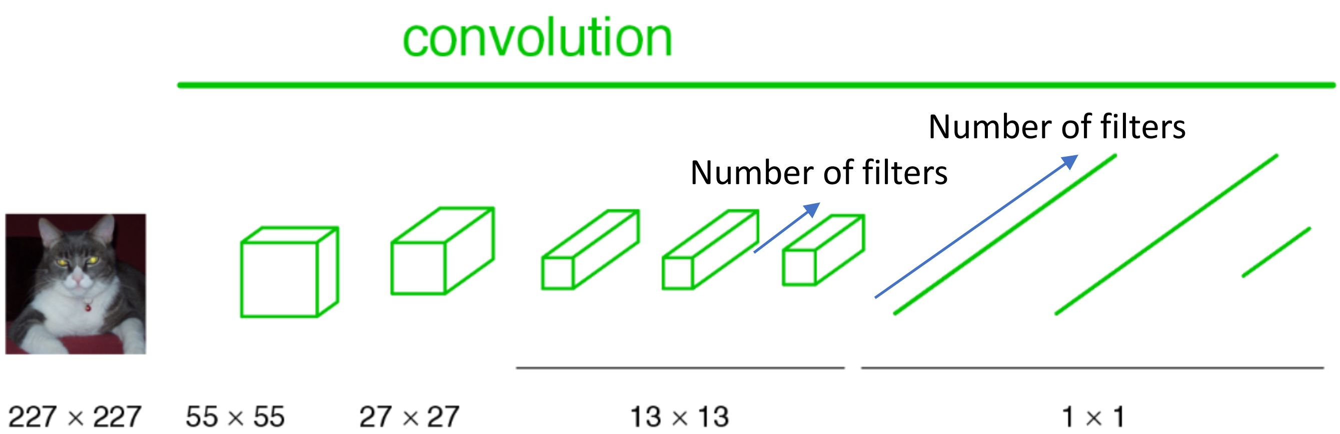
Yann LeCun

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



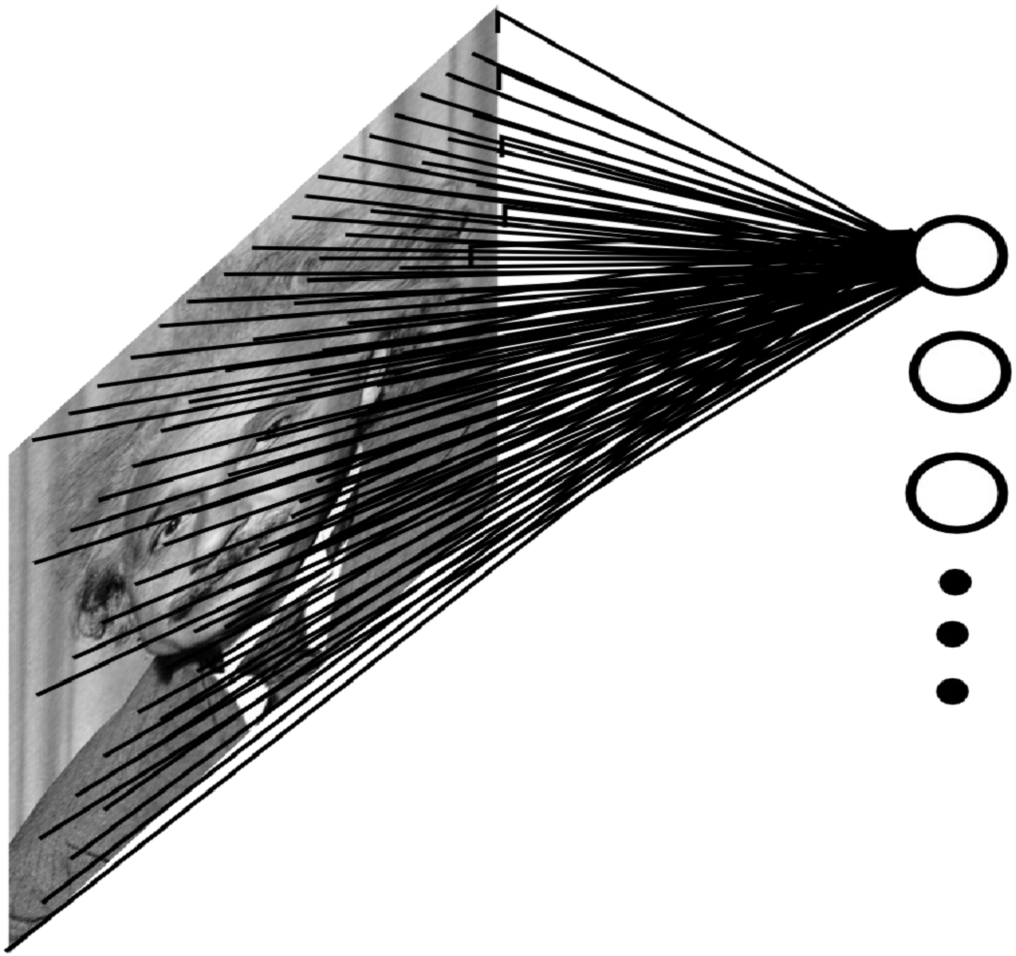
1x1 convolution operates across all filters in the previous layer, and is slid across all positions.

[Long et al.]

Back to the fully-connected perceptron…

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

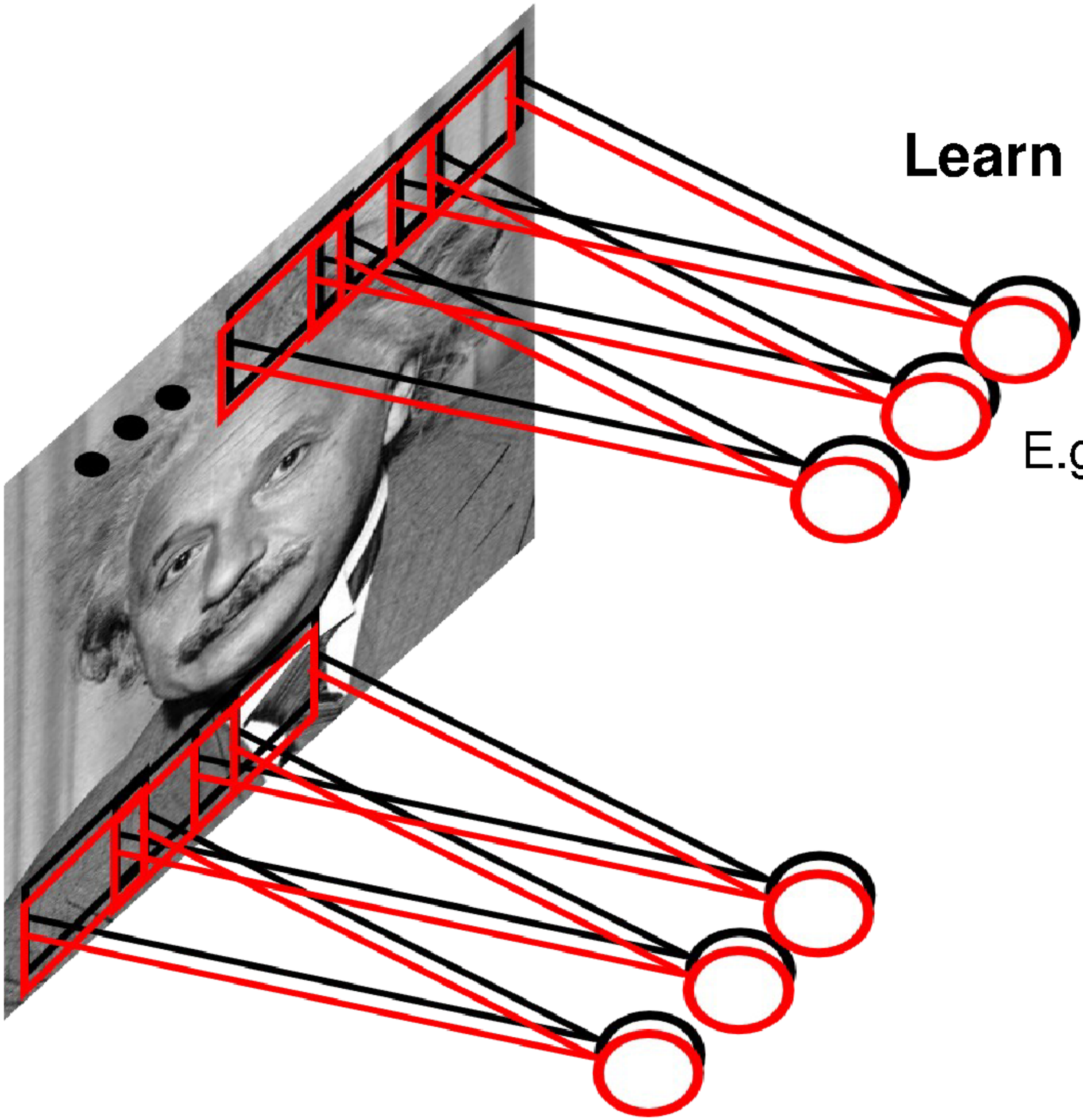
$$w \cdot x \equiv \sum_j w_j x_j,$$



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

[Long et al.]

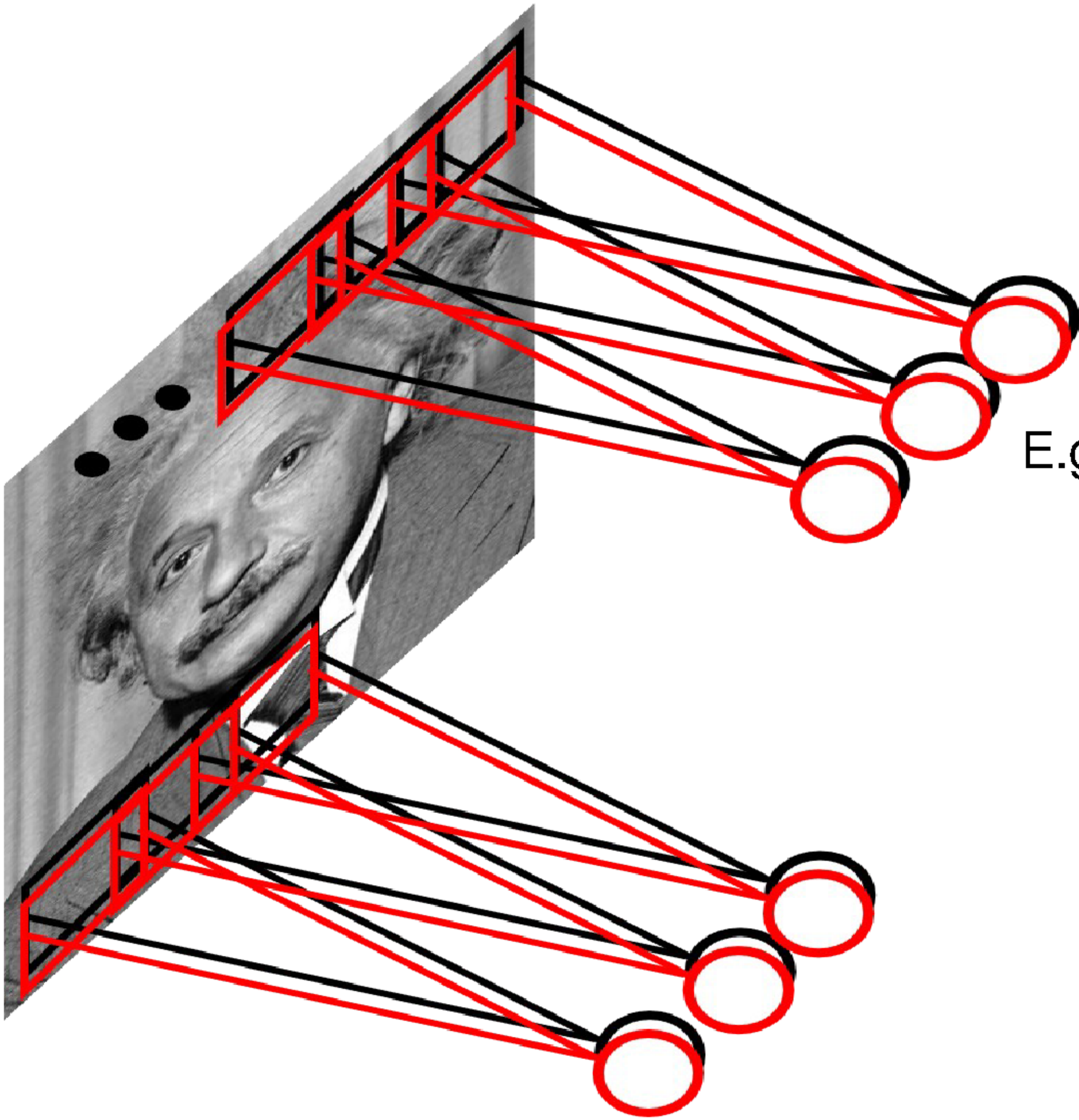
Convolutional Layer



Learn multiple filters.

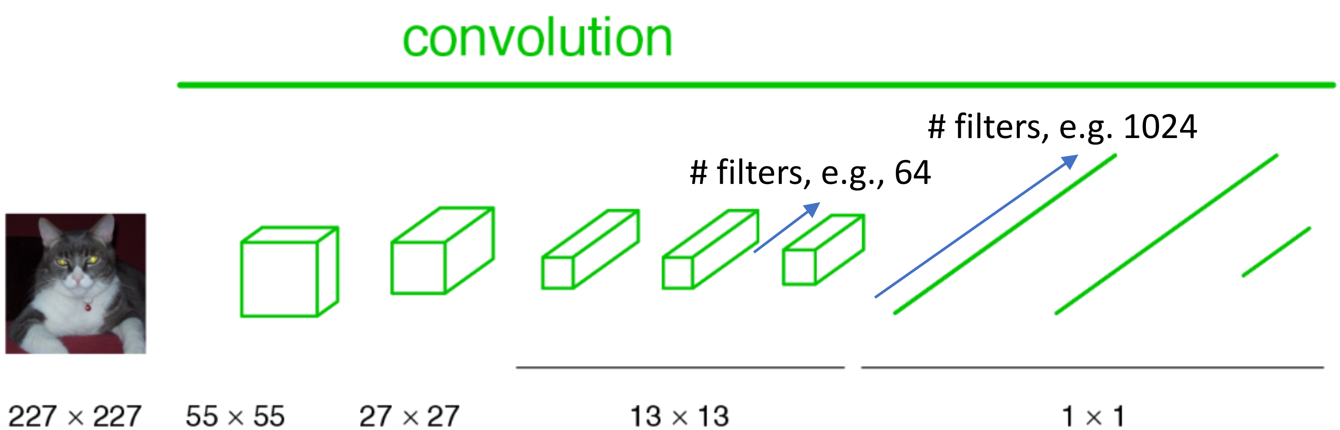
E.g.: 200x200 image
100 Filters
Filter size: 10x10
10K parameters

Convolutional Layer



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

Convolutionalization

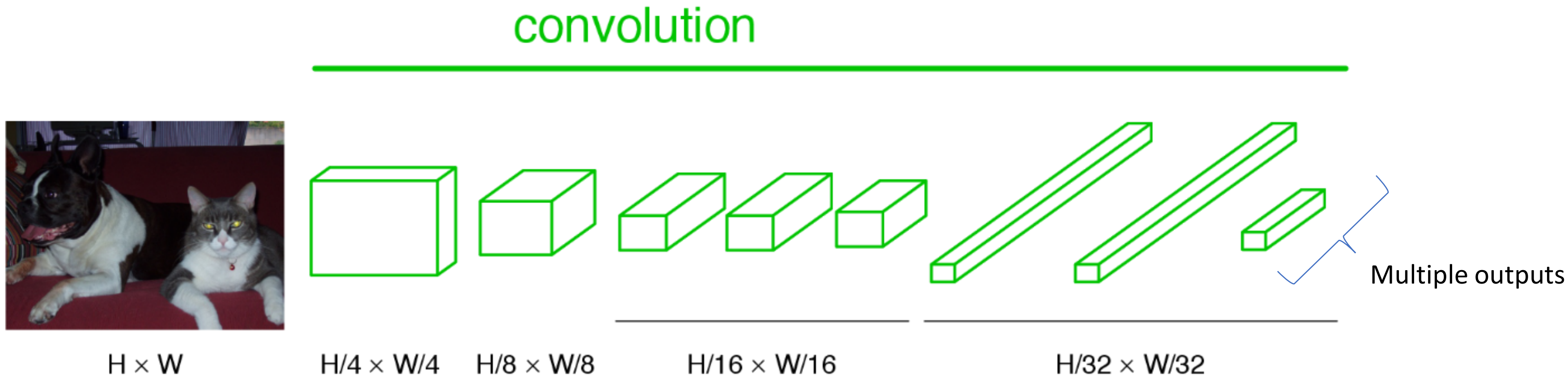


1x1 convolution operates across all filters in the previous layer, and is slid across all positions.

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.]

