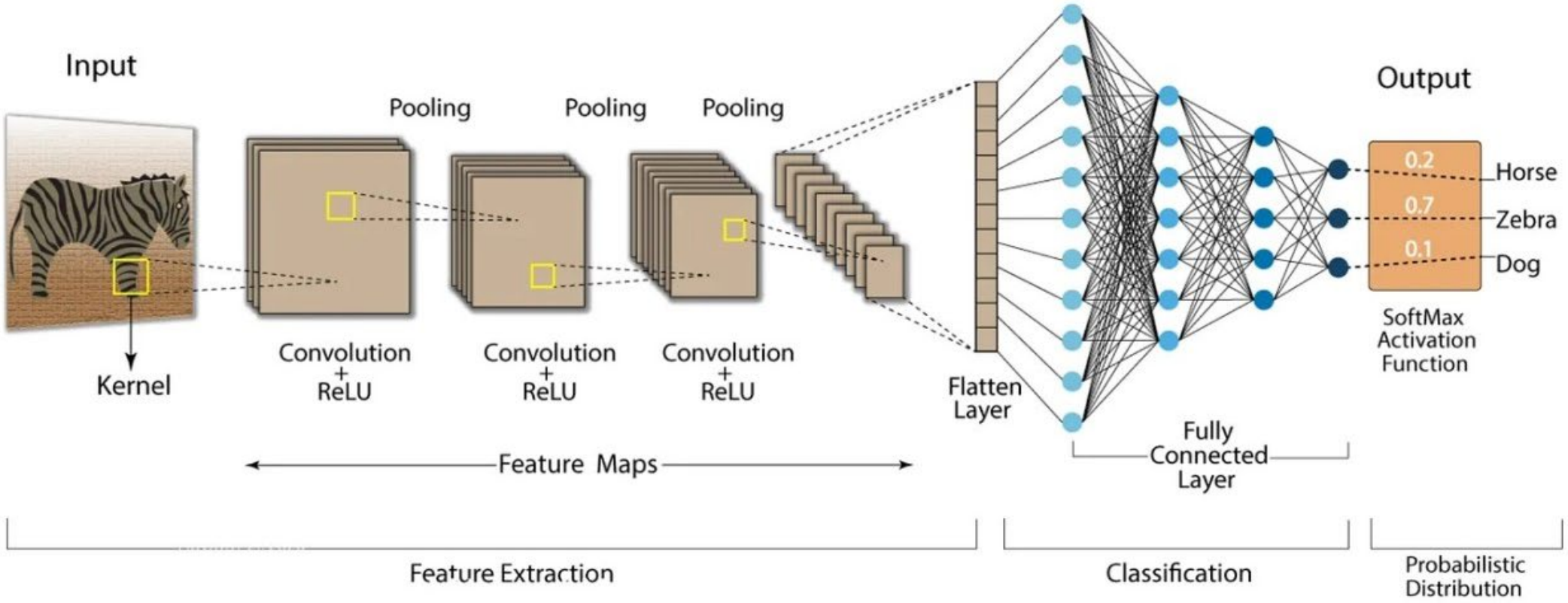
Introduction to Computer Vision

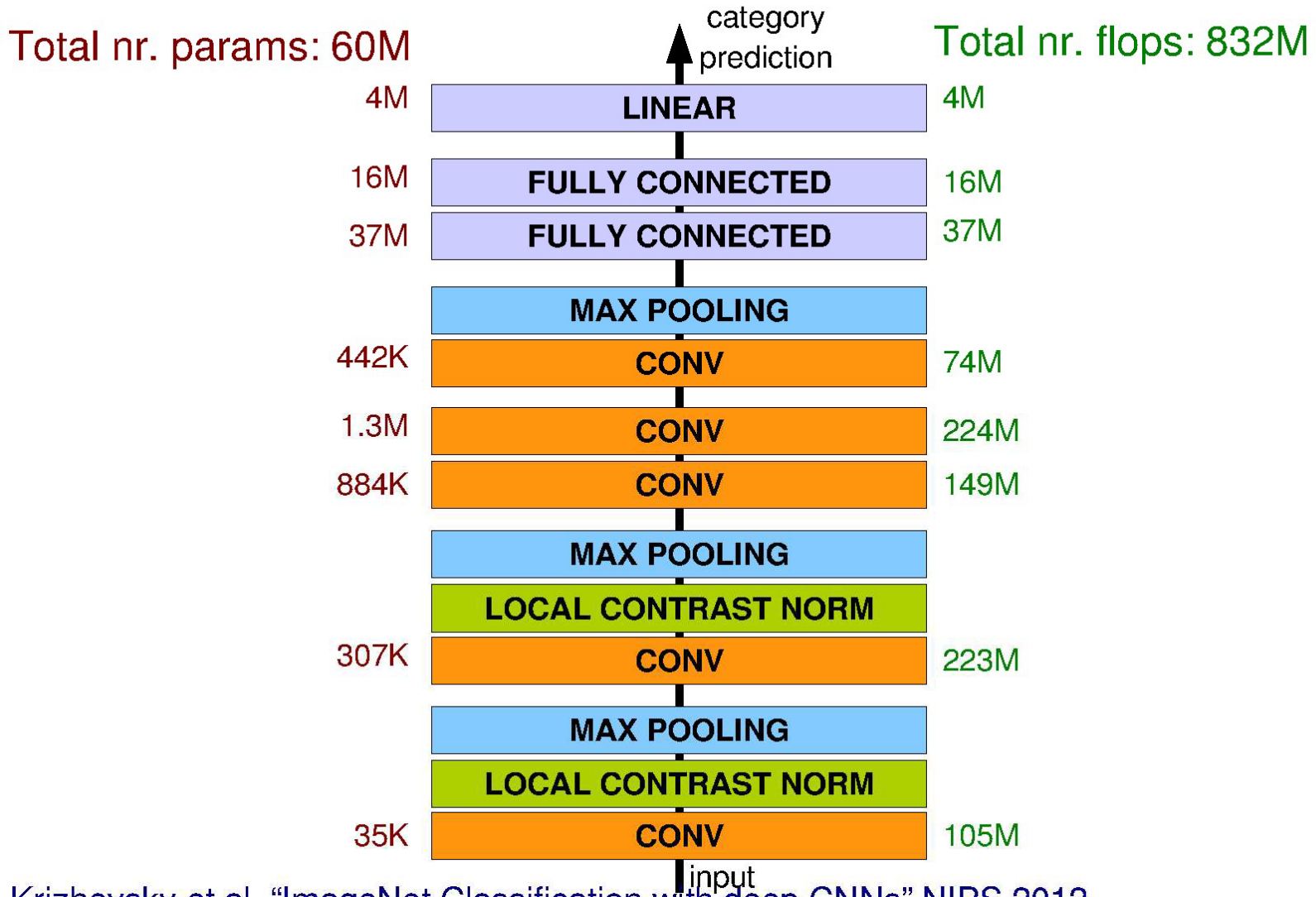
Kaveh Fathian
Assistant Professor
Computer Science Department
Colorado School of Mines

Lecture 22

CNN Architectures



Architecture for Classification



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


WE NEED TO GO
DEEPER

Beyond AlexNet

- Very Deep Convolutional Networks for Large-Scale Image Recognition
  - Karen Simonyan & Andrew Zisserman 2015

- These are pre-trained "VGG" networks

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					

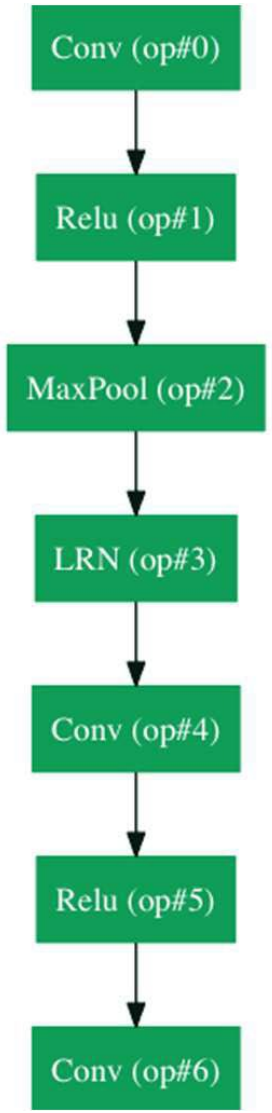
Table 2: **Number of parameters** (in millions).

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

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

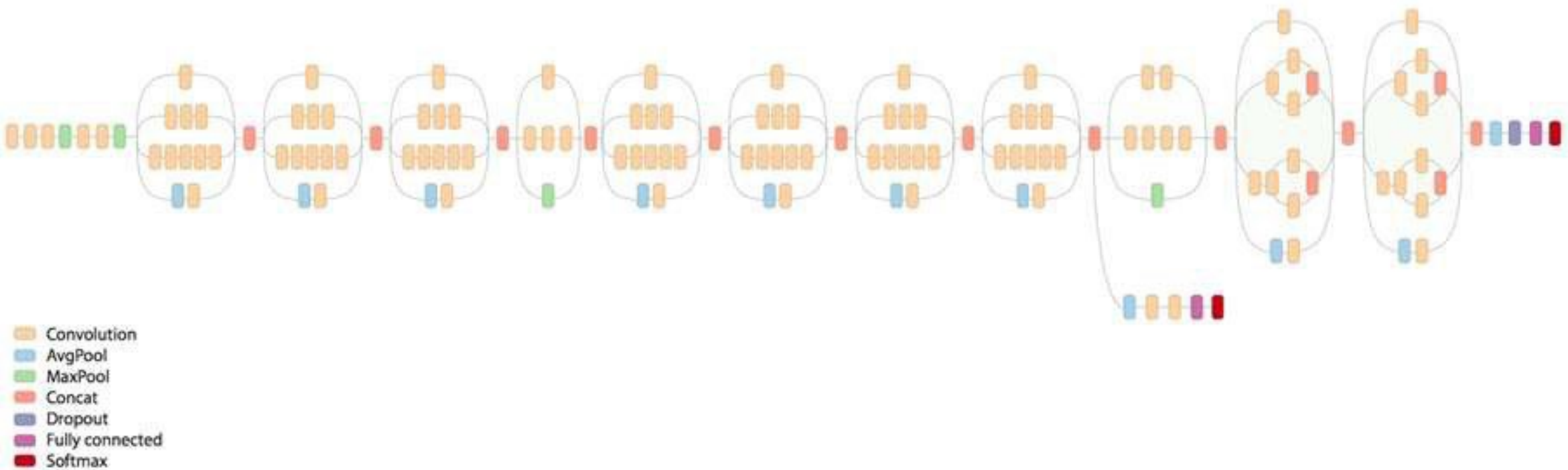
ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
B	256	224,256,288	28.2	9.6
C	256	224,256,288	27.7	9.2
	384	352,384,416	27.8	9.2
	[256; 512]	256,384,512	26.3	8.2
D	256	224,256,288	26.6	8.6
	384	352,384,416	26.5	8.6
	[256; 512]	256,384,512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