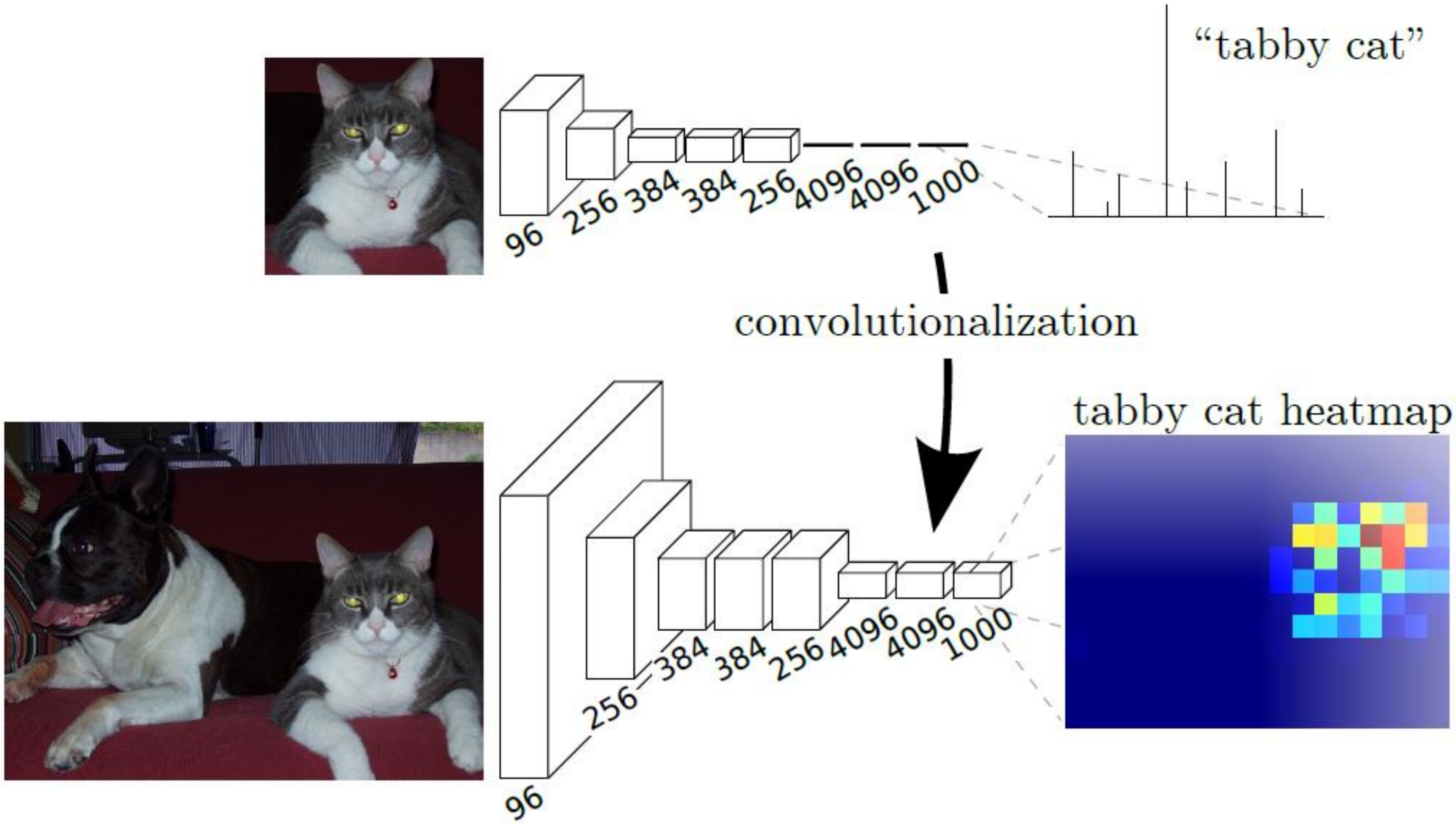
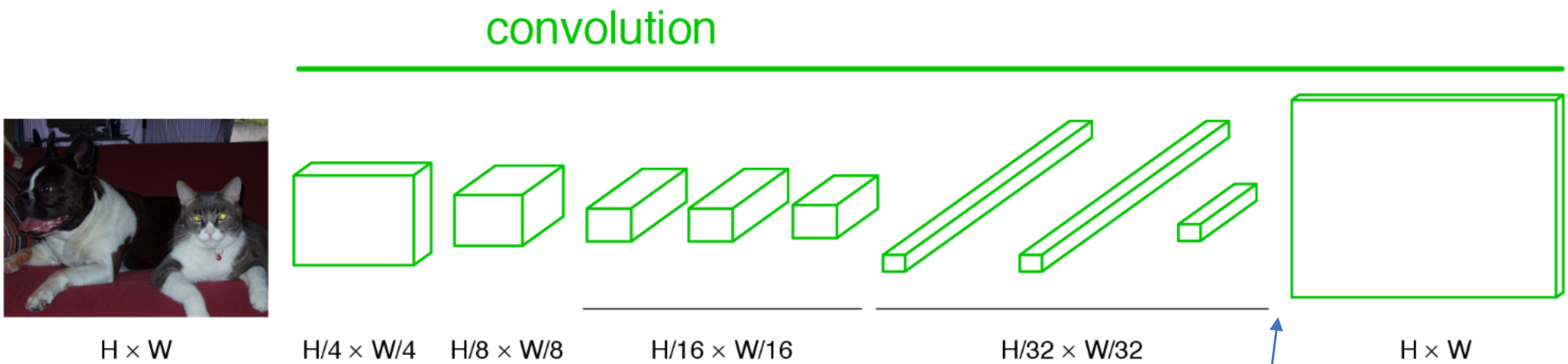


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

Long, Shelhamer, and Darrell 2014

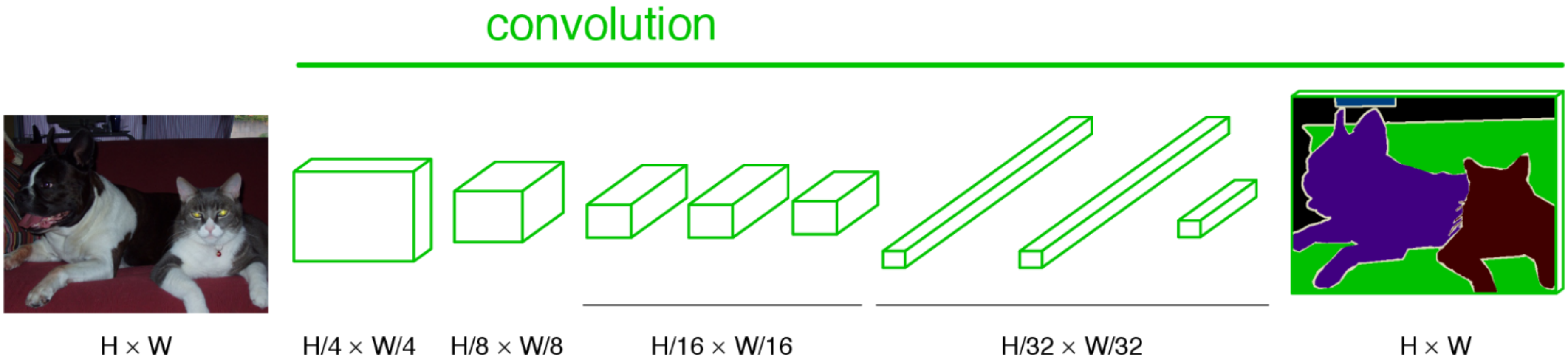
Upsampling the output



Some upsampling
algorithm to
return us to H x W

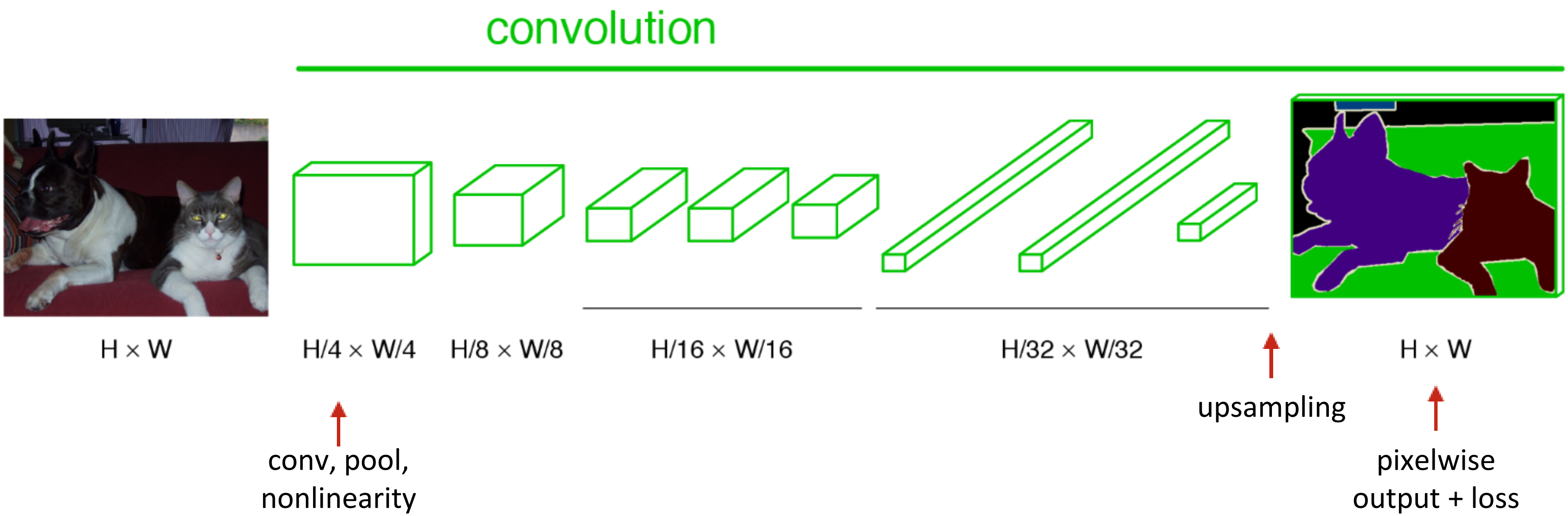
[Long et al.]

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



[Long et al.]

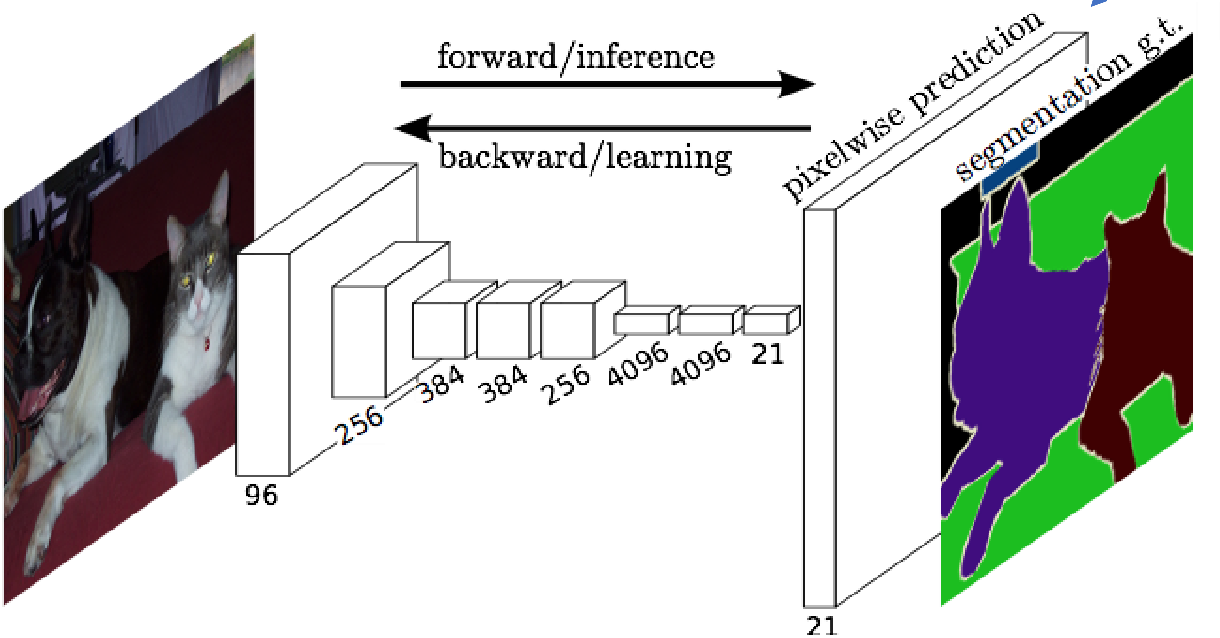
End-to-end, pixels-to-pixels network



[Long et al.]

What is the upsampling layer?

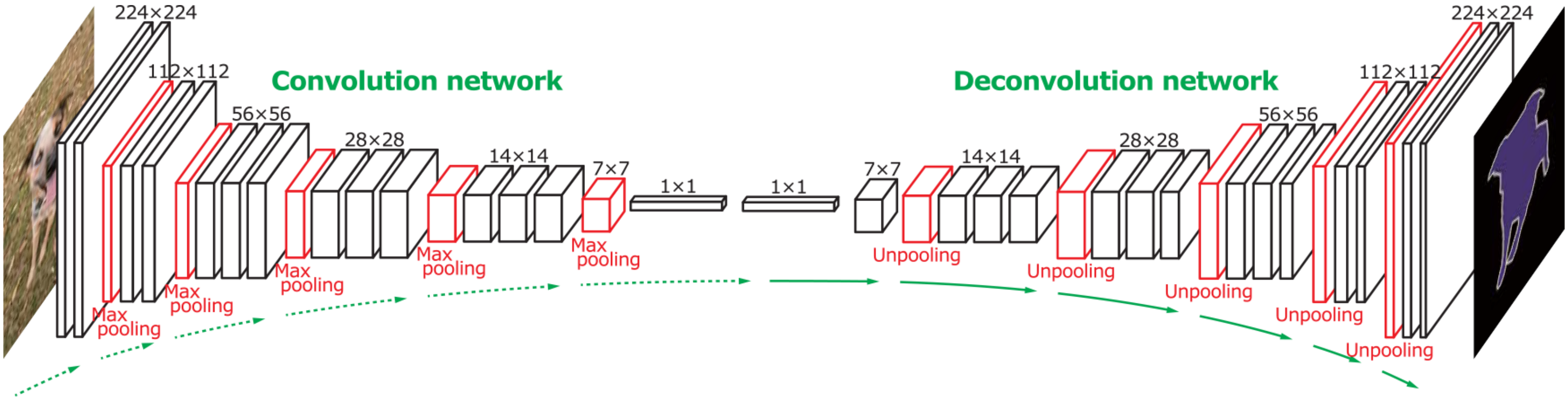
This one.



Hint: it's actually an upsampling *network*

[Long et al.]

‘Deconvolution’ networks *learn to upsample*



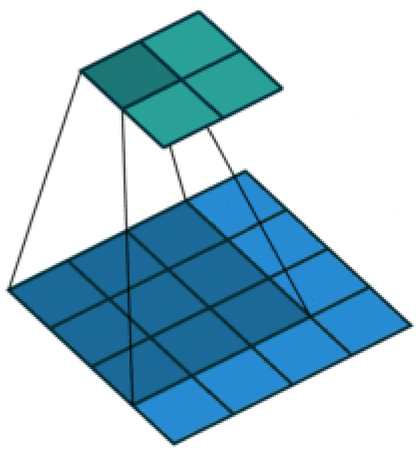
Often called “deconvolution”, but misnomer.
Not the deconvolution that we saw in deblurring -> that is division in the Fourier domain.

‘Transposed convolution’ is better.