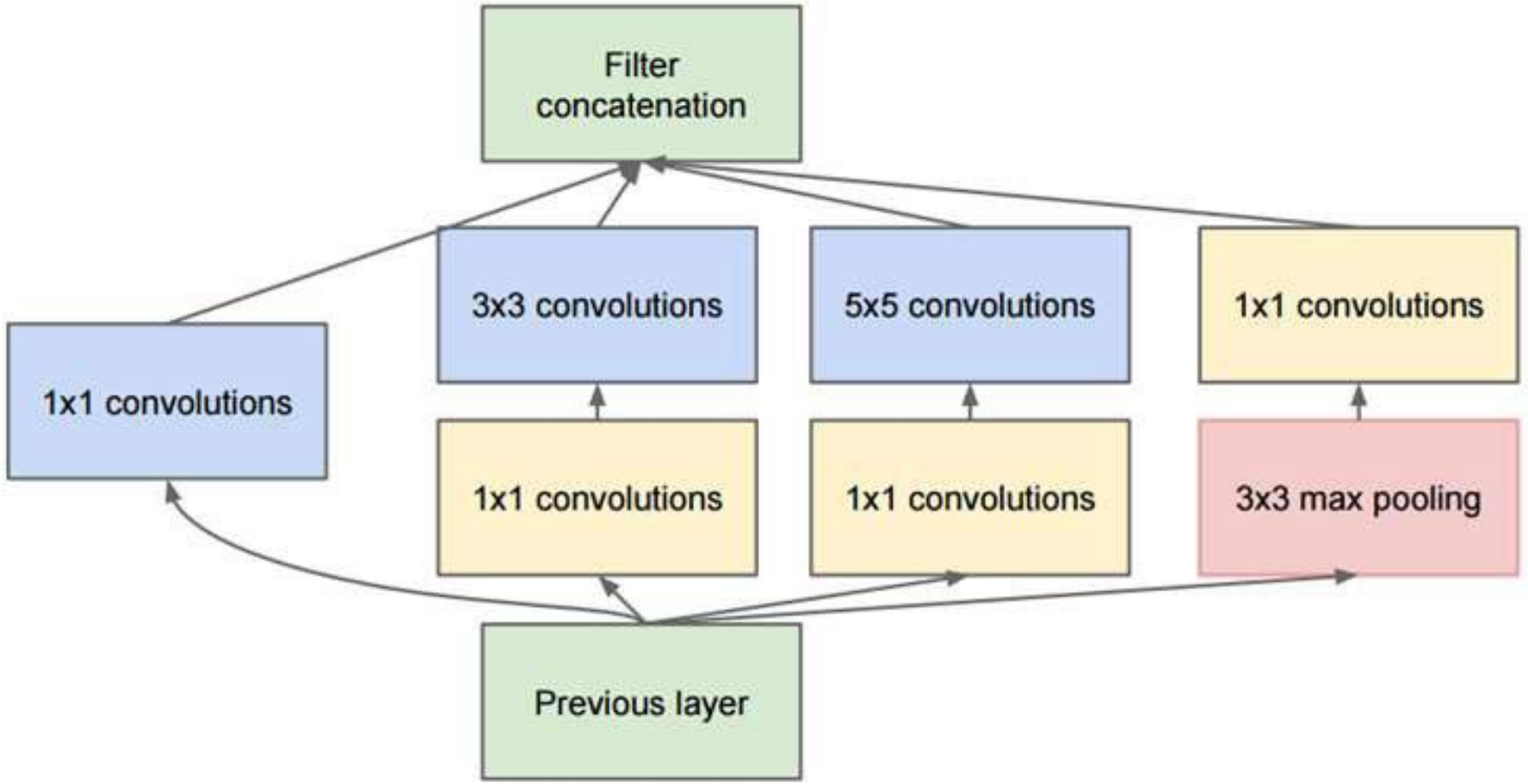
22 layers

6.67% error
ImageNet top 5



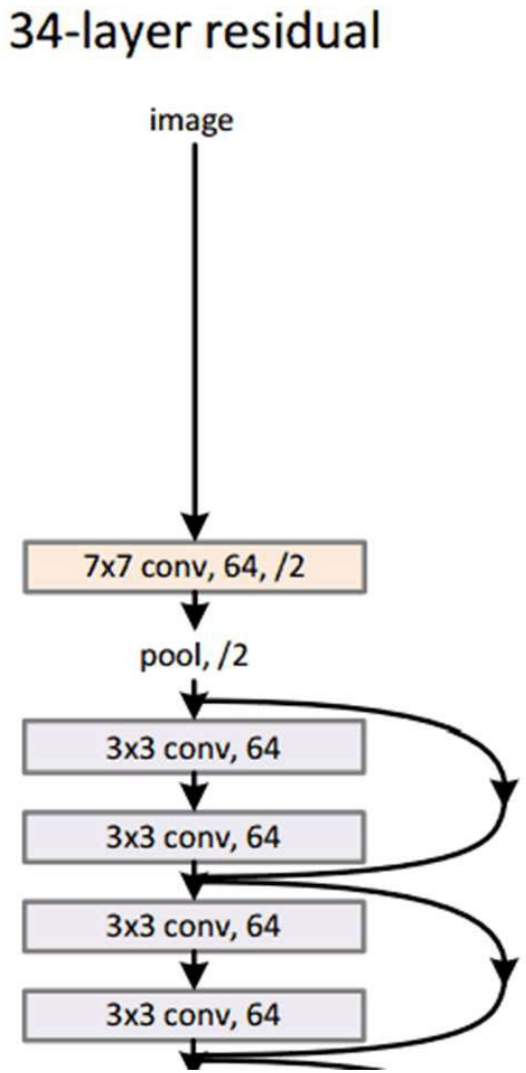
Another view of GoogLeNet's architecture.

Parallel layers



Full Inception module

ResNet (He et al., 2015)

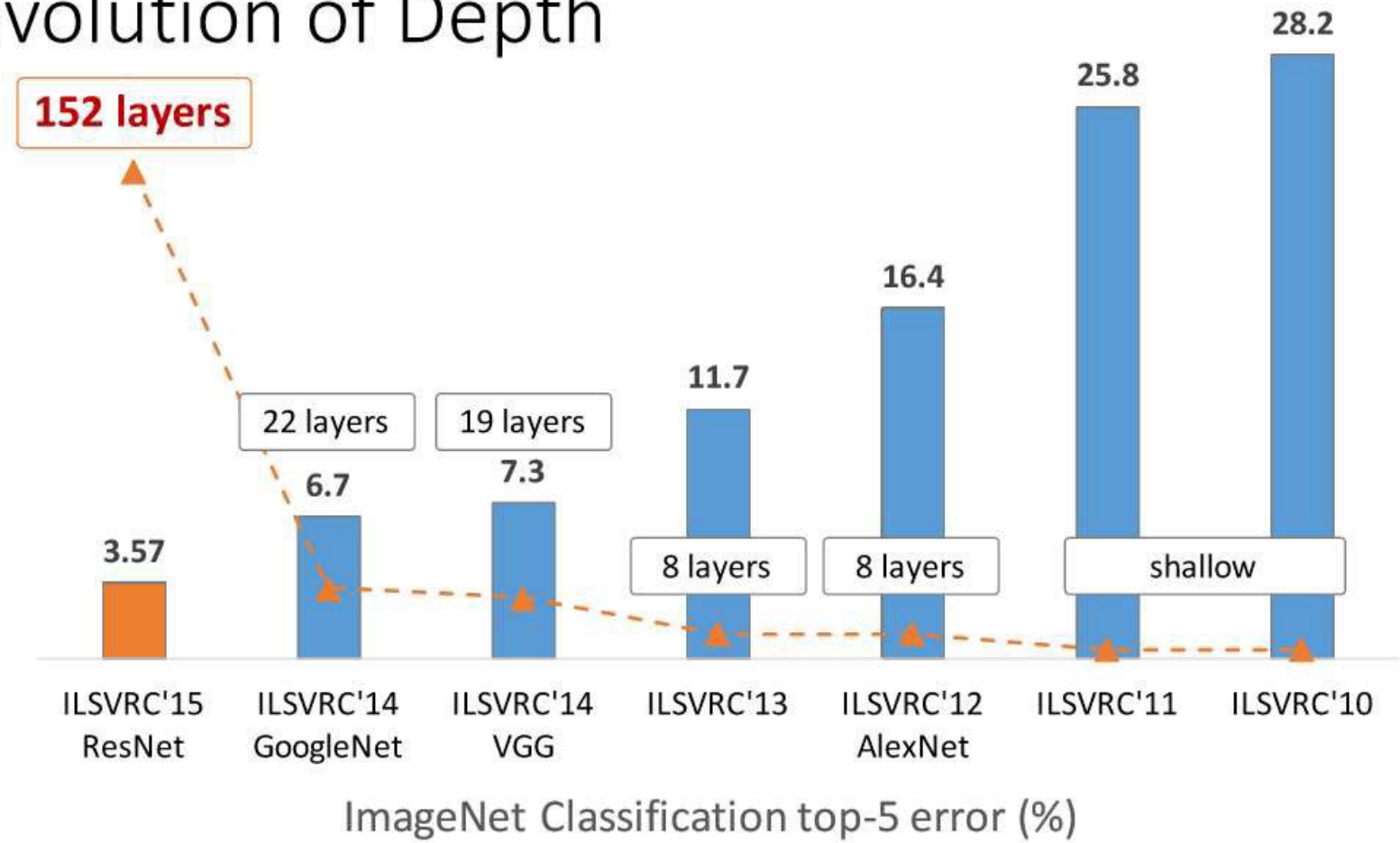


ResNet won ILSVRC 2015 with a top-5 error rate of 3.6%

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



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

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

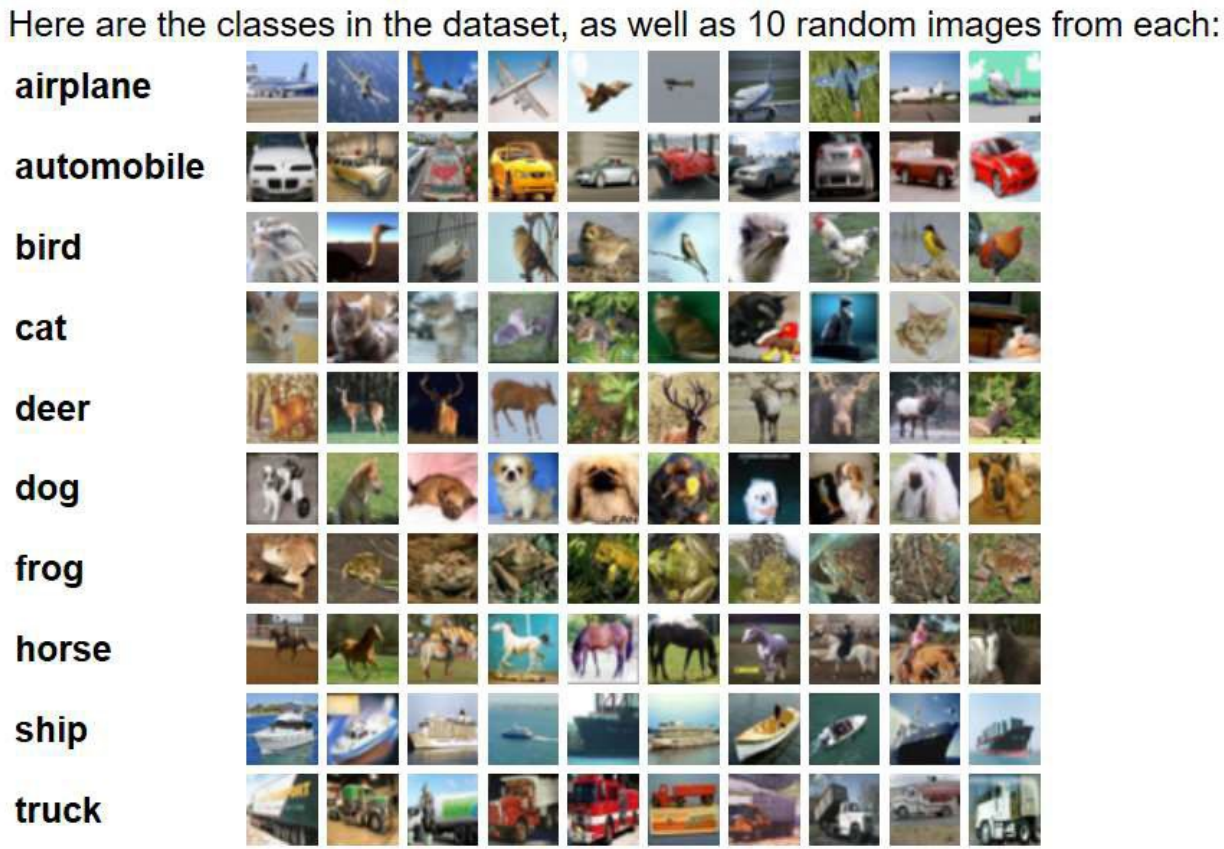
VGG, 19 layers
(ILSVRC 2014)

ResNet, 152 layers
(ILSVRC 2015)

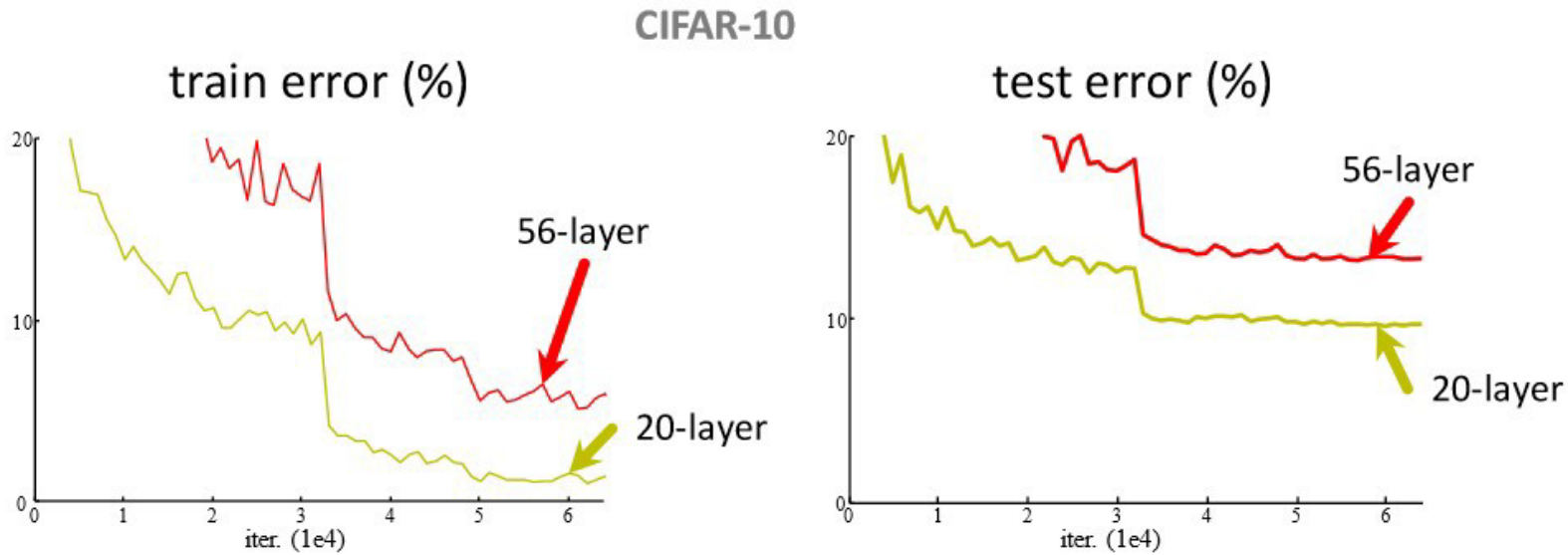
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

CIFAR-10

60,000 32x32 color images, 10 classes



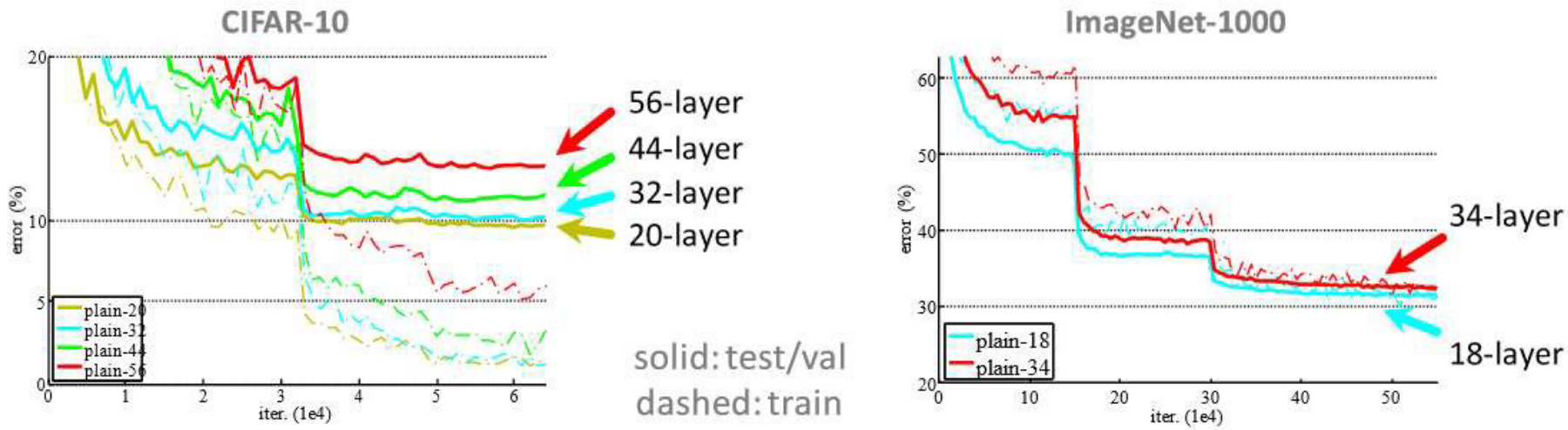
Simply stacking layers?



- *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

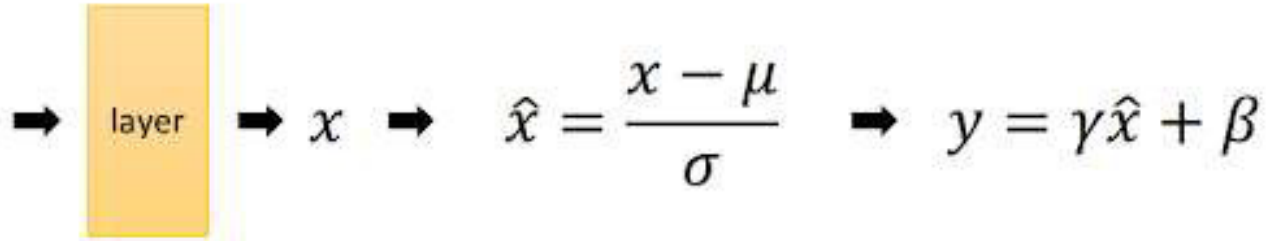
Backpropagation:

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

Chaining partial derivates over many layers is unstable!

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

Batch normalization



- μ: mean of x in mini-batch
- σ: std of x in mini-batch
- γ: scale
- β: shift

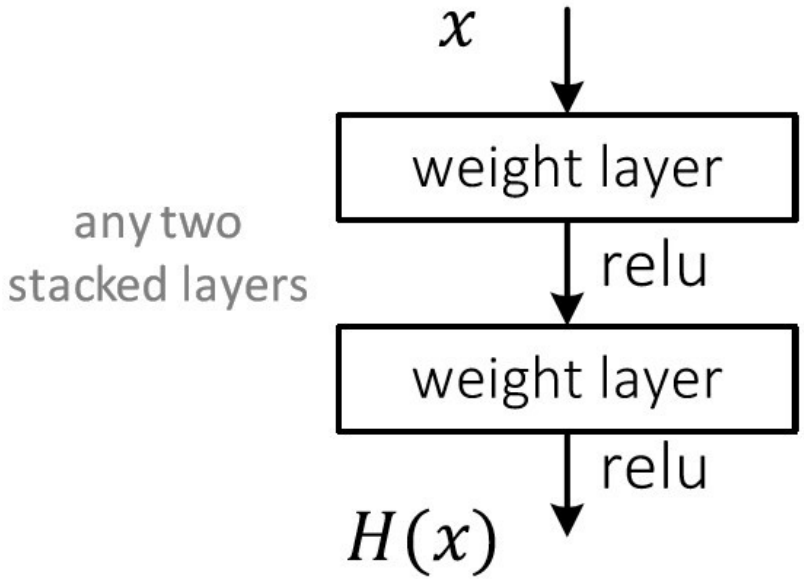
- μ, σ: functions of x, analogous to responses
- γ, β: parameters to be learned, analogous to weights

Rescales inputs so that gradients are well behaved

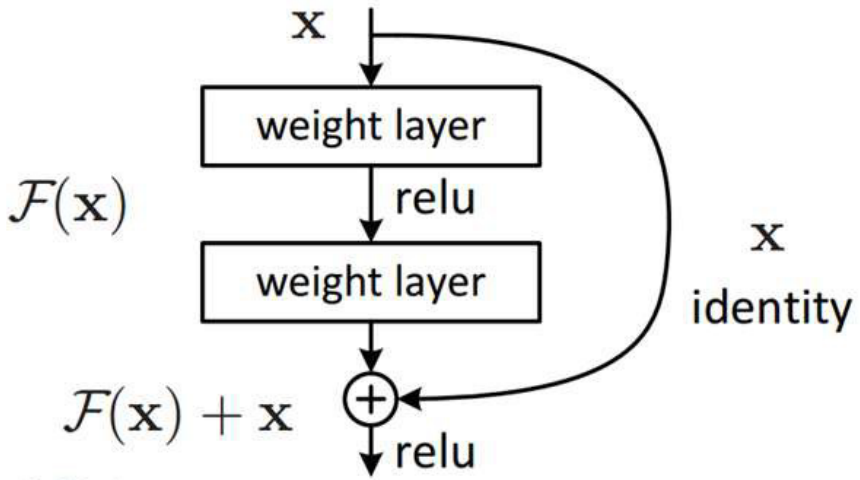
Regular net

$H(x)$ is any desired mapping,

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



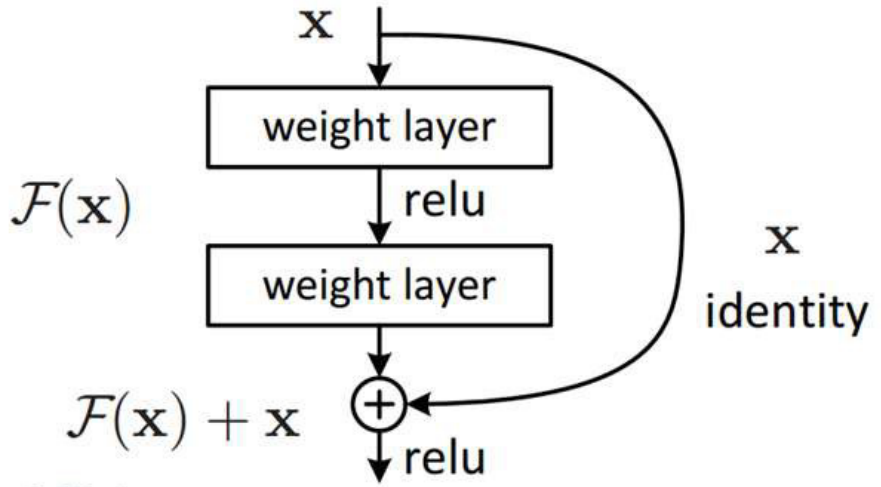
Residual Unit



A residual block

Residual Unit

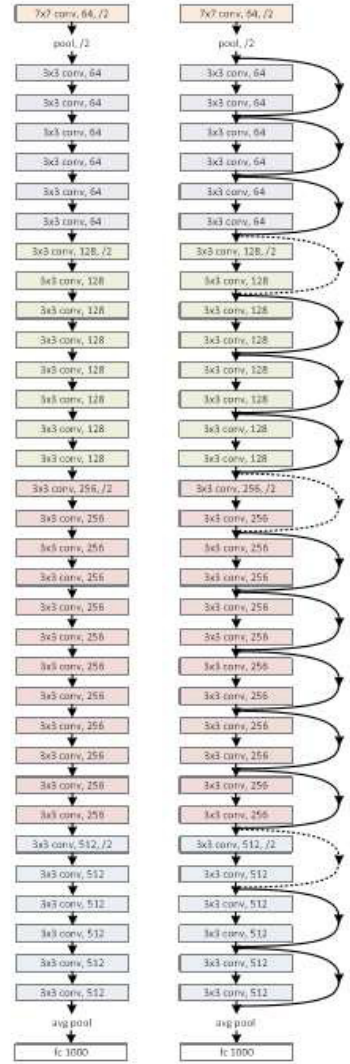
The inputs of a lower layer is added to a node in a higher layer.