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

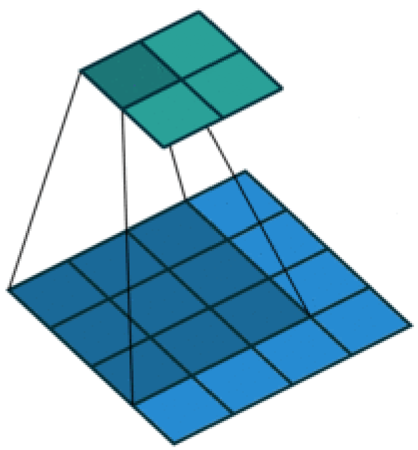
Upsampling with transposed convolution

Convolution

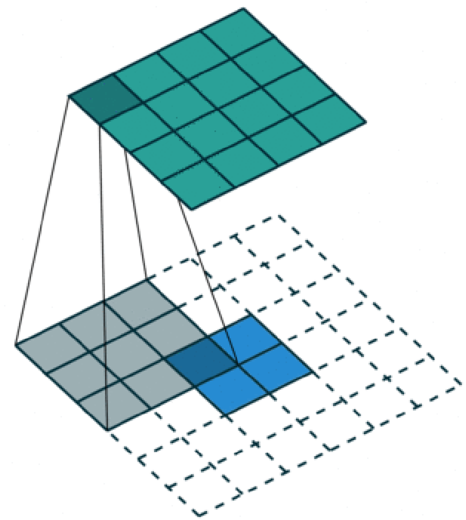


Upsampling with transposed convolution

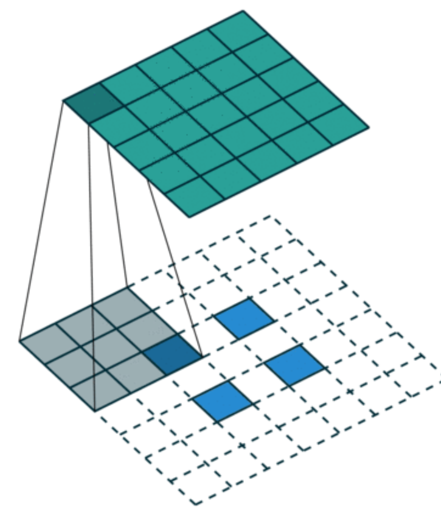
Convolution



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



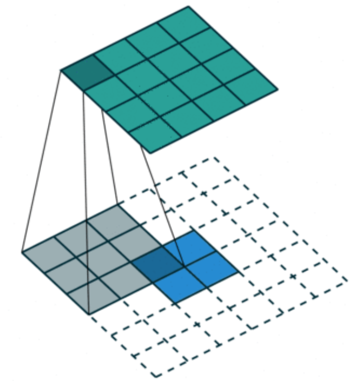
2x2, stride 1, 3x3 kernel, upsample to 4x4



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

Kernel

1	1	1
1	1	1
1	1	1



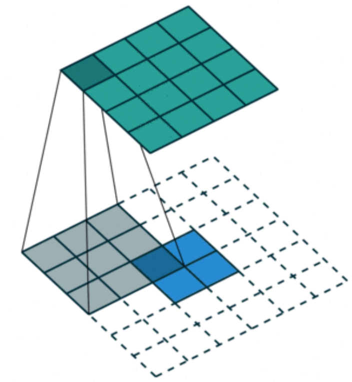
Feature map

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

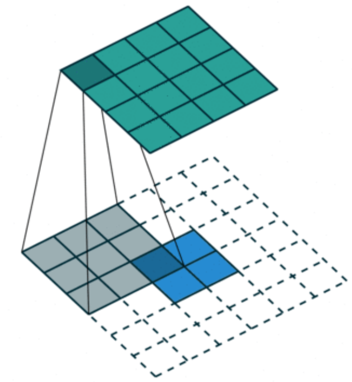
1	1	1
1	1	1
1	1	1

Input feature map

1	2
3	4

Padded input feature map

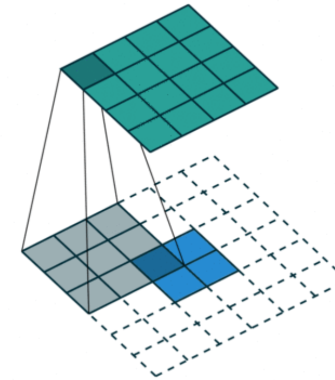
		1	2		
		3	4		



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

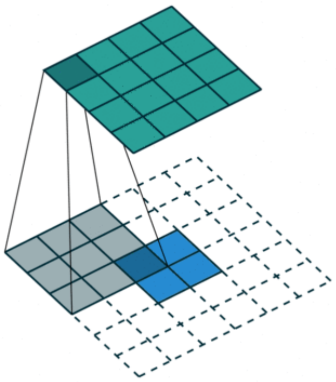
1	1	1
1	1	1
1	1	1

Input feature map

1	2
3	4

Padded input feature map

		1	2		
		3	4		



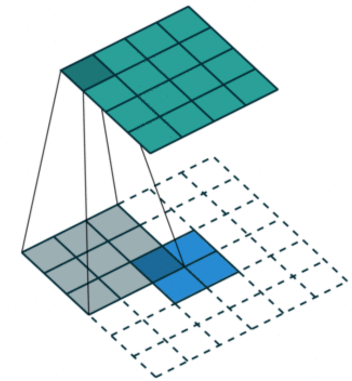
Output feature map

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

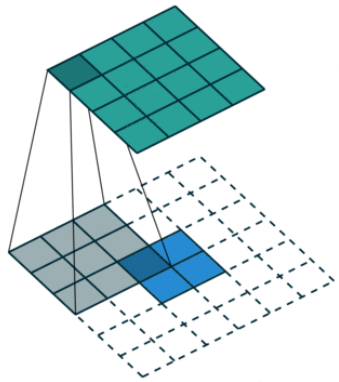
Padded input feature map

		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

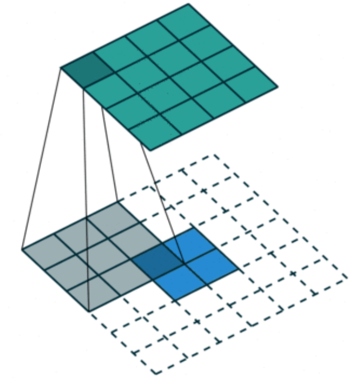
Padded input feature map

		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

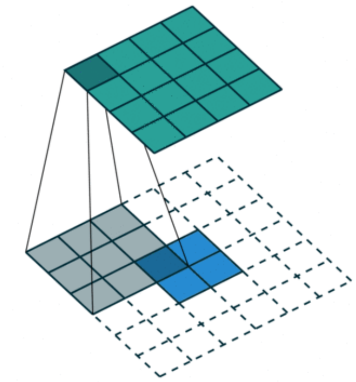
Padded input feature map

		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

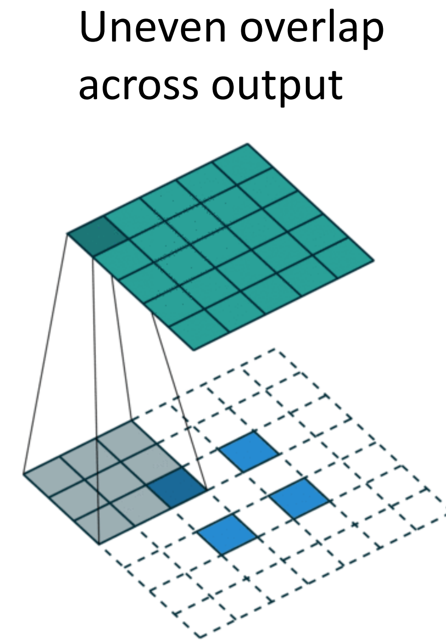
		1	2		
		3	4		

Cropped output feature map

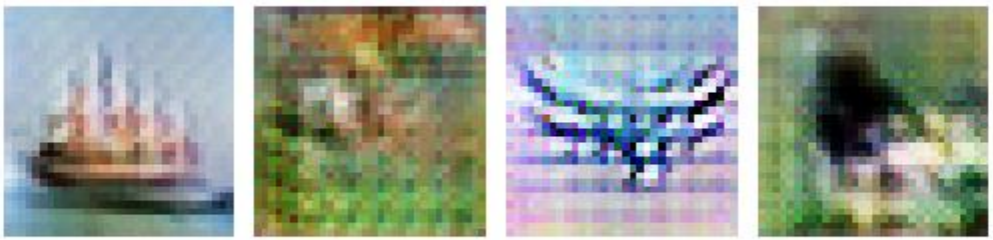
18	31	34	21
32	55	60	37
38	66	64	43
24	41	44	27

Inspired by andriys

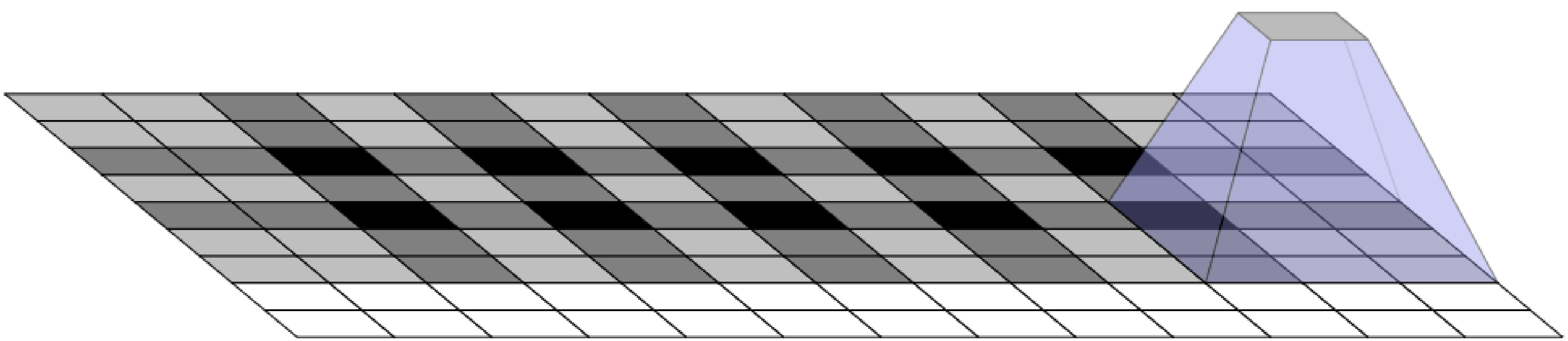
Is uneven overlap a problem?



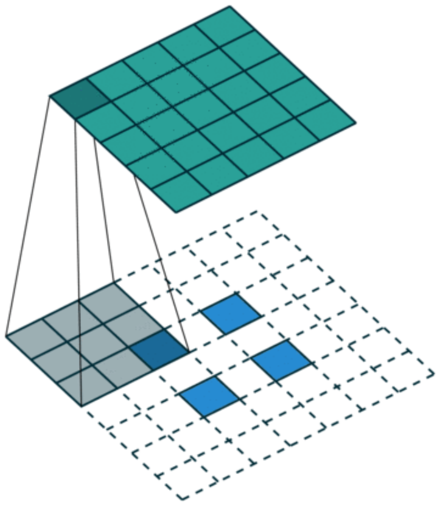
Yes = causes grid artifacts



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

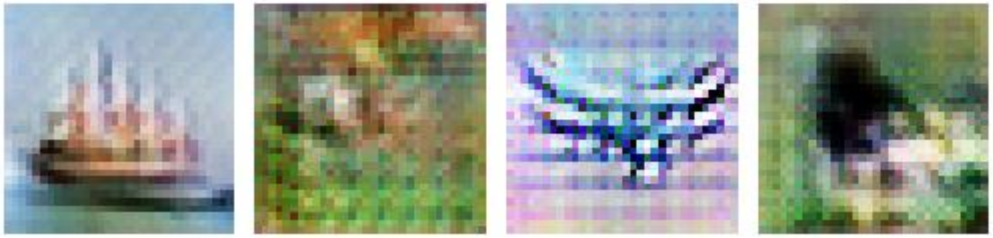


Uneven overlap
across output



Is uneven overlap a problem?

Yes = causes grid artifacts

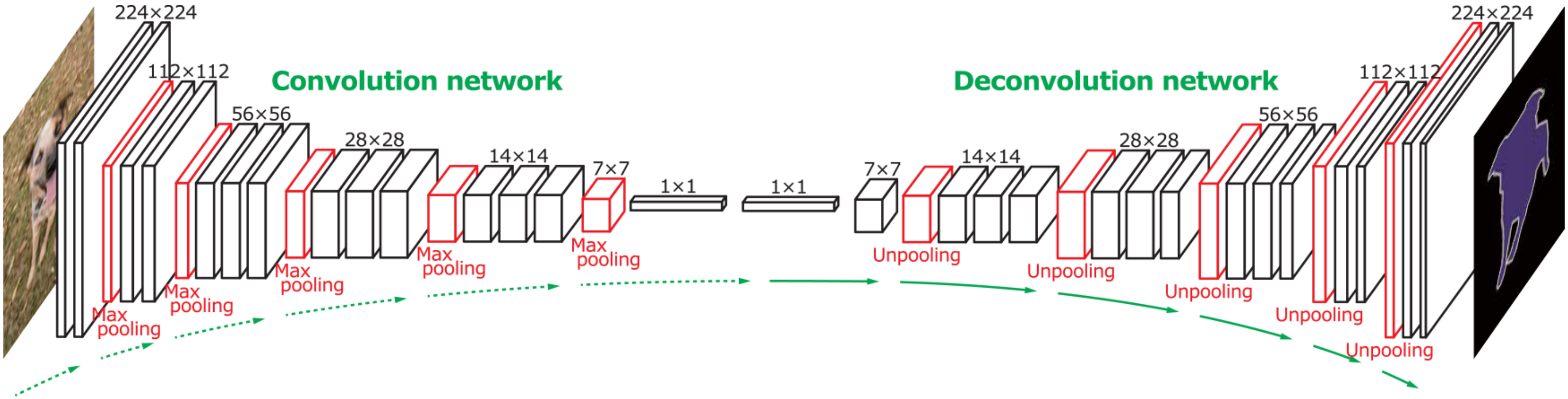


Could fix it by picking stride/kernel numbers which have no overlap…

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.

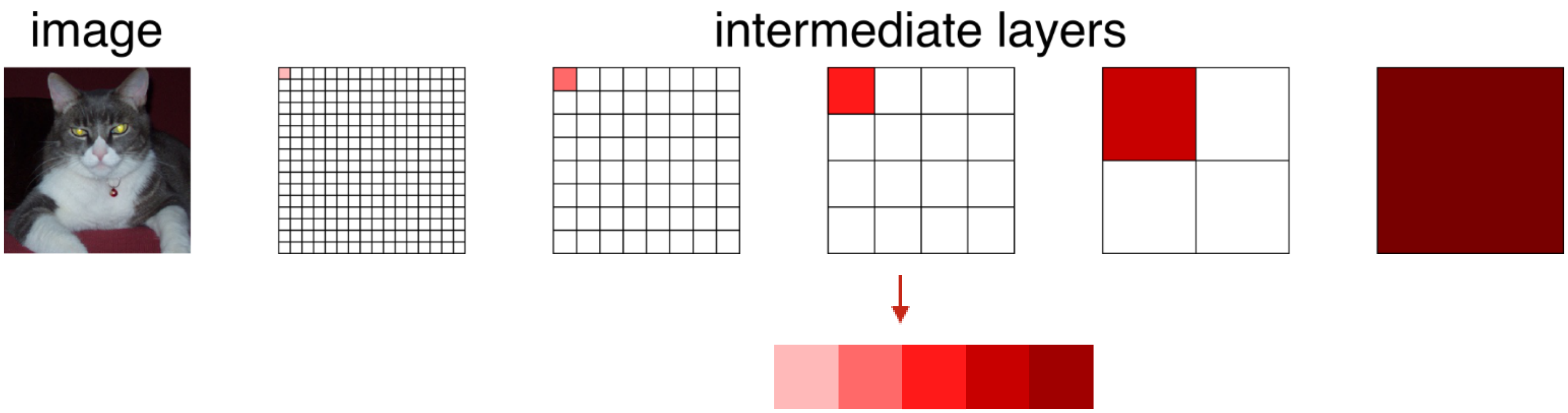
Zeiler et al., Deconvolutional Networks, CVPR 2010
Noh et al., Learning Deconvolution Network for Semantic Segmentation, ICCV 2015

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)

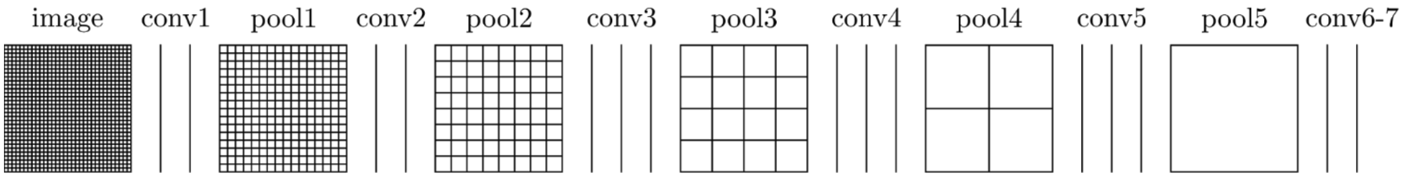


Fuse features into **deep jet**

(cf. Hariharan et al. CVPR15 "hypercolumn")

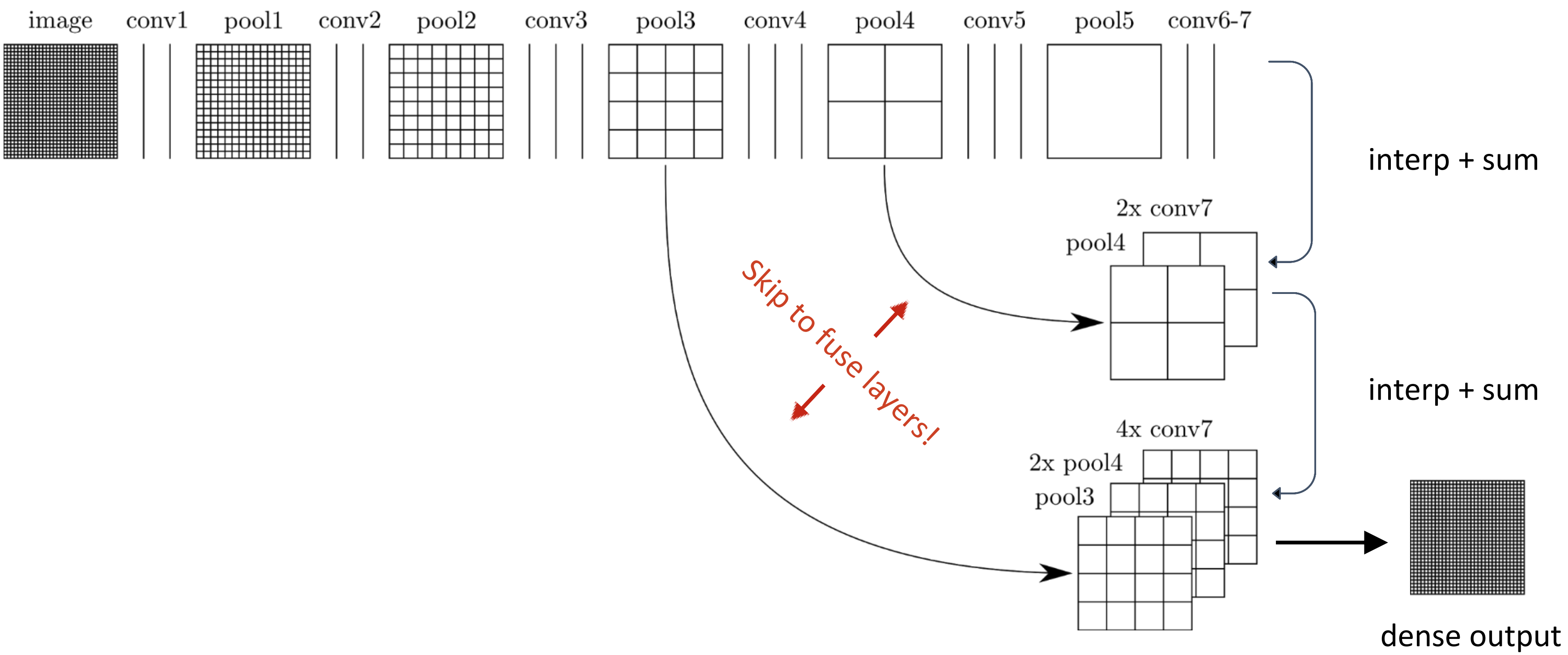
[Long et al.]