A residual block

Network “Design”

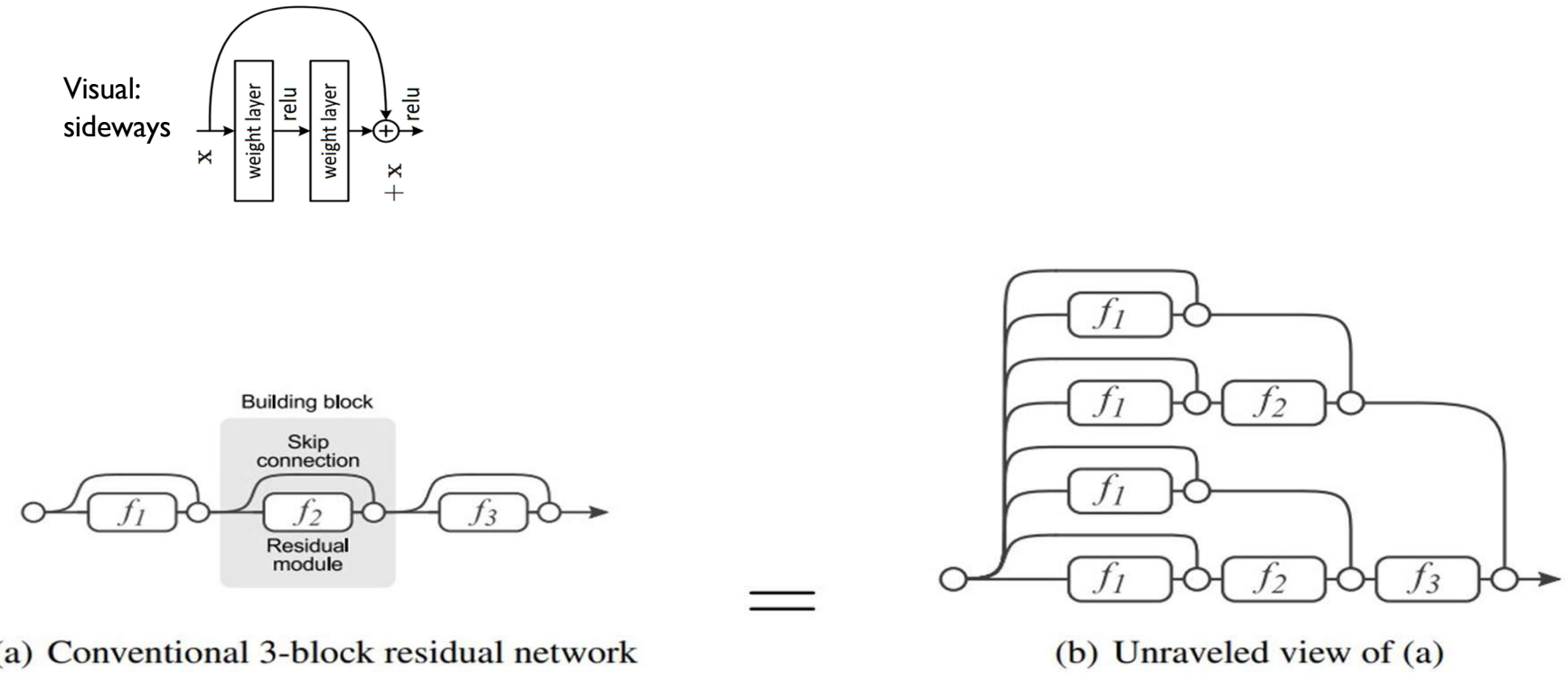
plain net



ResNet

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

Residual Networks Behave Like Ensembles of Relatively Shallow Networks



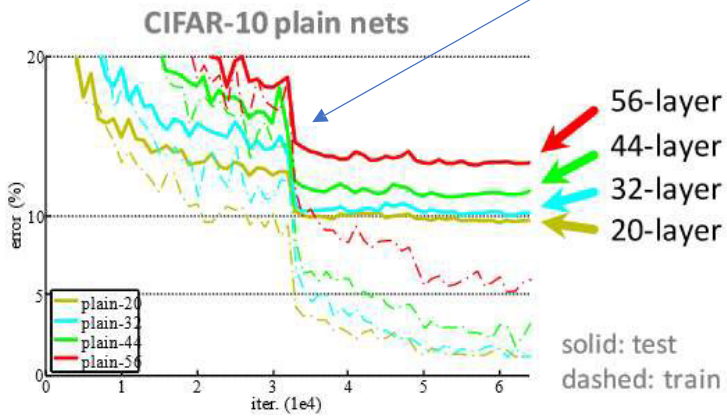
(a) Conventional 3-block residual network

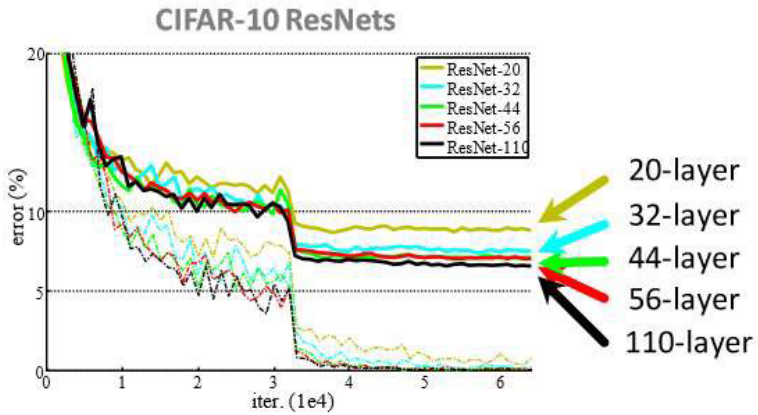
(b) Unraveled view of (a)

Why so steep?

Training rate change – lower allows finer exploration of narrow 'valleys' in energy landscape.

CIFAR-10 experiments

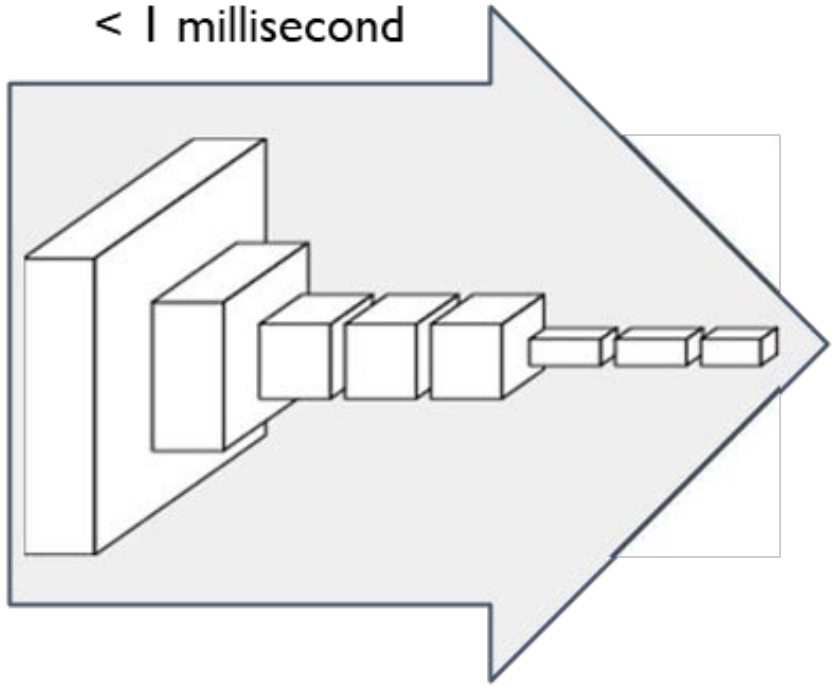




- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

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

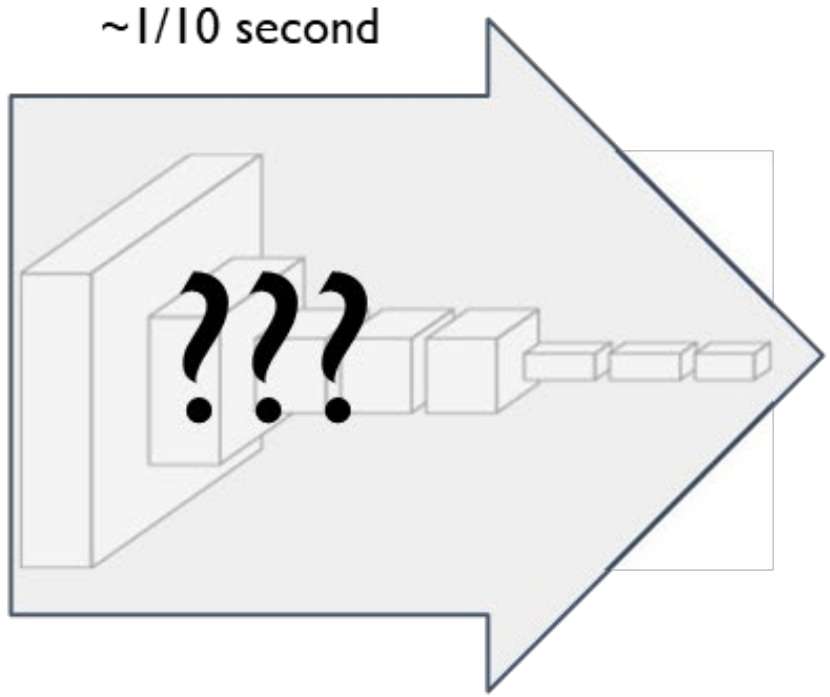
CNNs Perform Classification

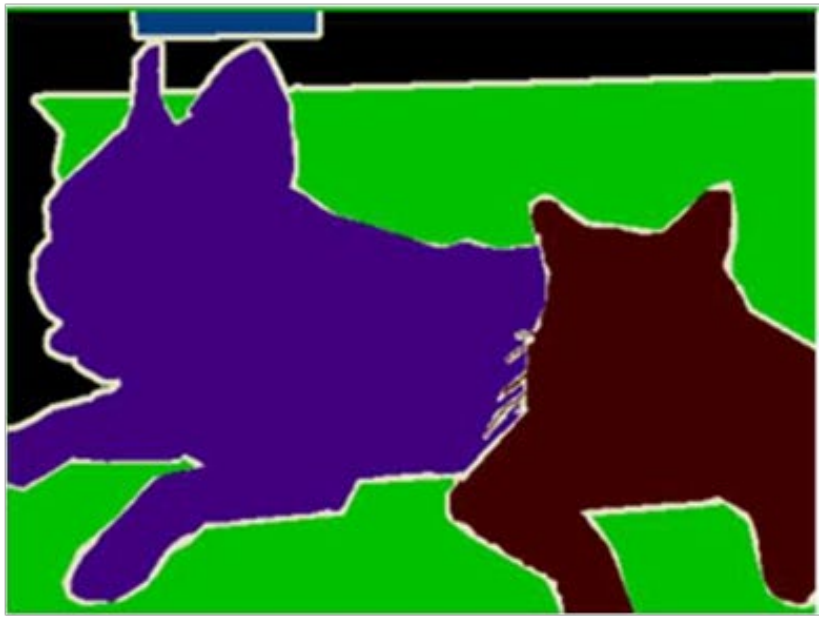


"cat"

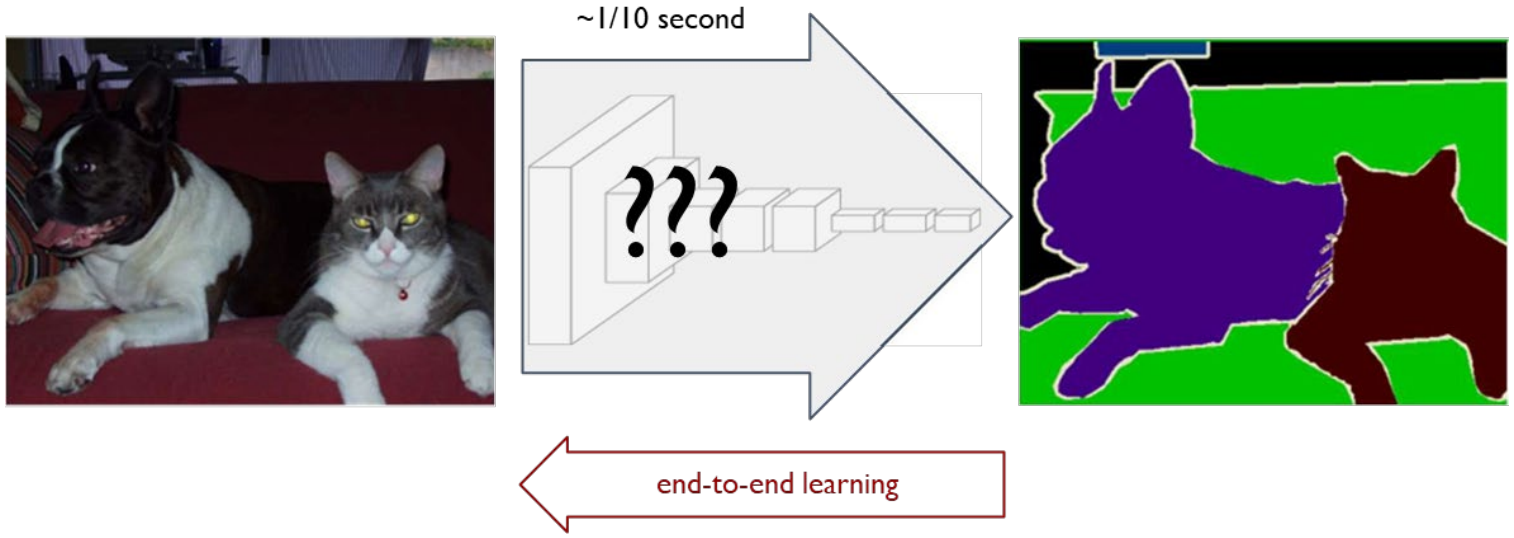
1000-dim vector

How about Semantic Segmentation?





Semantic Segmentation Problem

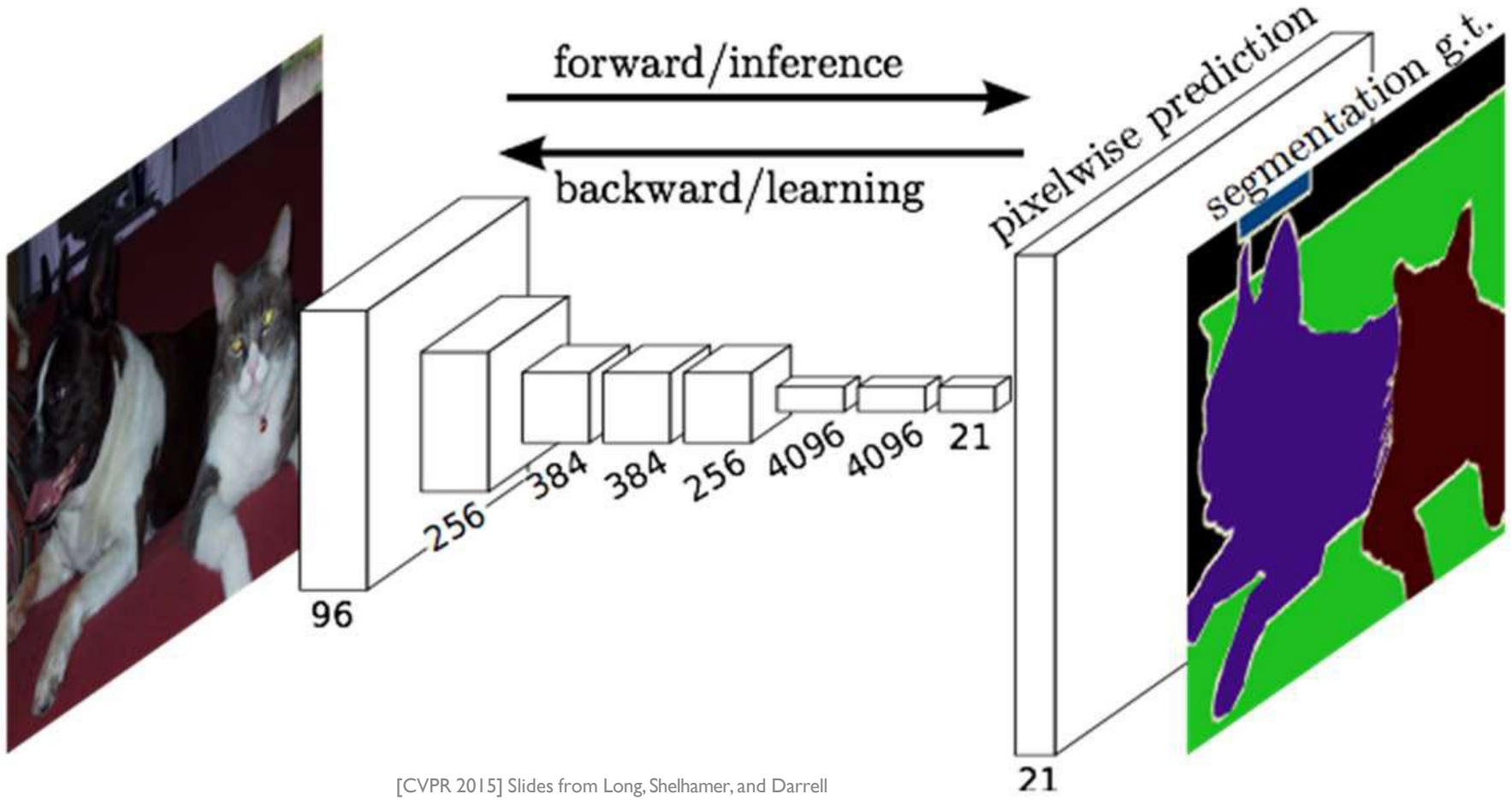


- We want a label at **every pixel**
- Current network gives us a label for the whole image

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.

Fully Convolutional Networks for Semantic Segmentation



[CVPR 2015] Slides from Long, Shelhamer, and Darrell

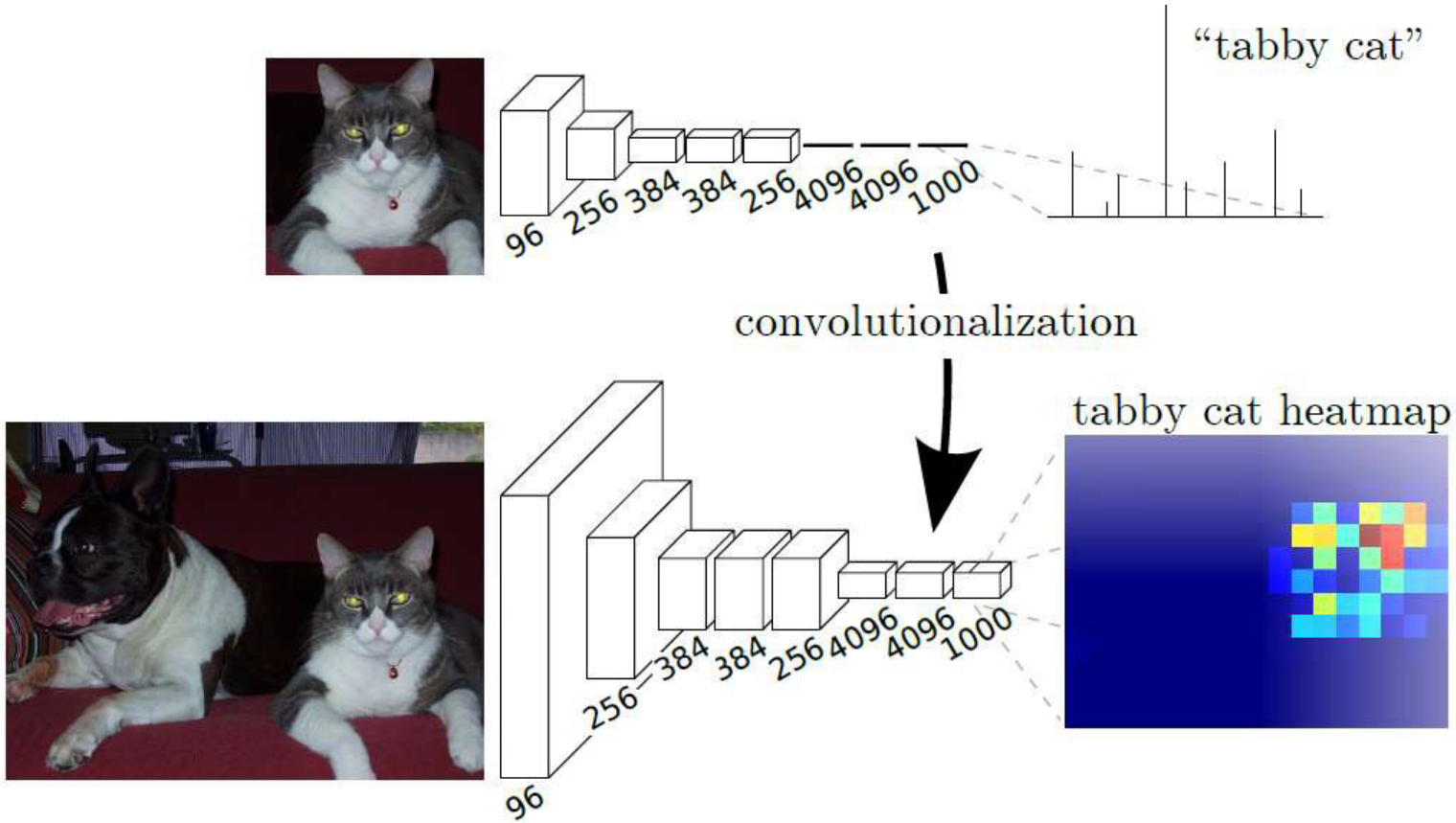
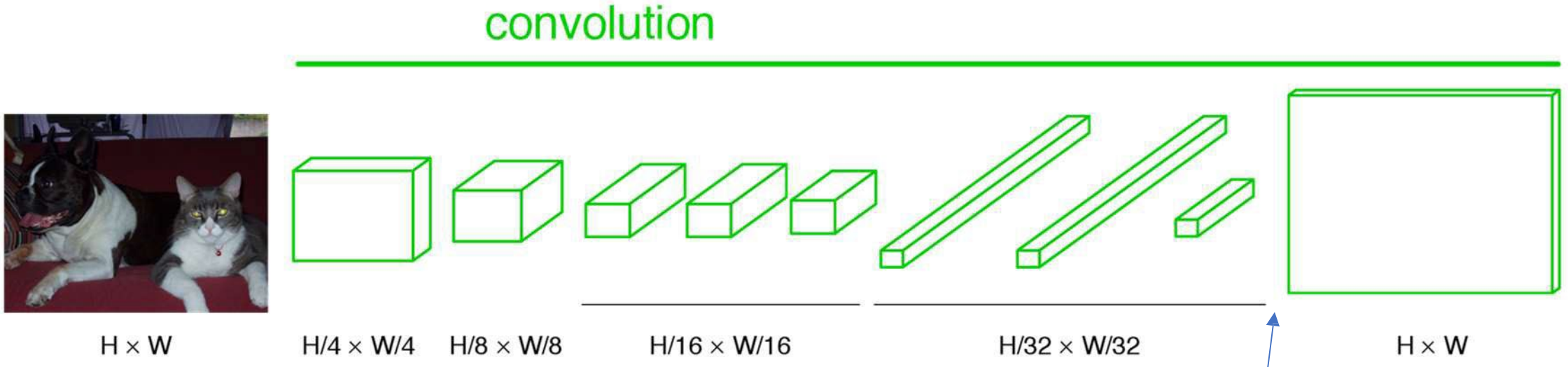