Learning upsampling kernels with skip layer refinement



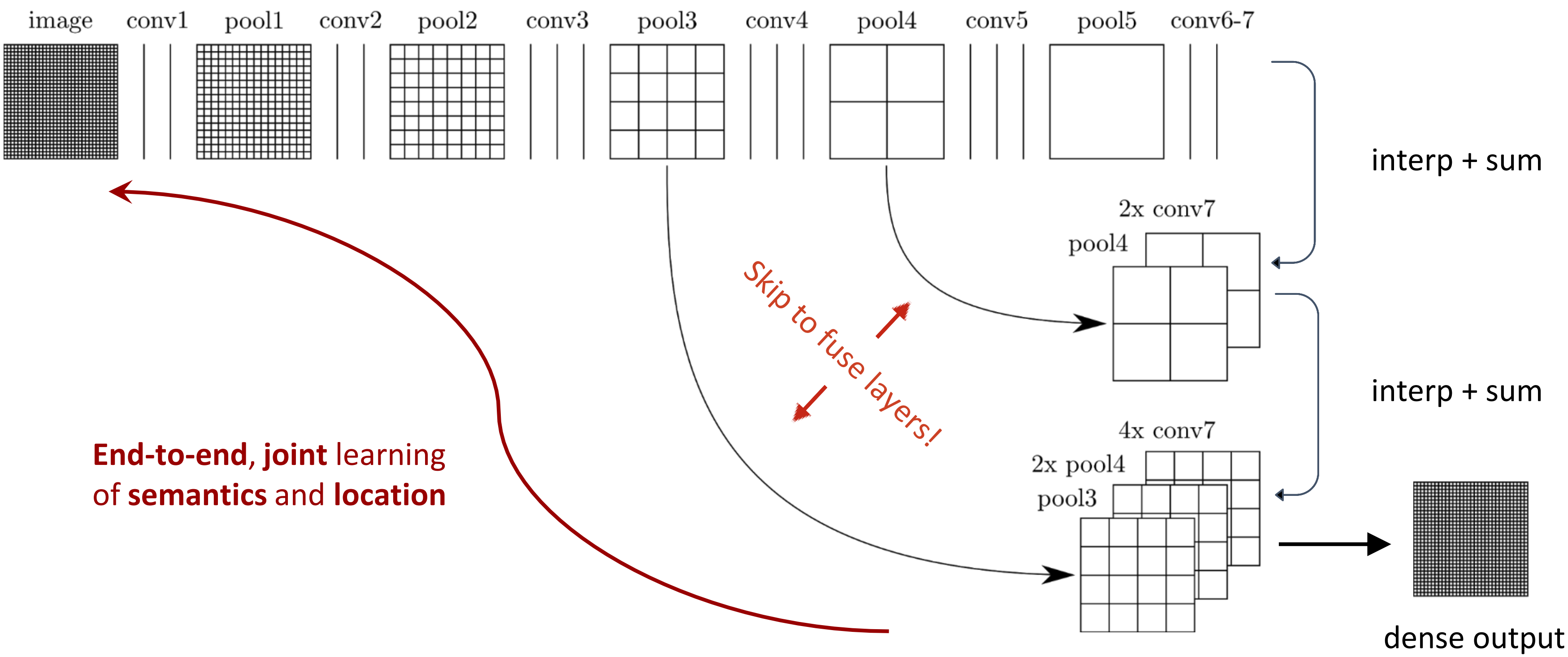
[Long et al.]

Learning upsampling kernels with skip layer refinement



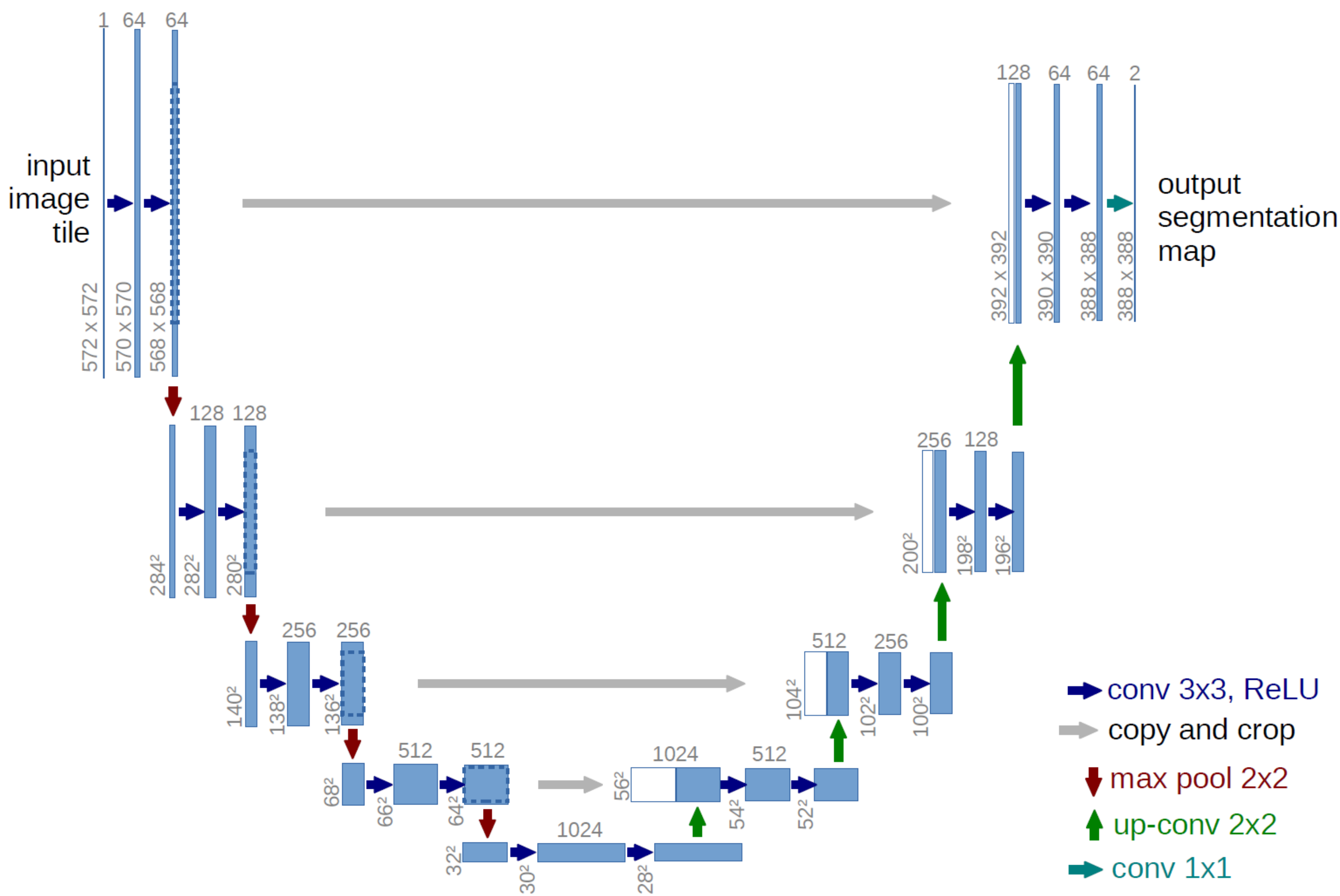
[Long et al.]

Learning upsampling kernels with skip layer refinement

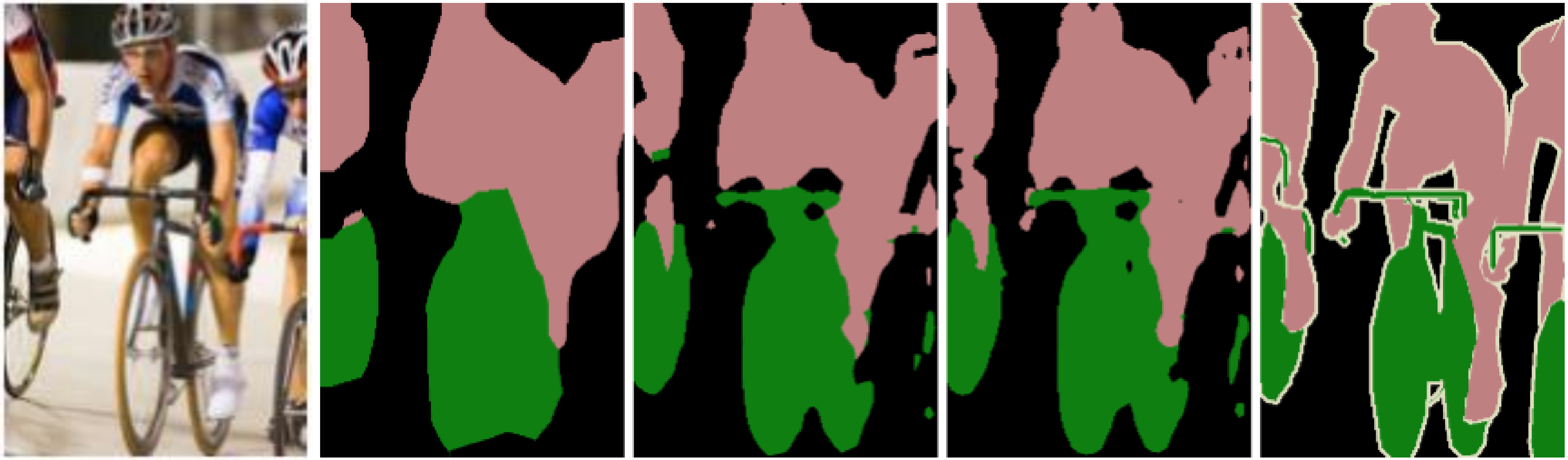


[Long et al.]

UNet [Ronneberger et al., 2015]

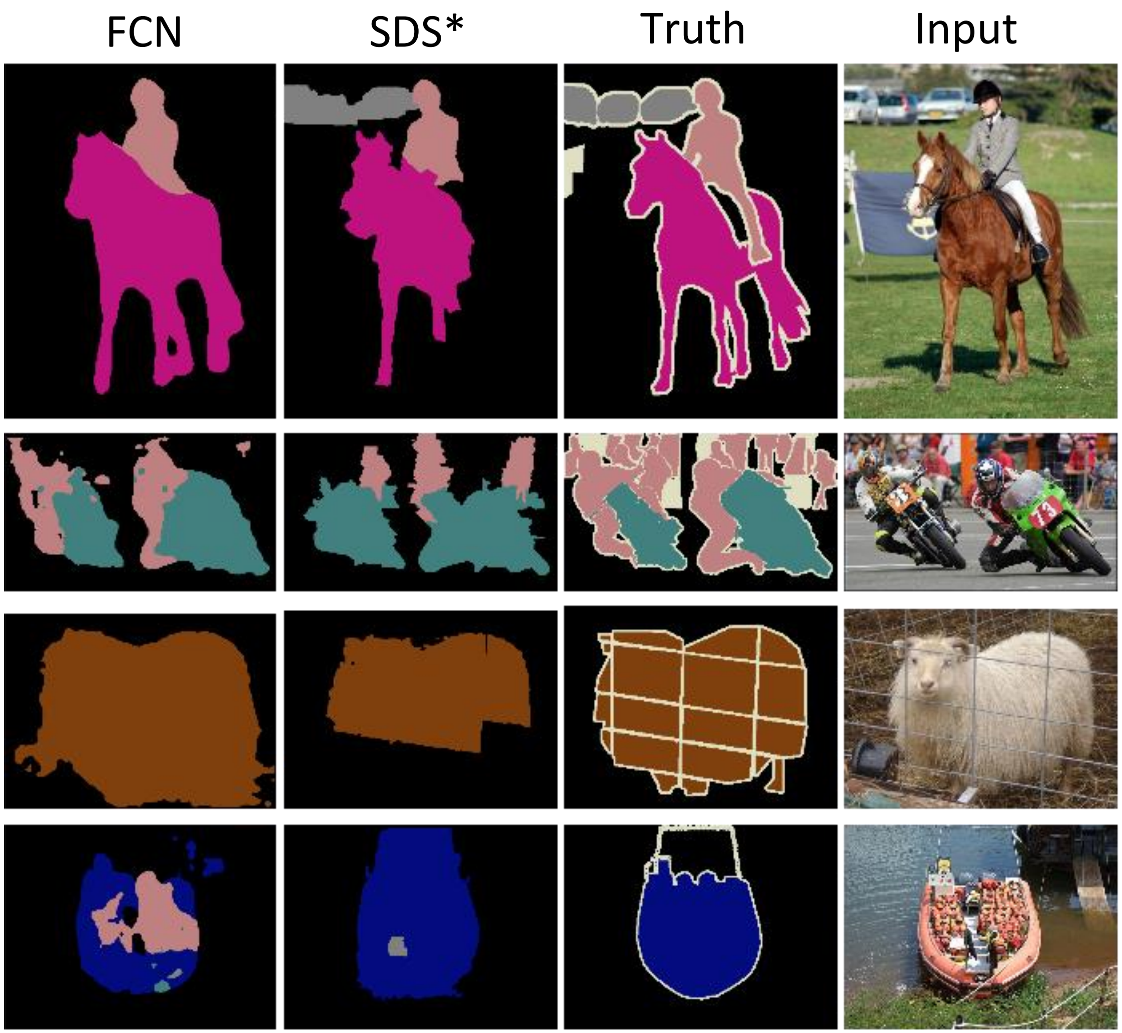


Skip layer refinement



[Long et al.]