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

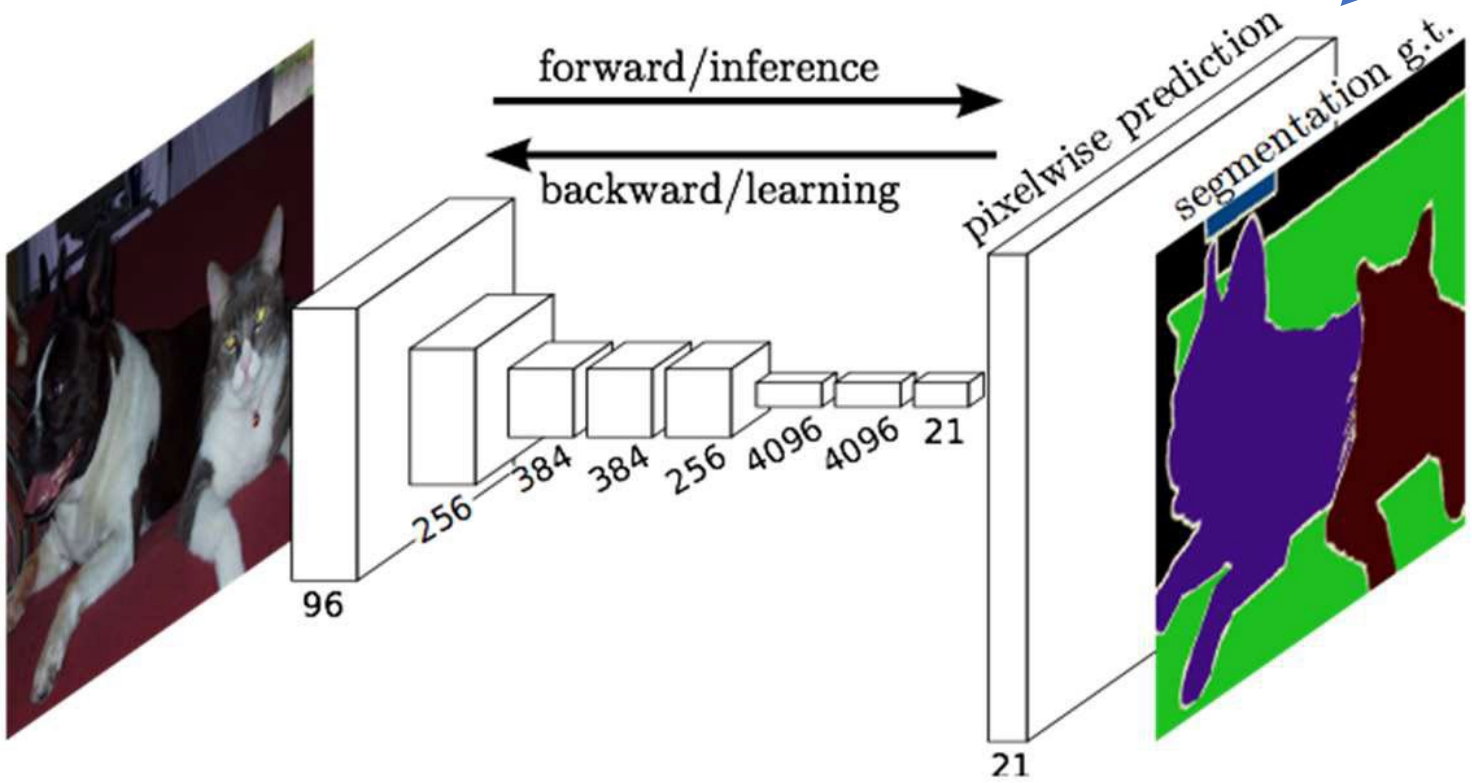
Upsampling the Output



Some upsampling algorithm to return us to H x W

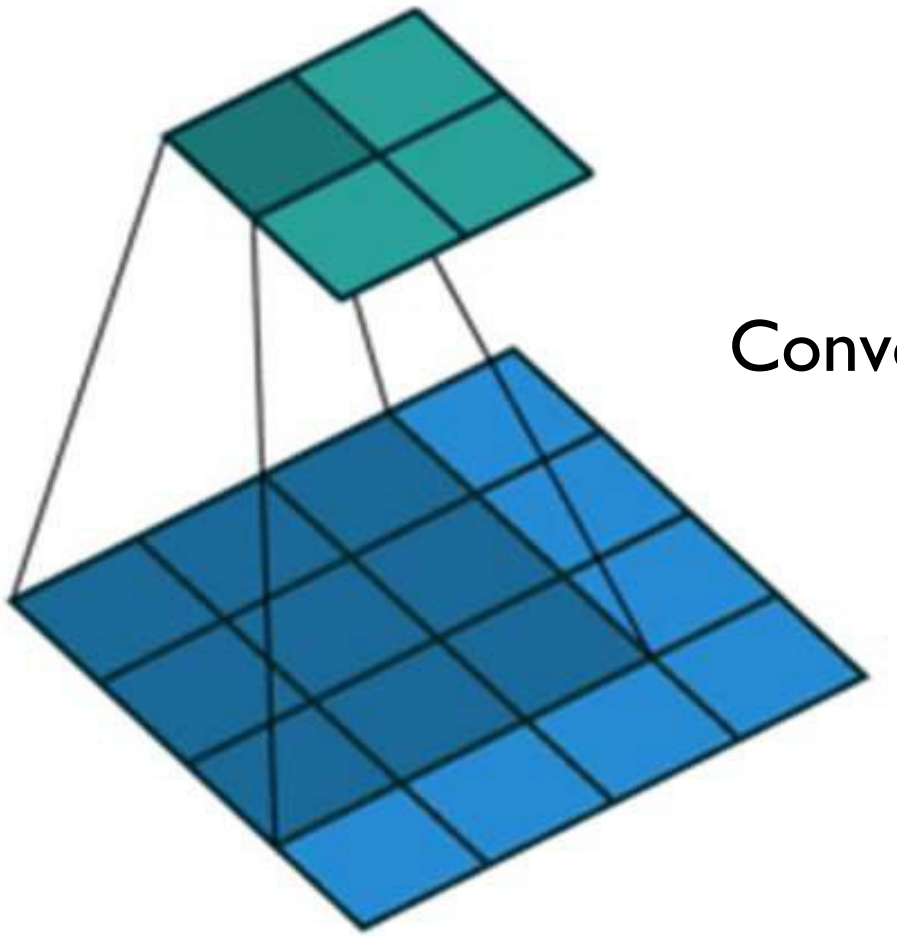
What is the upsampling layer?

This one



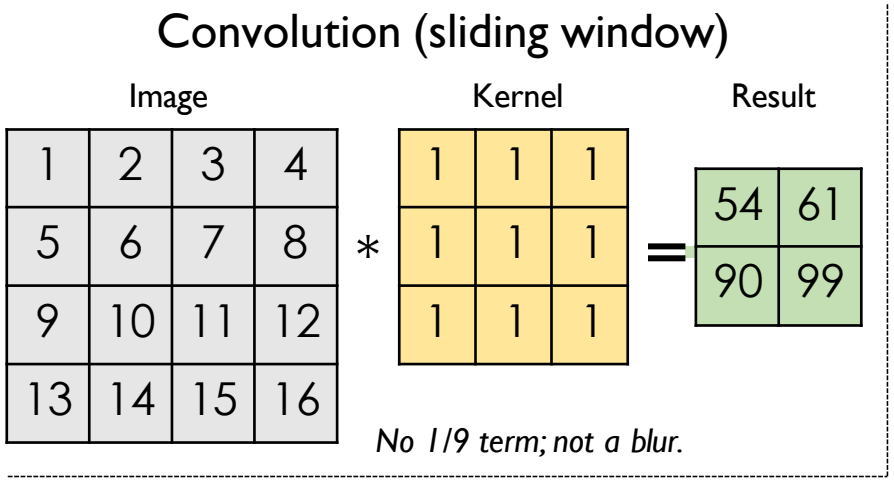
Note: sometimes we use an upsampling ***network***