Results



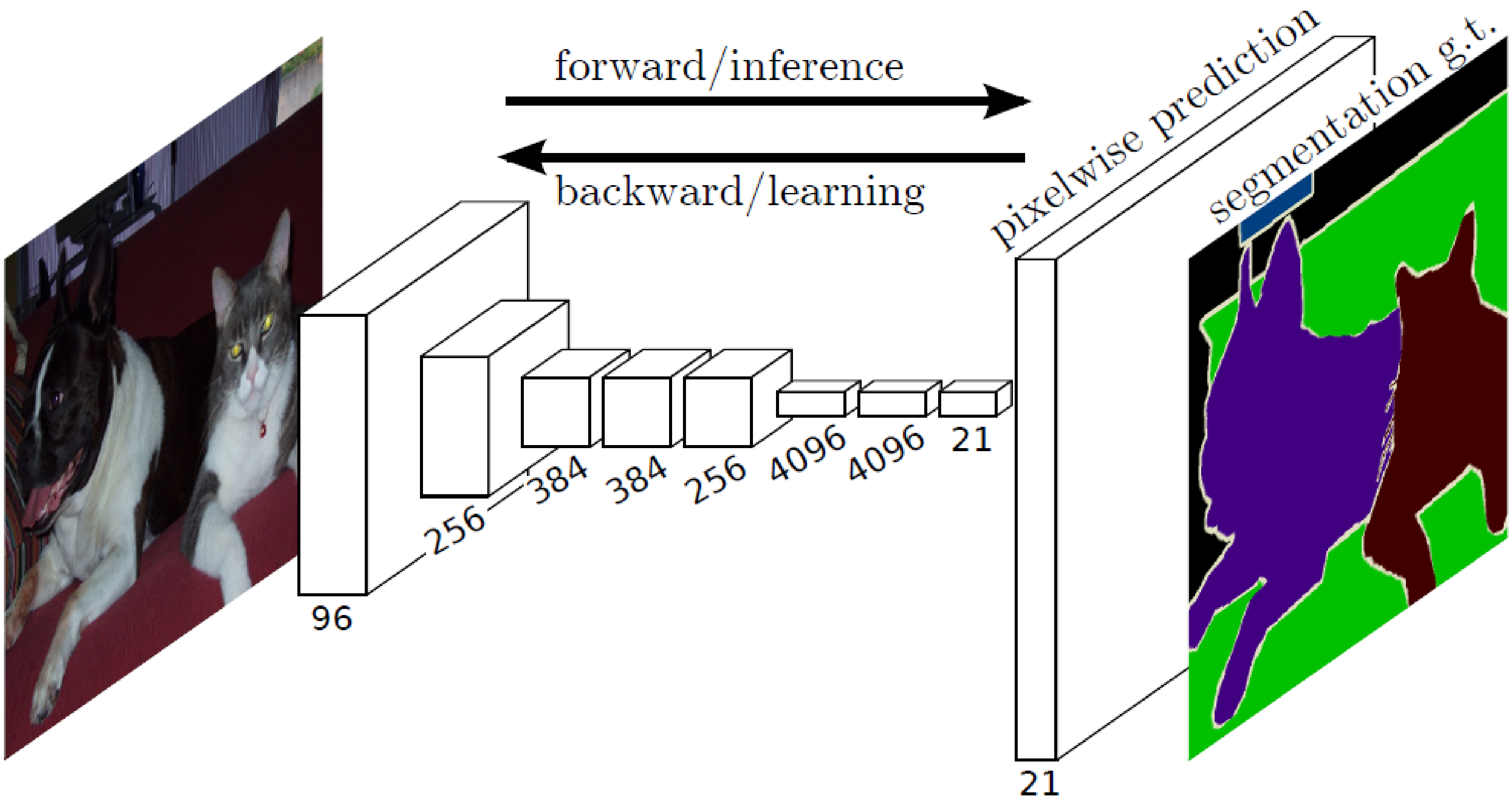
Relative to prior state-of-the-art SDS:

- 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

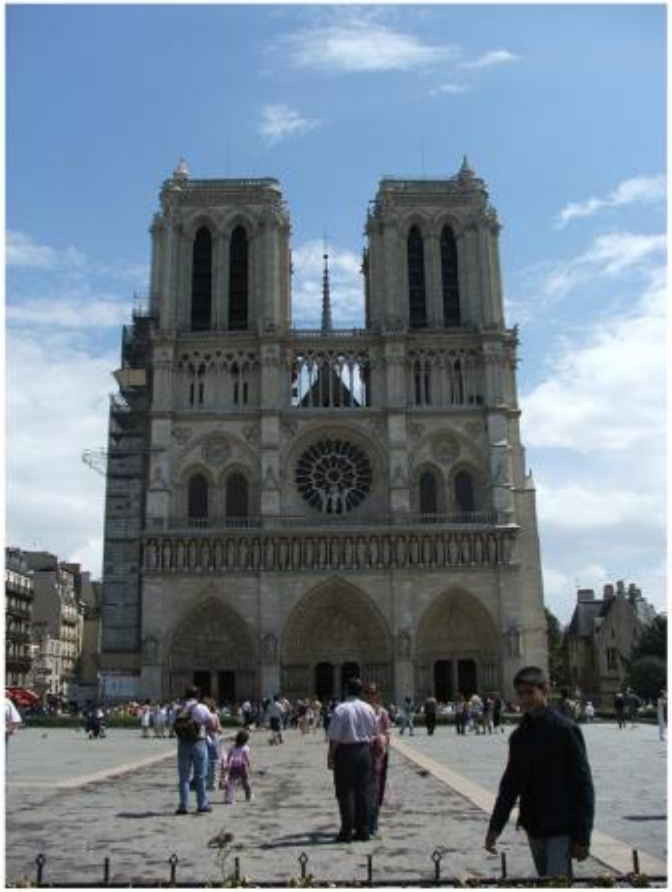
How much can an image tell about its geographic location?

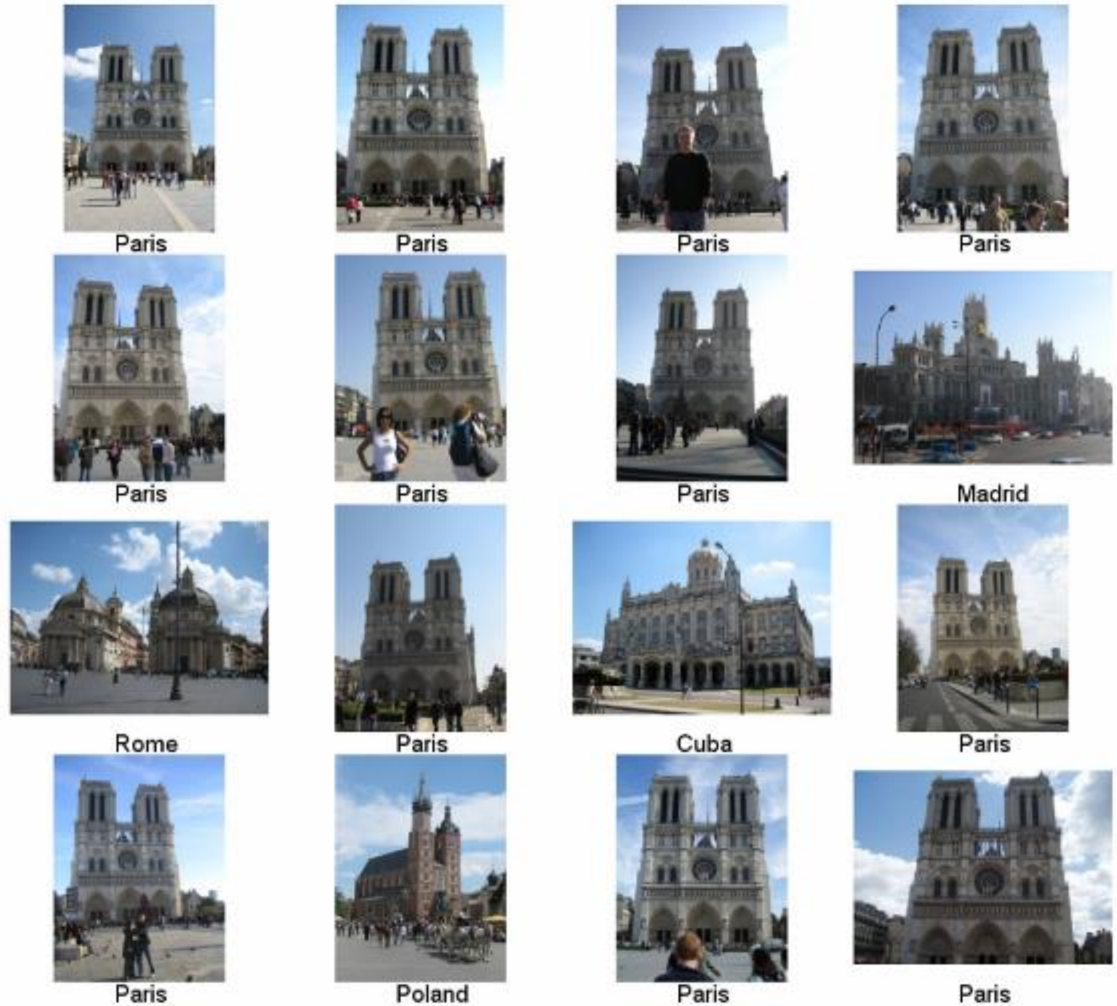
6 million geo-tagged Flickr images

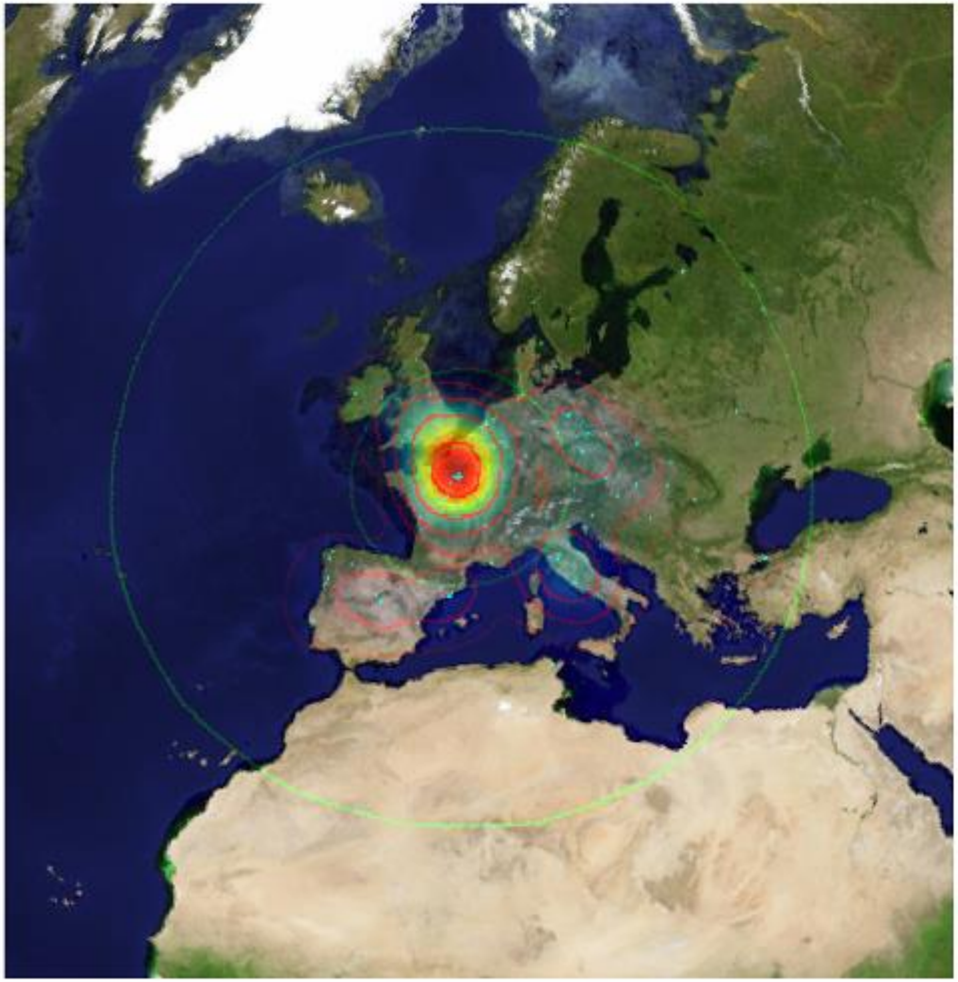
http://graphics.cs.cmu.edu/projects/im2gps/

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

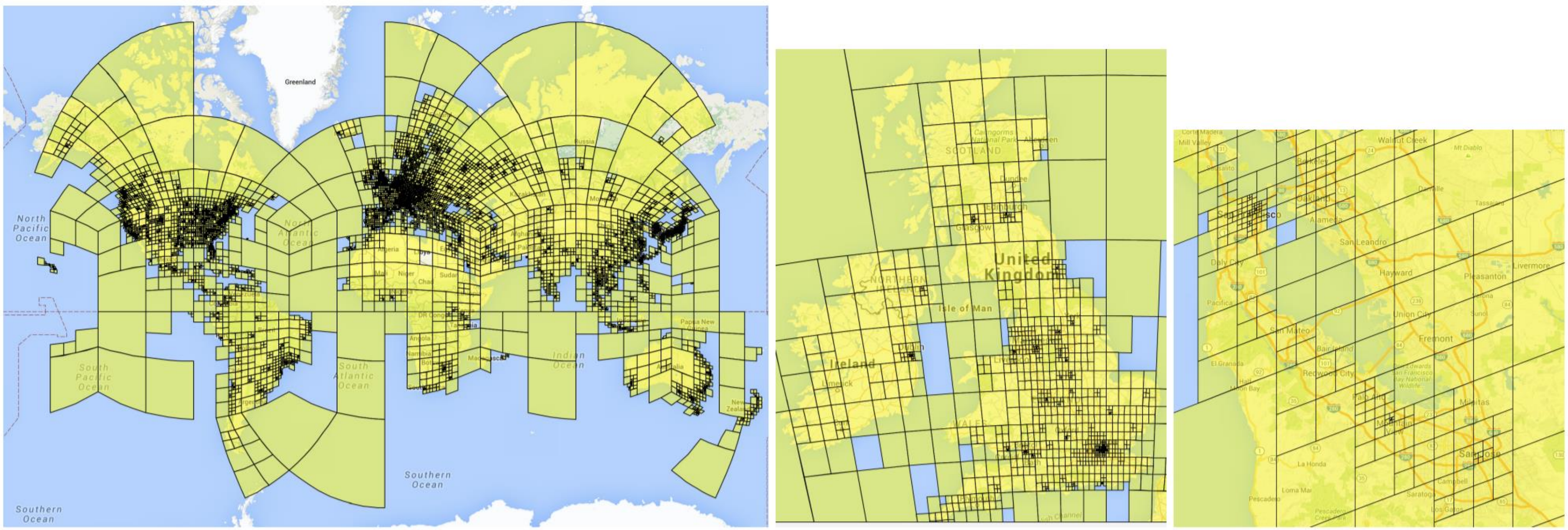


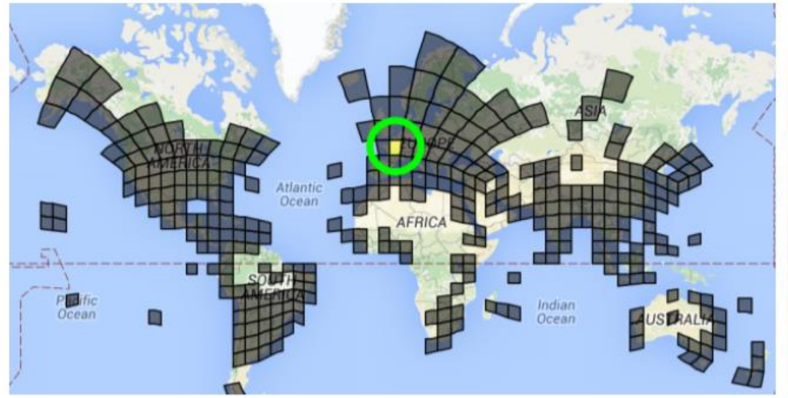
Figure 2. Left: Adaptive partitioning of the world into 26,263 S2 cells. Right: Detail views of Great Britain and Ireland and the San

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)

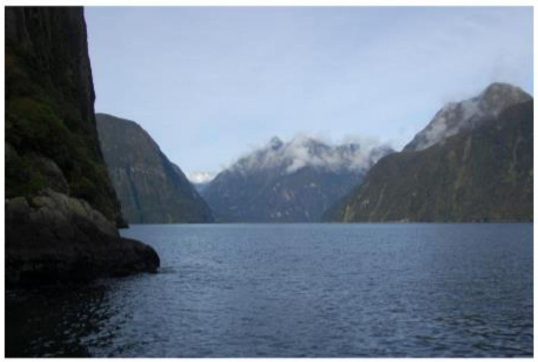
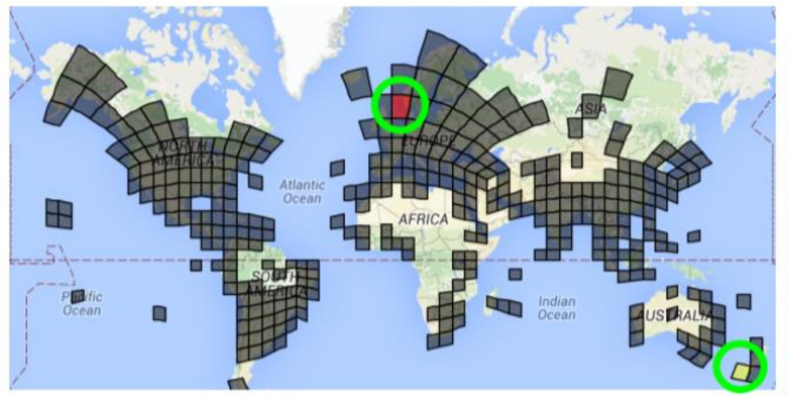