Upsampling with Transposed Convolution

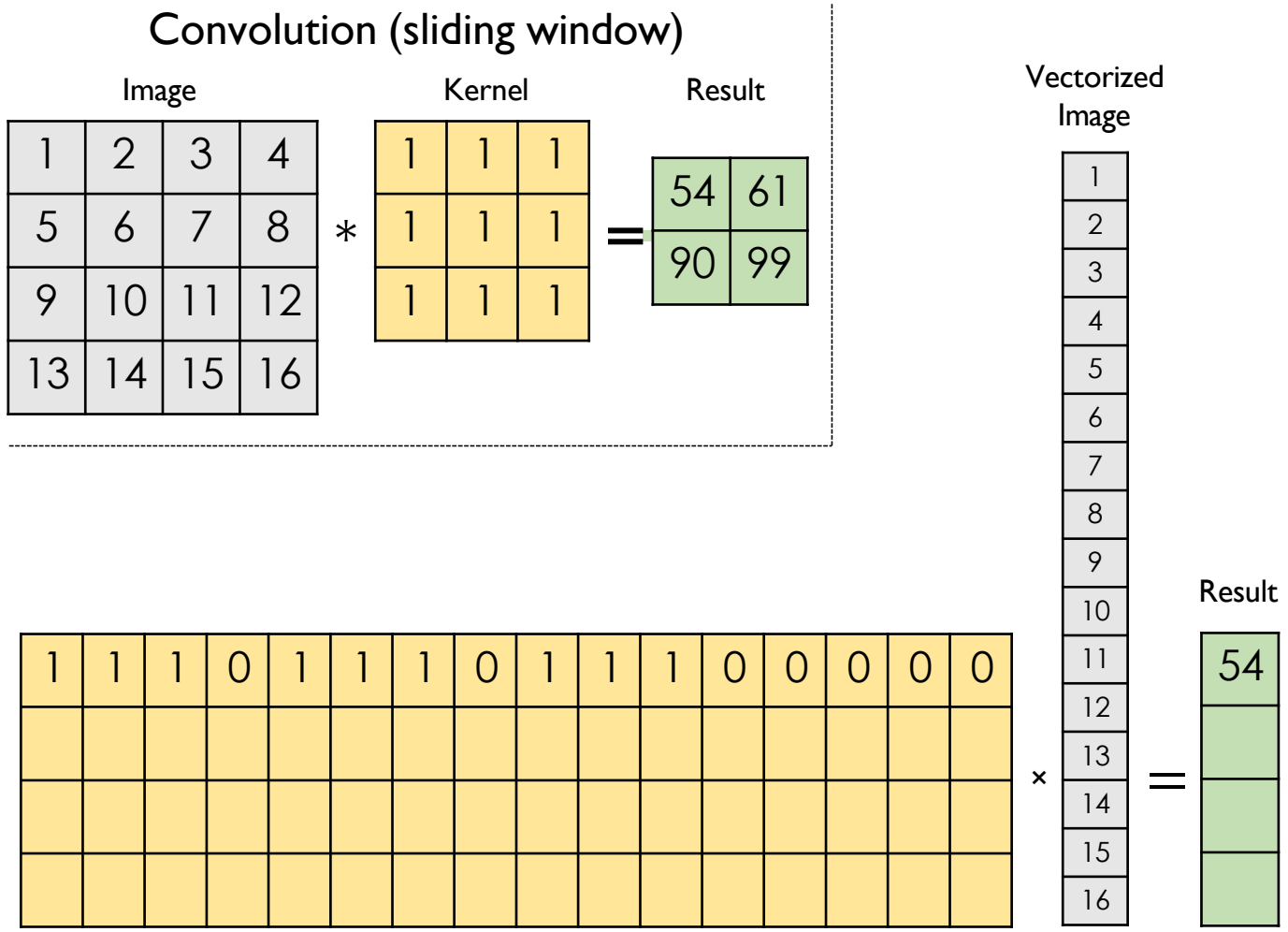


Convolution

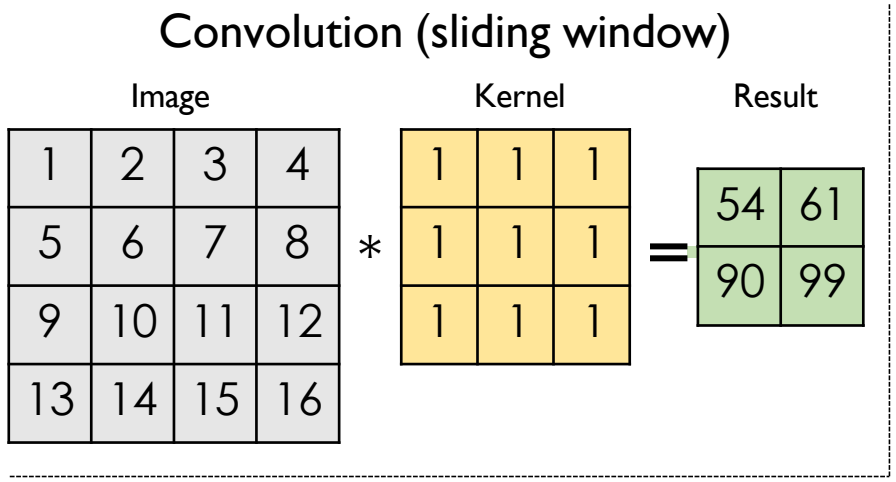
Convolution as Matrix-Vector Multiplication



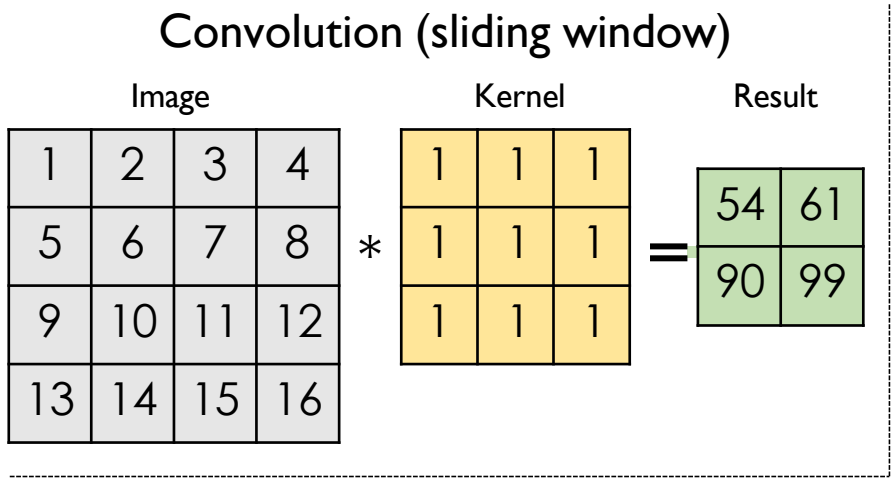
Convolution as Matrix-Vector Multiplication



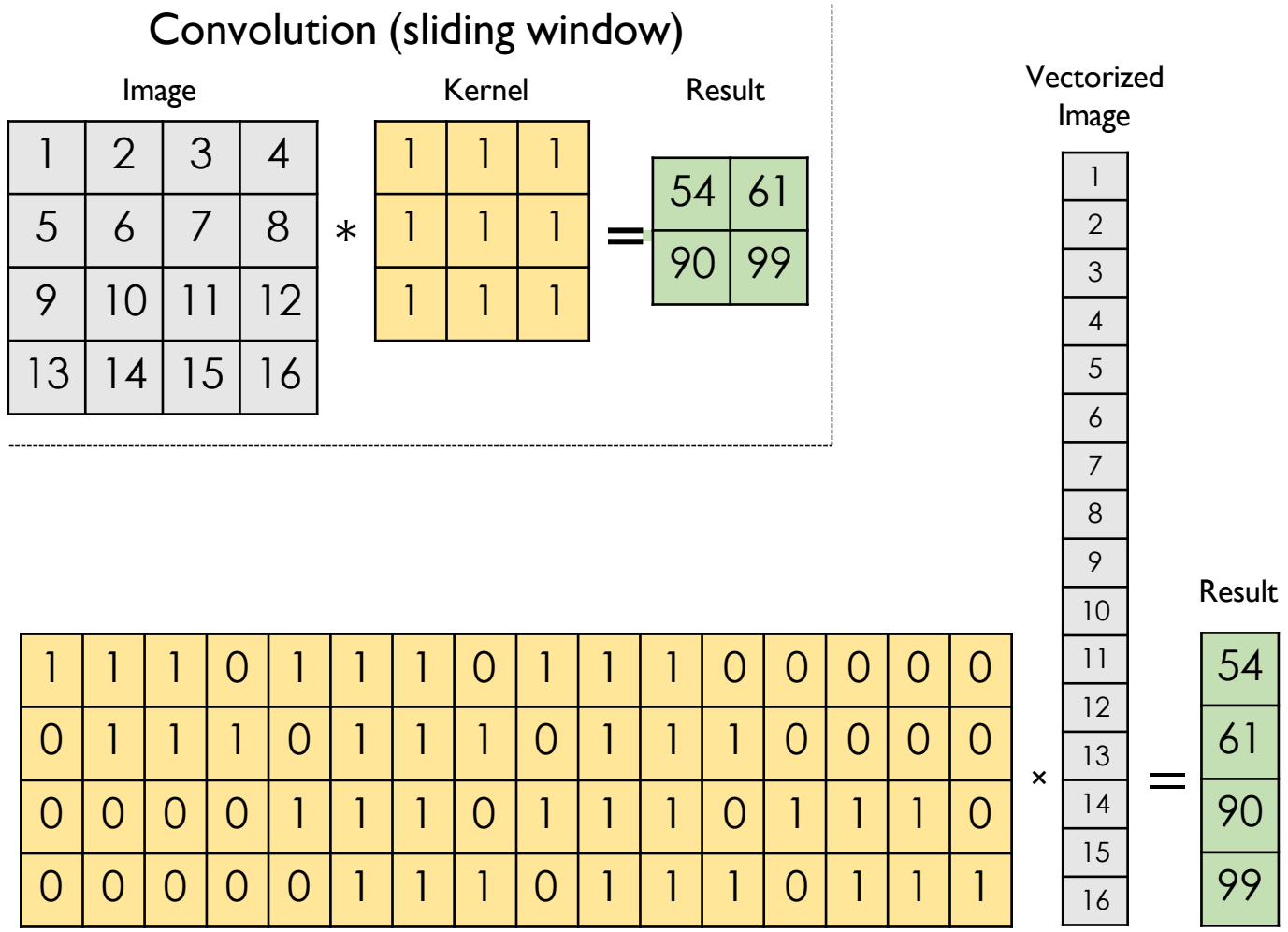
Convolution as Matrix-Vector Multiplication



Convolution as Matrix-Vector Multiplication

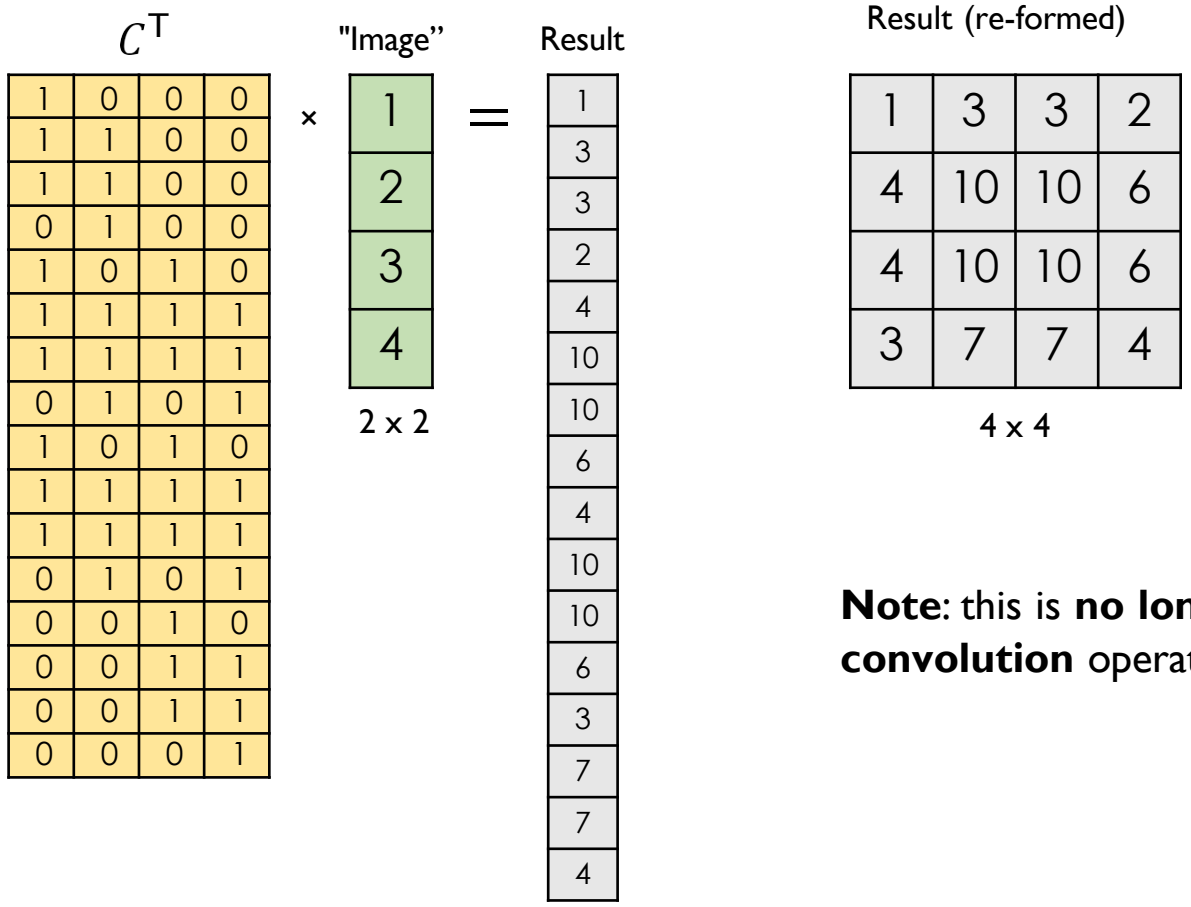


Convolution as Matrix-Vector Multiplication



Transposed Convolution

Transposed Convolution Matrix



1	0	0	0
1	1	0	0
1	1	0	0
0	1	0	0
1	0	1	0
1	1	1	1
1	1	1	1
0	1	0	1
1	0	1	0
1	1	1	1
1	1	1	1
0	1	0	1
0	0	1	0
0	0	1	1
0	0	1	1
0	0	0	1