Photo CC-BY-NC by edwin.11



(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

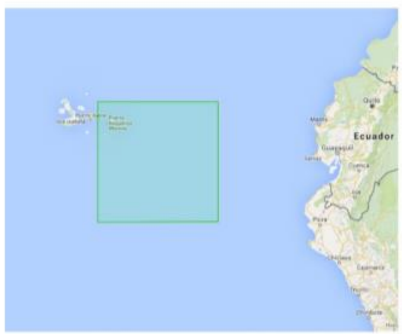
Photo by ryan + sarah / CC BY NC

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

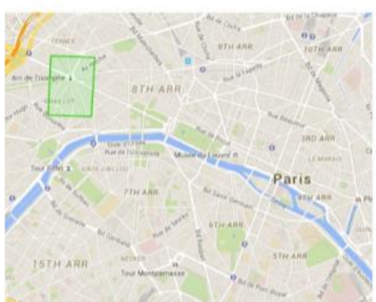
Photo by p.j.k. / CC BY NC

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

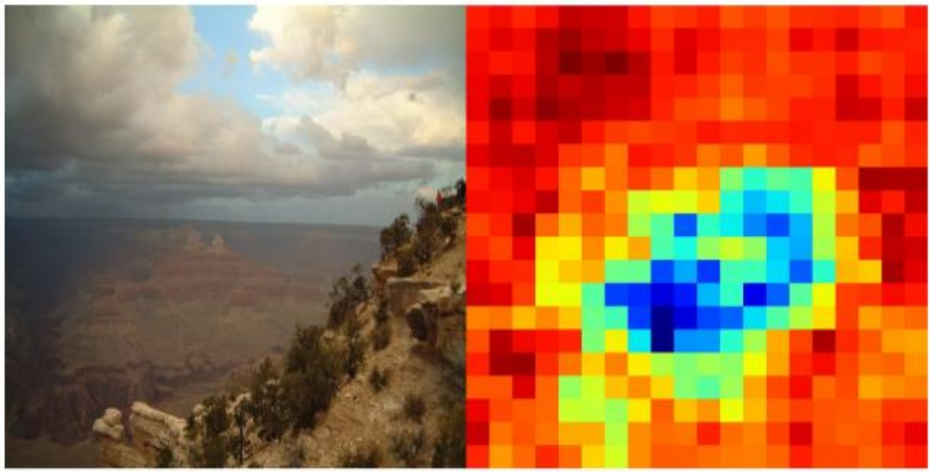
Photo by CedEm photographies / CC BY NC

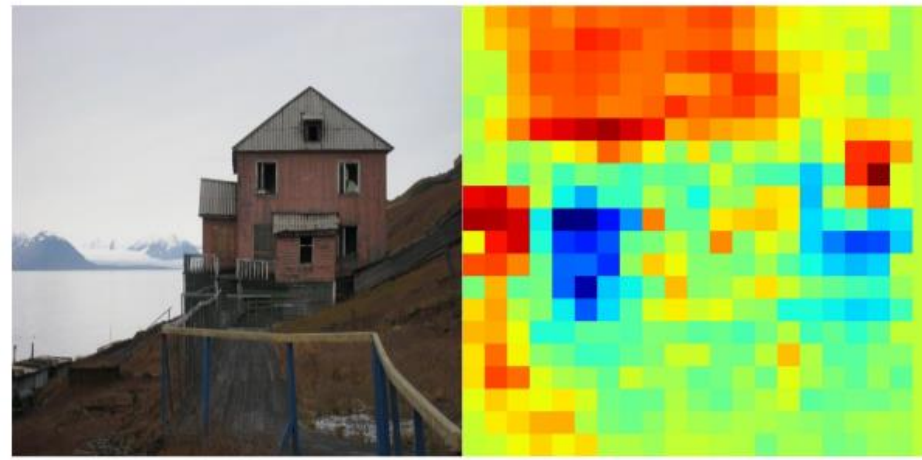
PlaNet vs im2gps (2008, 2009)

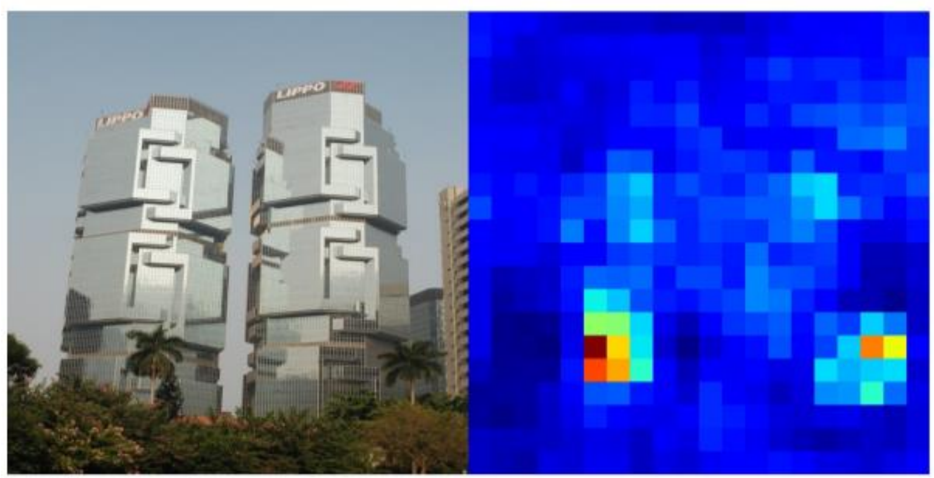
Method	Street 1 km	City 25 km	Region 200 km	Country 750 km	Continent 2500 km
Im2GPS (orig) [17]		12.0%	15.0%	23.0%	47.0%
Im2GPS (new) [18]	2.5%	21.9%	32.1%	35.4%	51.9%
PlaNet	**8.4%**	**24.5%**	**37.6%**	**53.6%**	**71.3%**

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

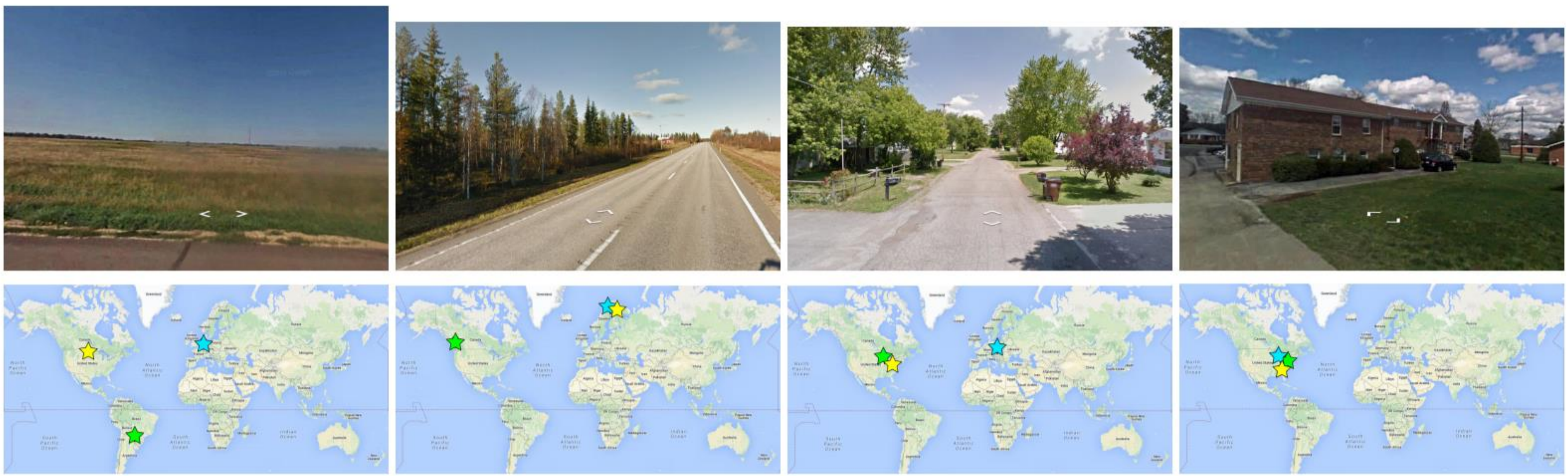
Spatial support for decision



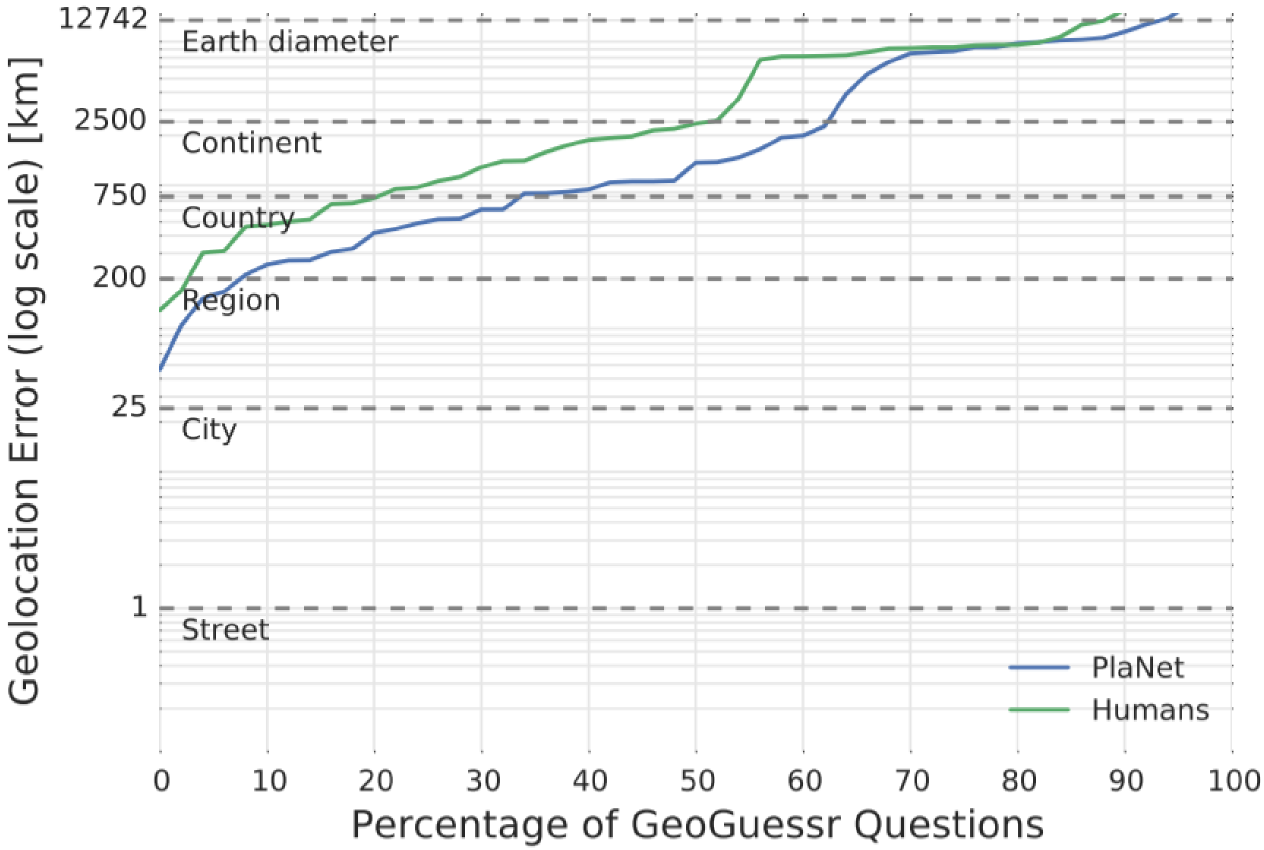




PlaNet vs Humans



PlaNet vs. Humans



PlaNet summary

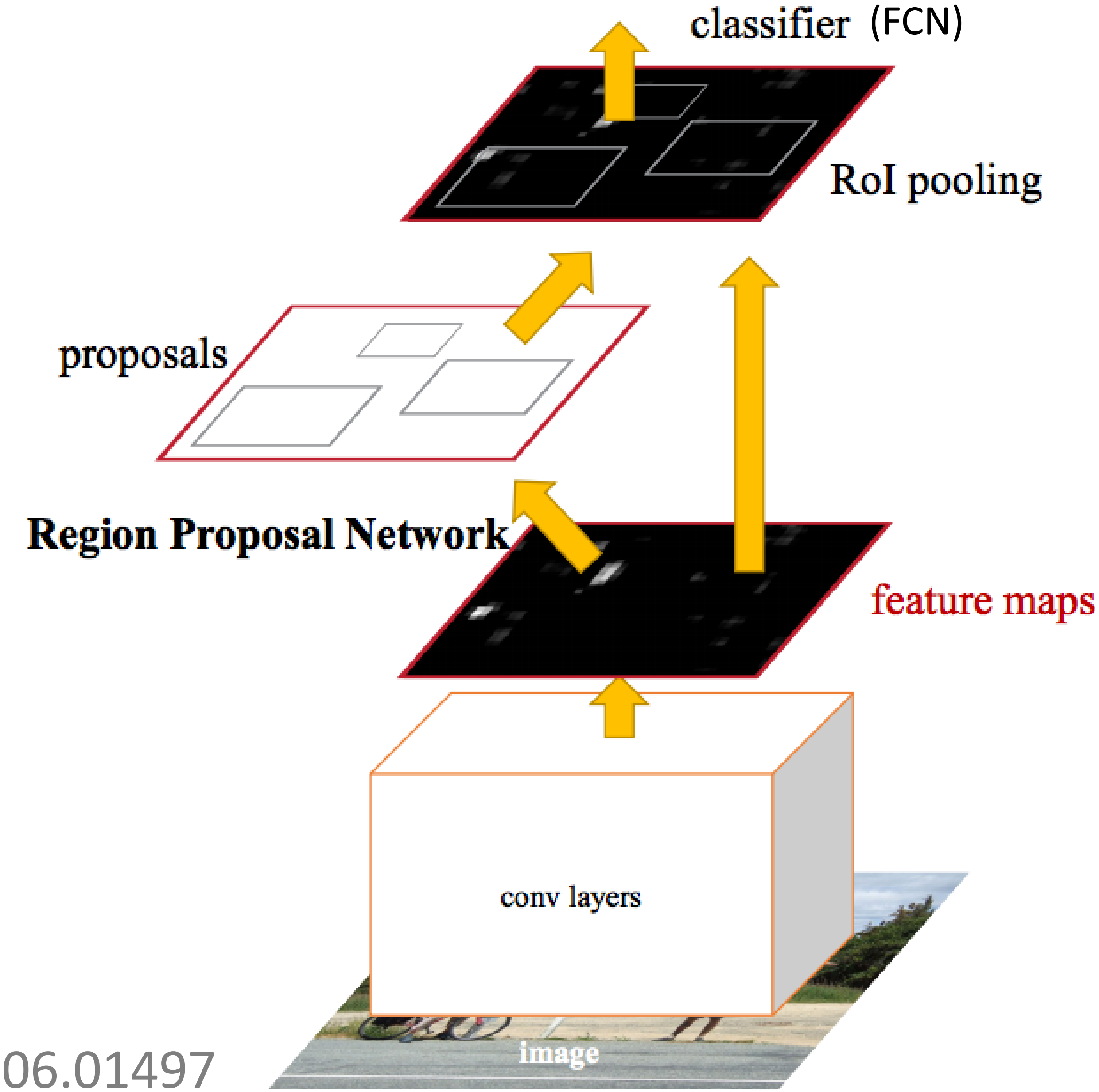
- Very fast geolocalization method by categorization.
- Uses far more training data than previous work (im2gps)
- Better than humans!

Even more: Faster R-CNN

'Region Proposal Network' uses CNN feature maps.

Then, FCN on top to classify.

End to end object detection.



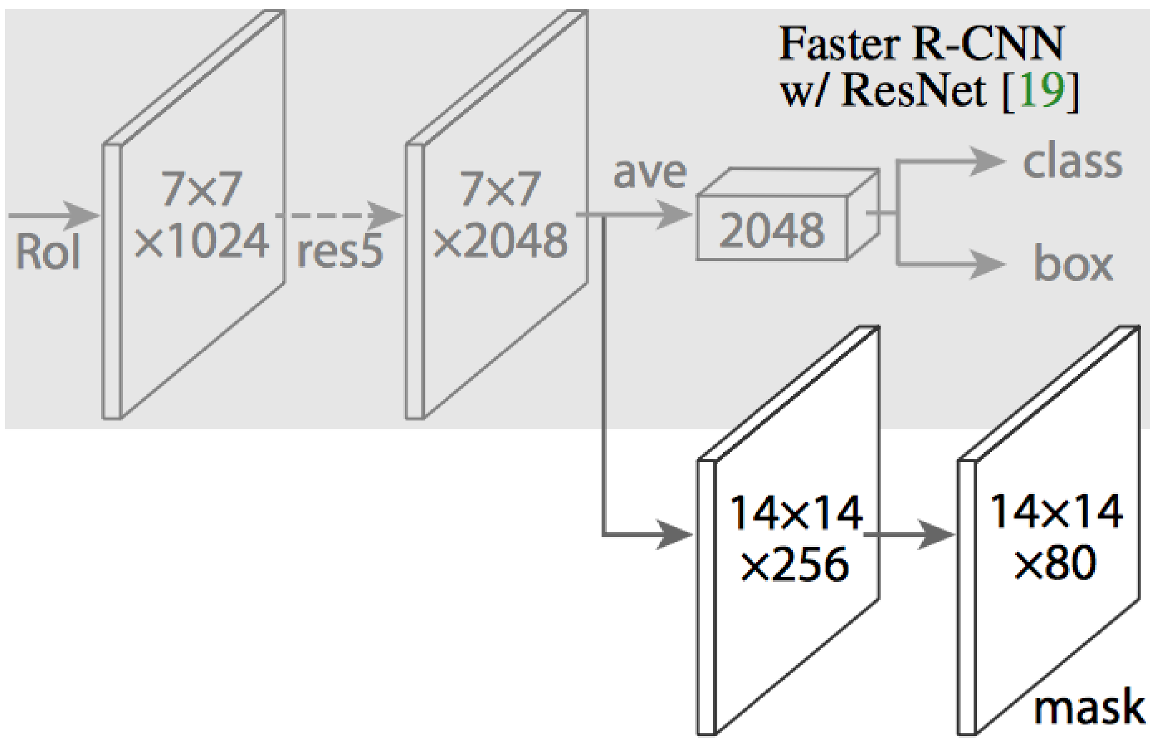
Ren et al. 2016
https://arxiv.org/abs/1506.01497

Even more! Mask R-CNN

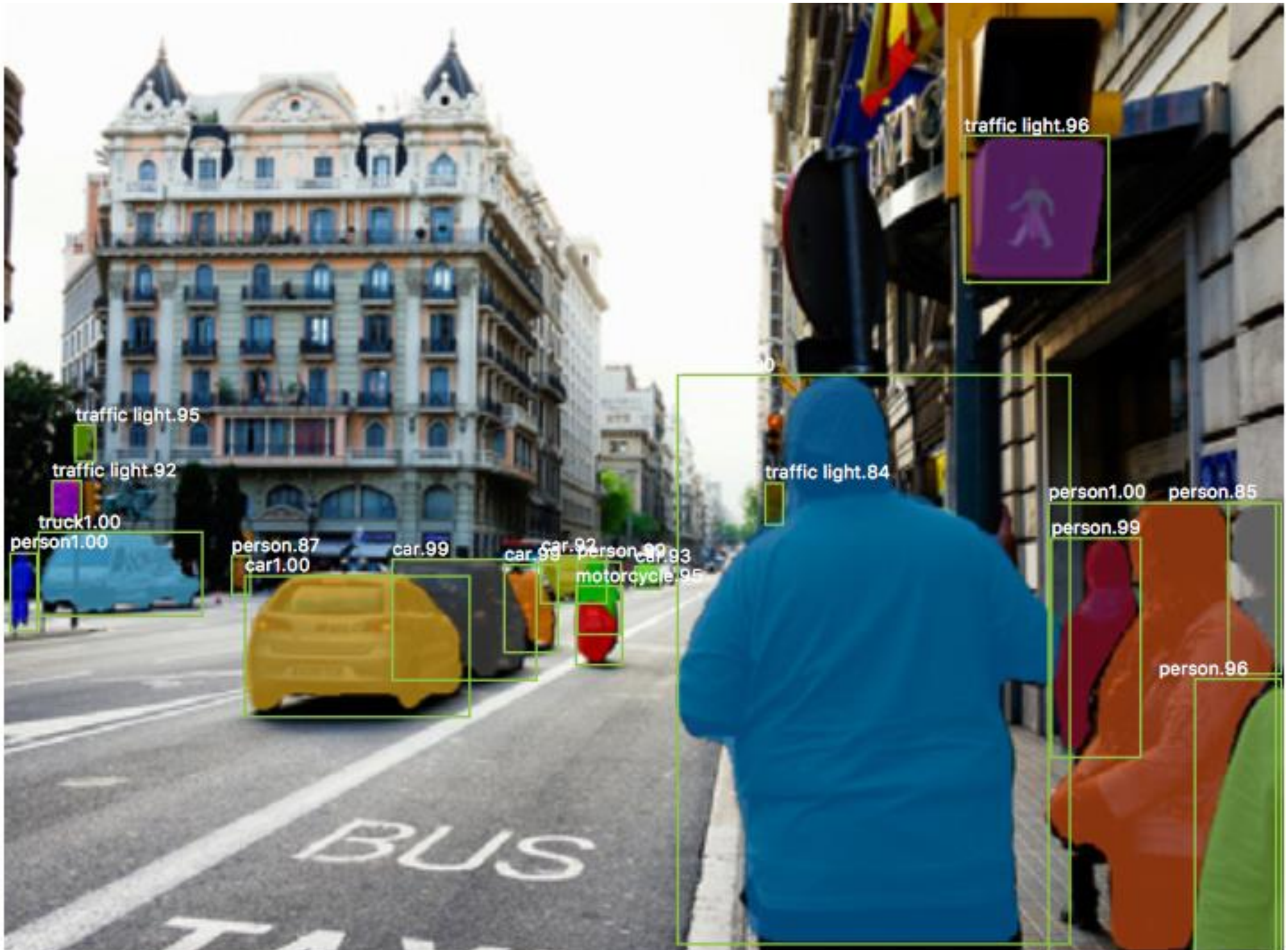
Extending Faster R-CNN for Pixel Level Segmentation

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

Add new training data: segmentation masks

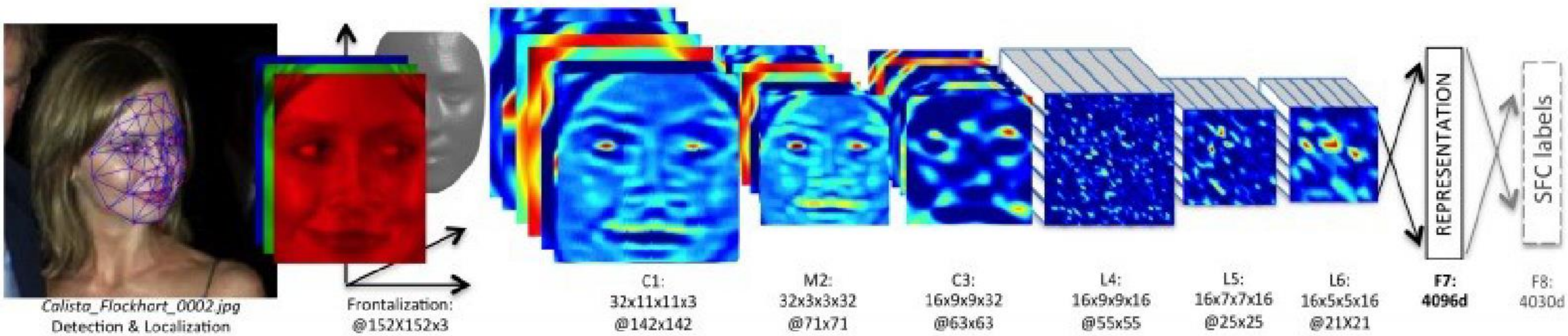


Second output which is segmentation mask


traffic light.96
traffic light.95
traffic light.92
traffic light.84
truck1.00
person1.00
person.87
car1.00
car.99
car.99
car.92
car.93
motorcycle.95
person1.00
person.85
person.99
person.96
BUS

CONV NETS: EXAMPLES

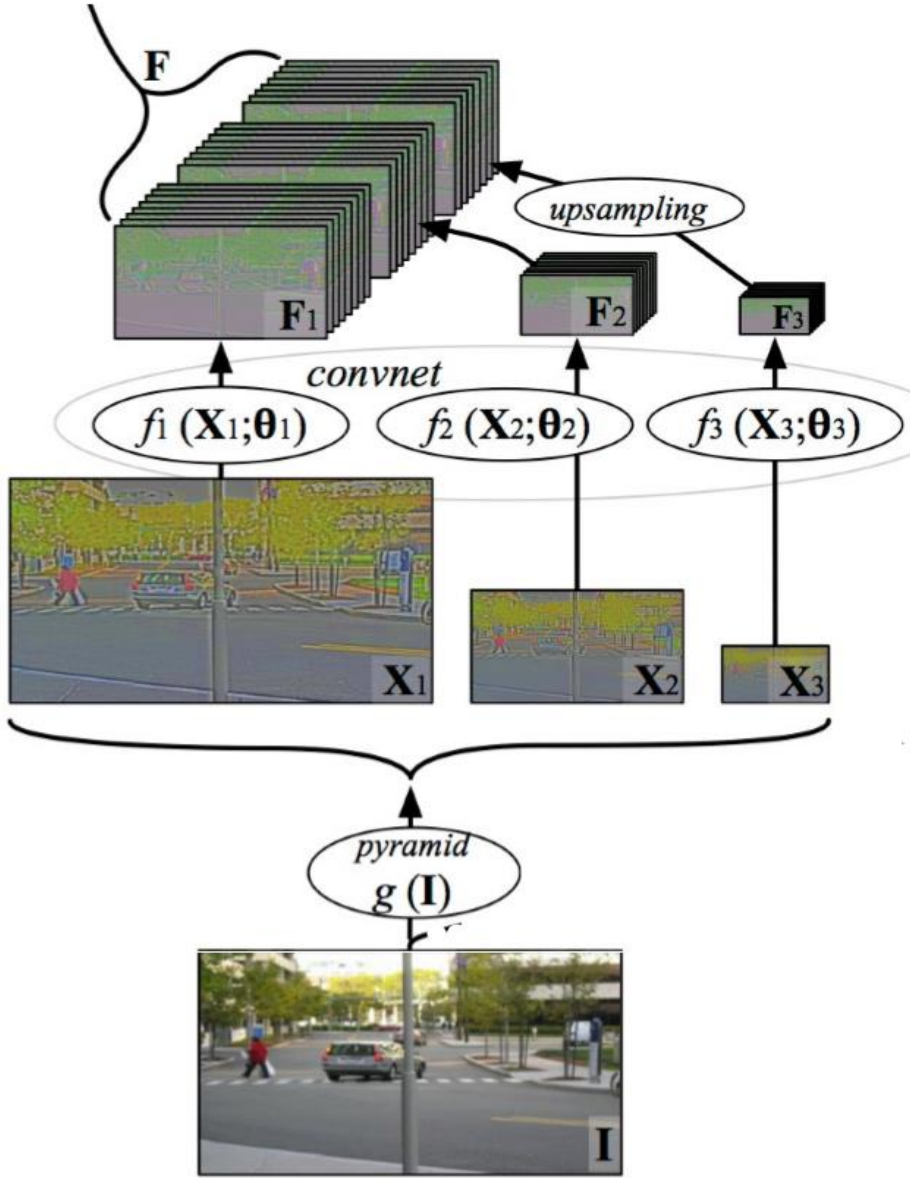
- **Face Verification & Identification**



Taigman et al. "DeepFace..." CVPR 2014

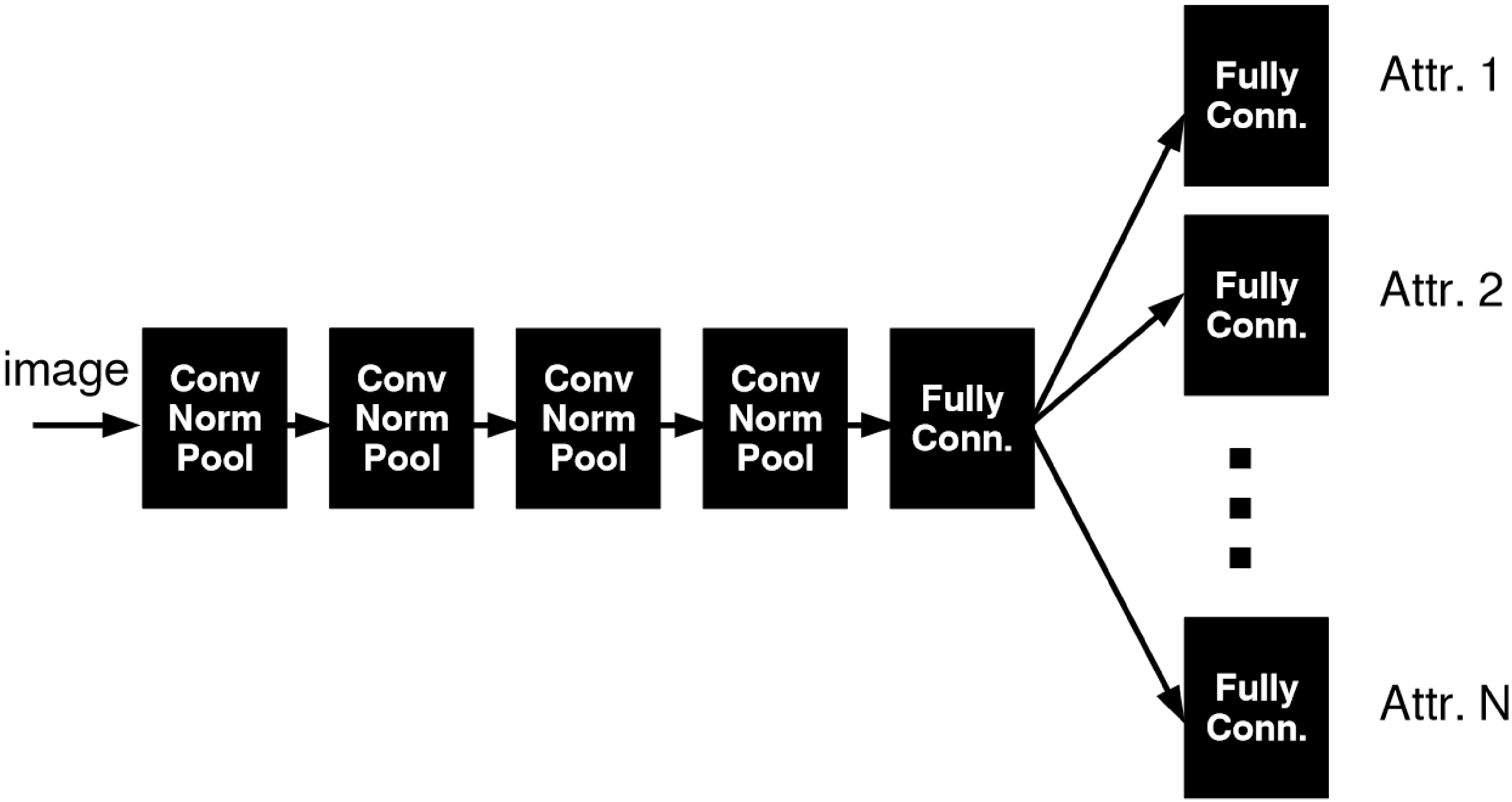
Ranzato

Fancier Architectures: Multi-Scale



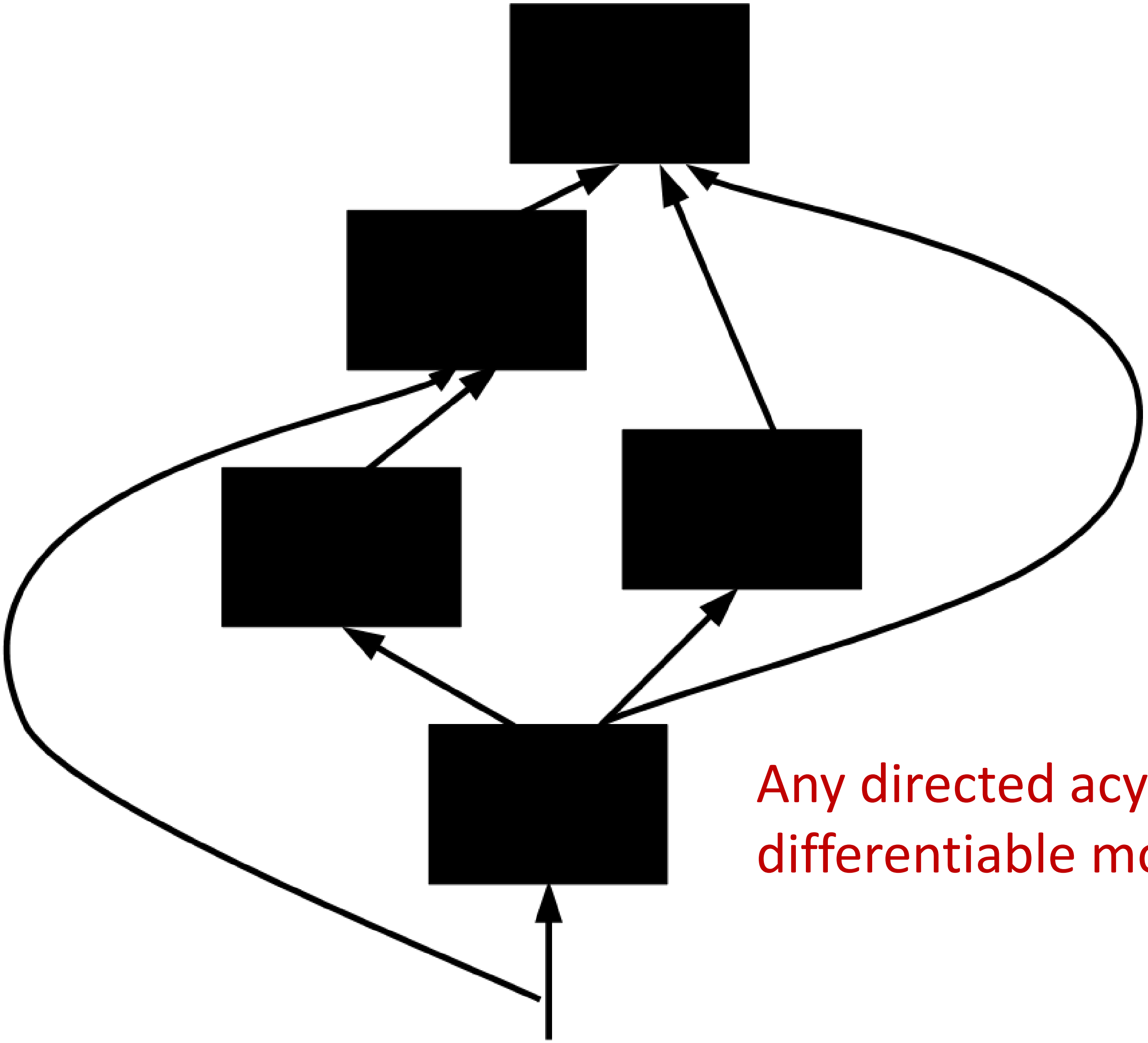
Farabet et al. "Learning hierarchical features for scene labeling" PAMI 2013

Fancier Architectures: Multi-Task



Zhang et al. “PANDA..” CVPR 2014

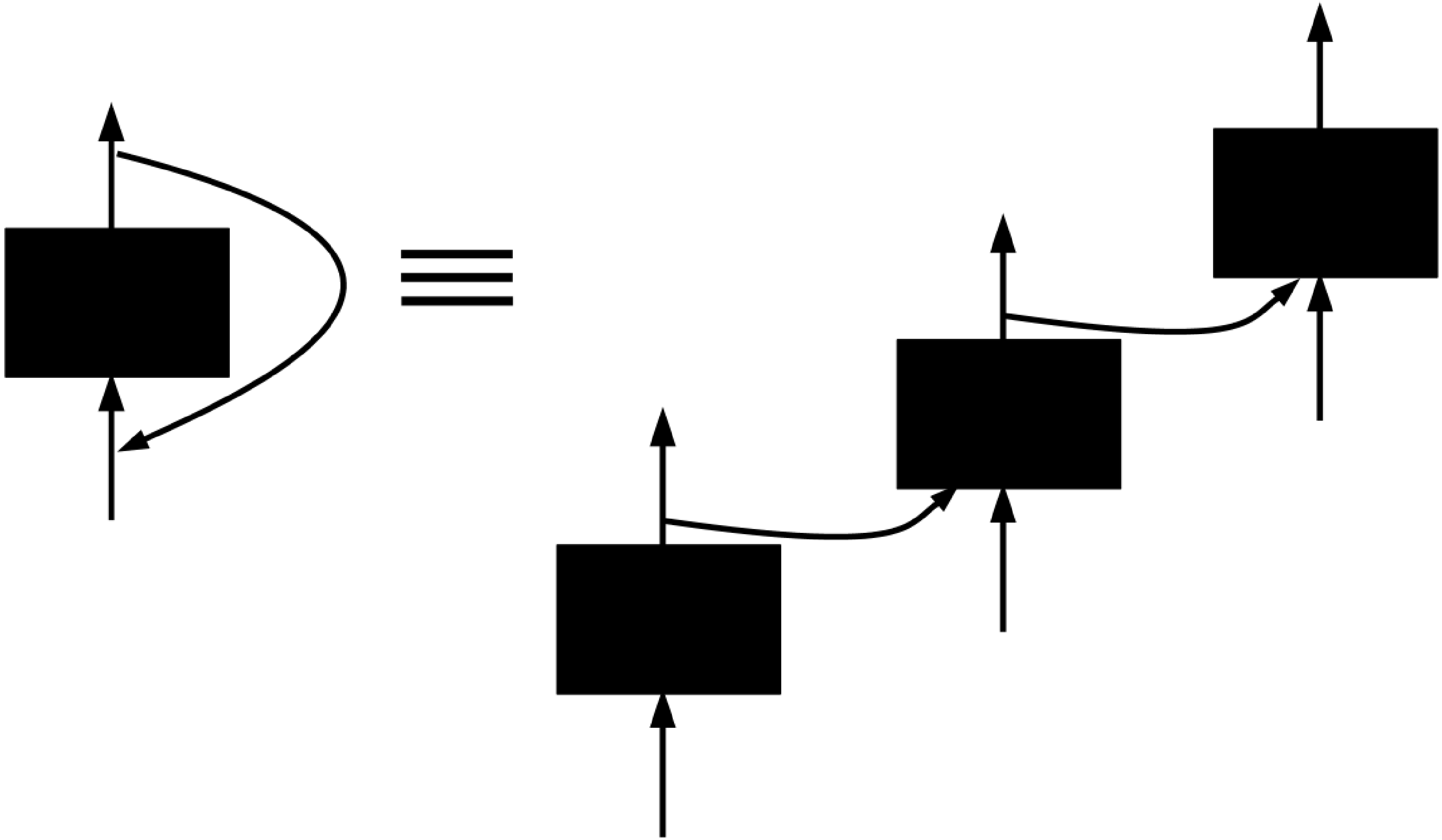
Fancier Architectures: Generic DAG



Any directed acyclic graph (DAG) of differentiable modules is allowed.

Fancier Architectures: Generic DAG

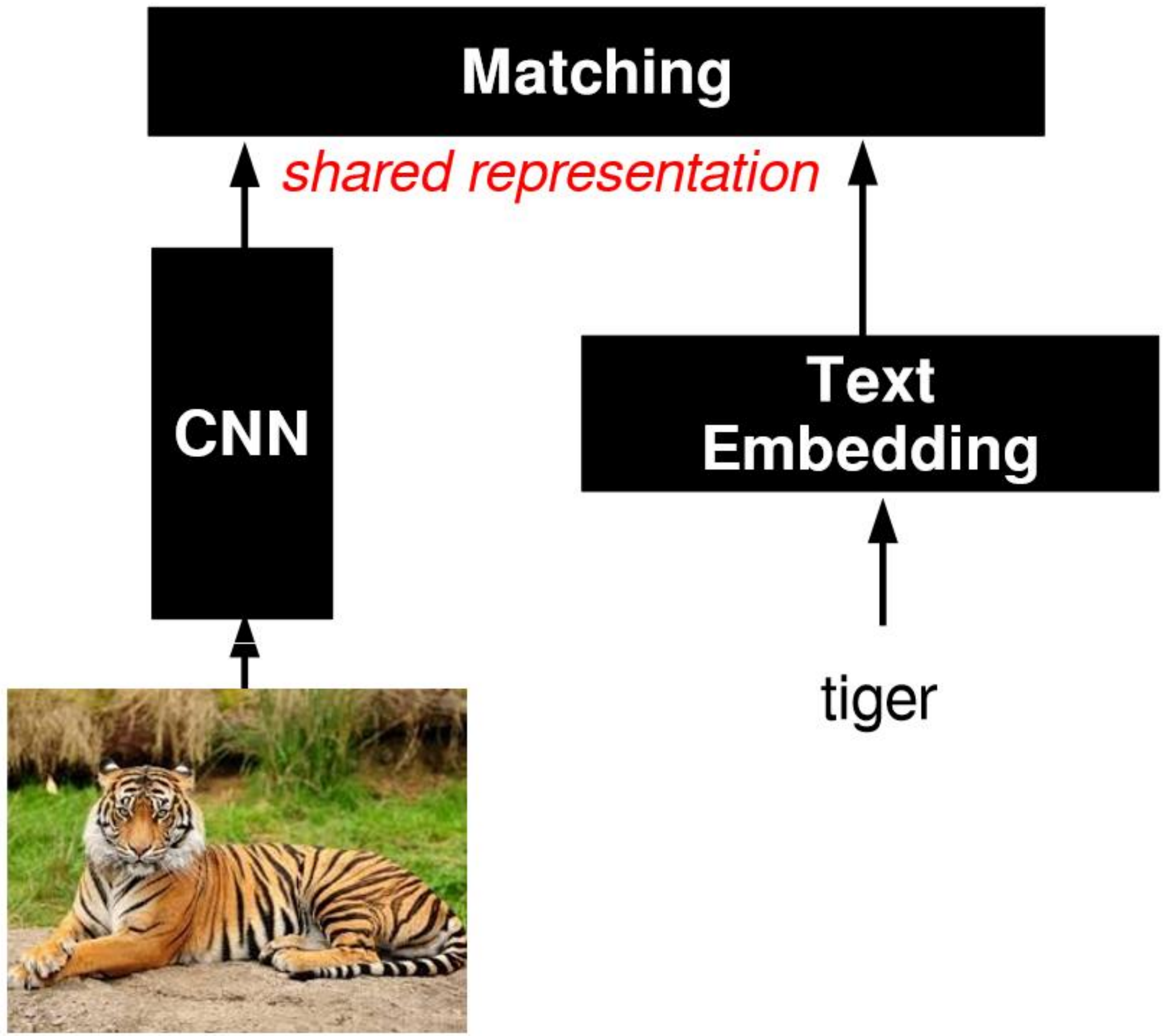
If there are cycles (RNN), one needs to un-roll it.



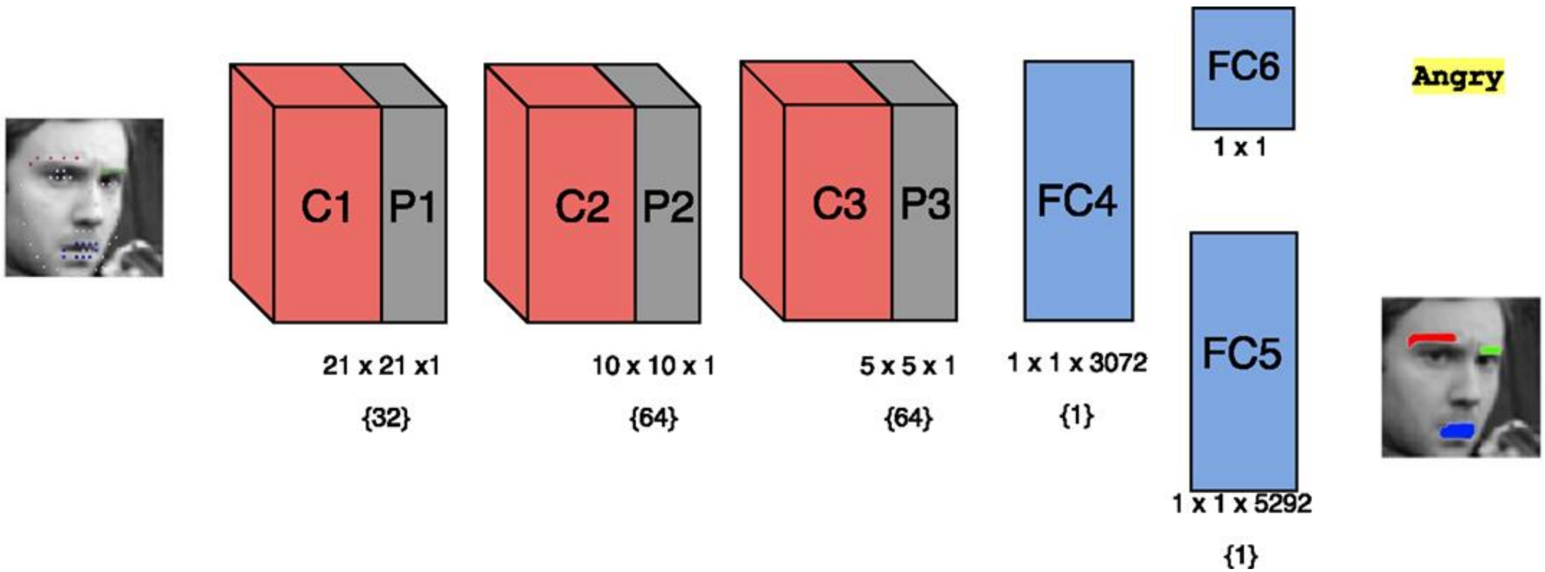
Pinheiro, Collobert “Recurrent CNN for scene labeling” ICML 2014
Graves “Offline Arabic handwriting recognition..” Springer 2012

What about learning across ‘domains’?

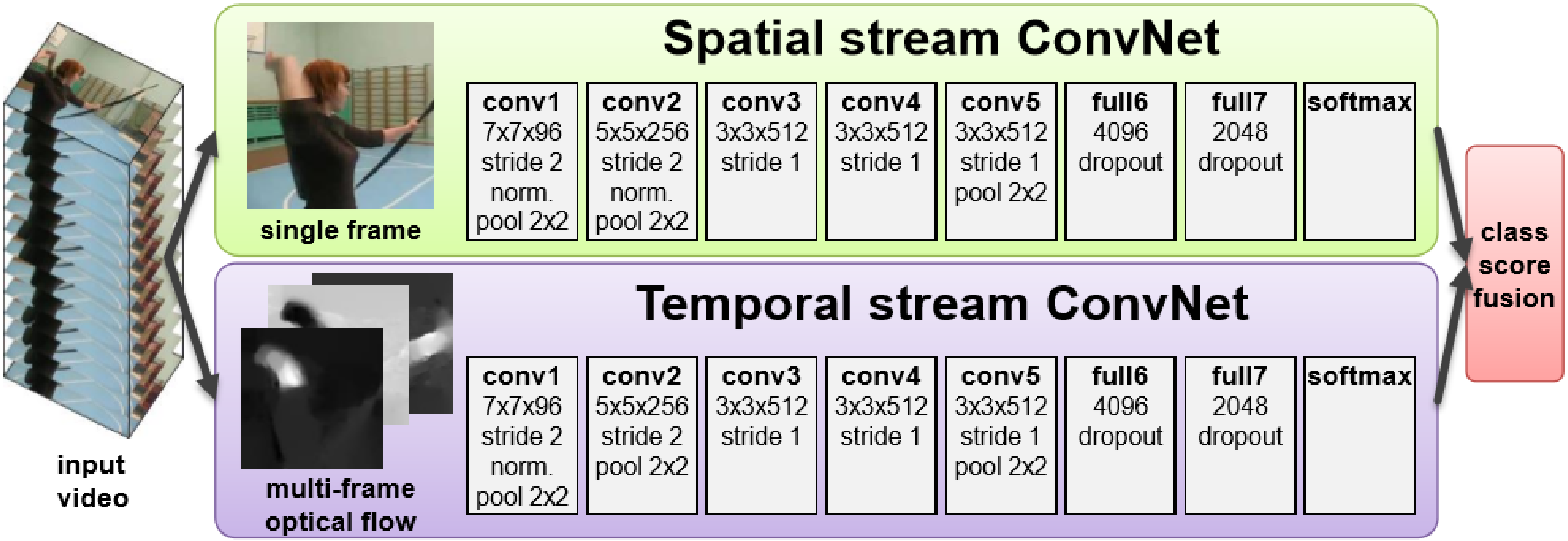
Fancier Architectures: Multi-Modal



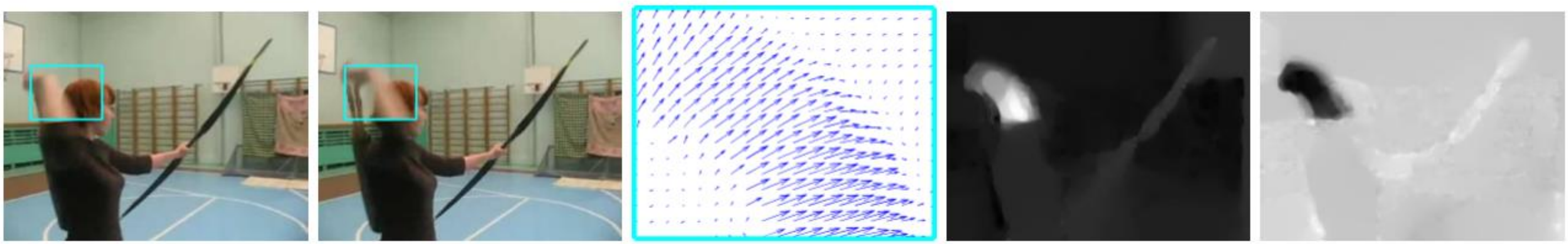
Frome et al. “Devise: a deep visual semantic embedding model” NIPS 2013


C1
P1
21 x 21 x1
{32}
C2
P2
10 x 10 x 1
{64}
C3
P3
5 x 5 x 1
{64}
FC4
1 x 1 x 3072
{1}
FC6
1 x 1
FC5
1 x 1 x 5292
{1}
Angry

Two-stream networks – *action recognition*



[Simonyan et al. 2014]



(a) (b) (c) (d) (e)

[Simonyan et al. 2014]