1
2
3
4

1
3
3
2
4
10
10
6
4
10
10
6
3
7
7
4

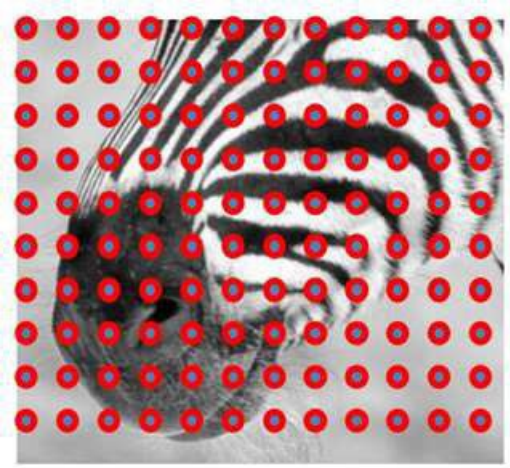
1	3	3	2
4	10	10	6
4	10	10	6
3	7	7	4

Note: this is **no longer a convolution** operation.

Analog to Upsampling/Downsampling

Downsampling (no aliasing):

- Blur image (convolve)
- Throw away every second sample along x,y



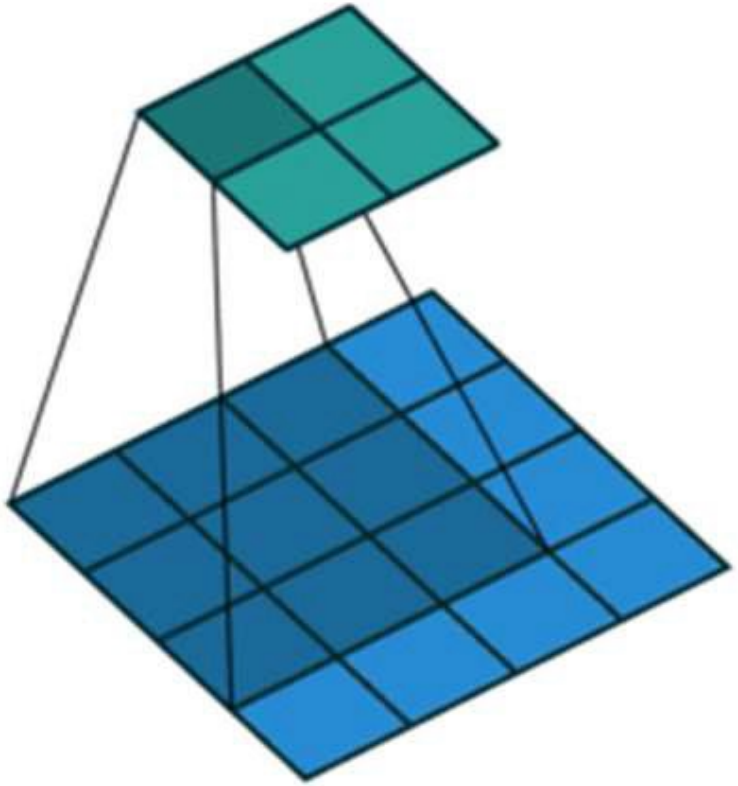
Upsampling:

- Add zeros between every sample along x,y
- Bilinearly interpolate by convolving with tent kernel

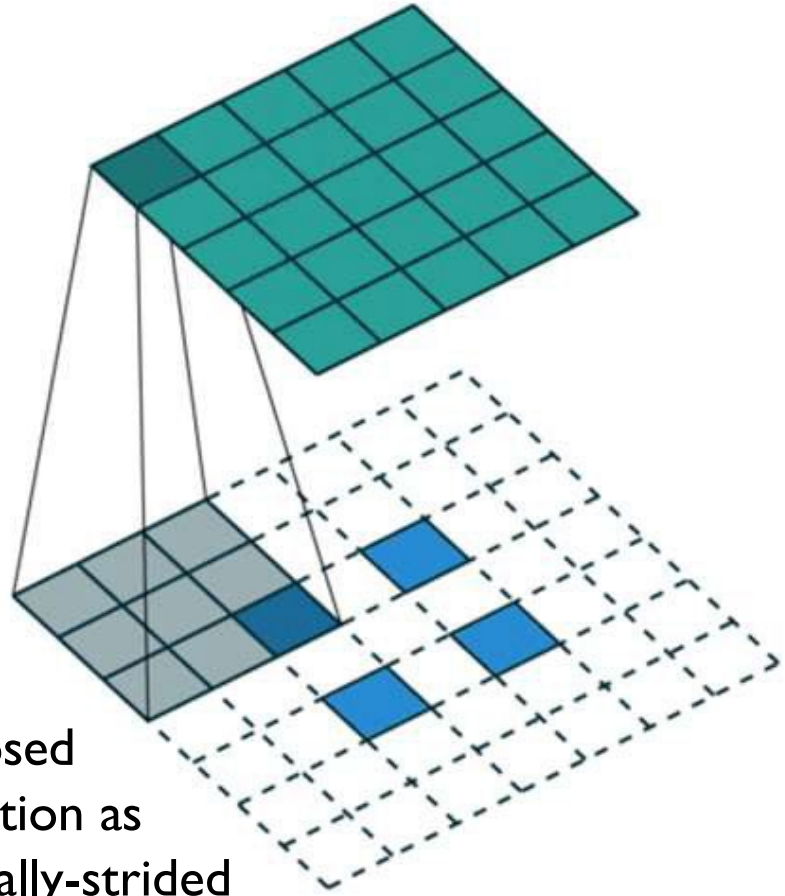
Convolve with bilinear (tent) kernel

Analog to Upsampling/Downsampling

Convolution



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

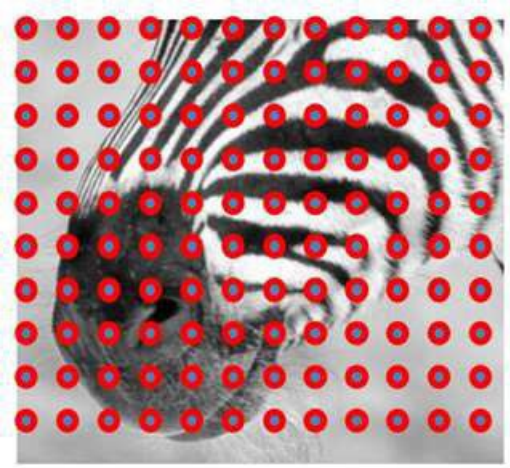


Transposed convolution as fractionally-strided convolution

Analog to Upsampling/Downsampling

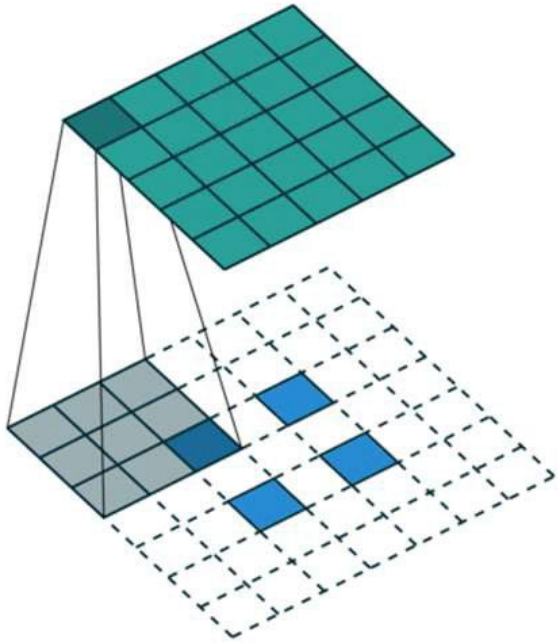
Downsampling (no aliasing):

- Blur image (convolve)
- Throw away every second sample along x,y



Upsampling:

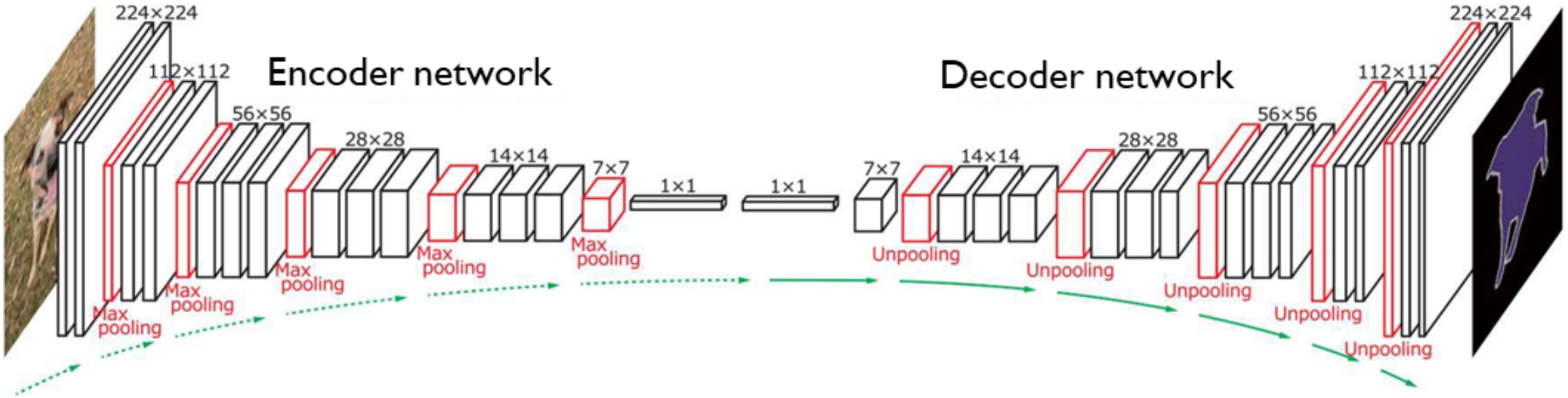
- Add zeros between every sample along x,y
- Bilinearly interpolate by convolving with tent kernel



- Upsampling analogy implies that **strided** convolution can be used to implement transpose convolution.

- Called '*fractionally-strided convolution*' as we insert empty space (new zero-valued pixels) 'halfway' between pixels.

Decoder Networks Learn to Upsample

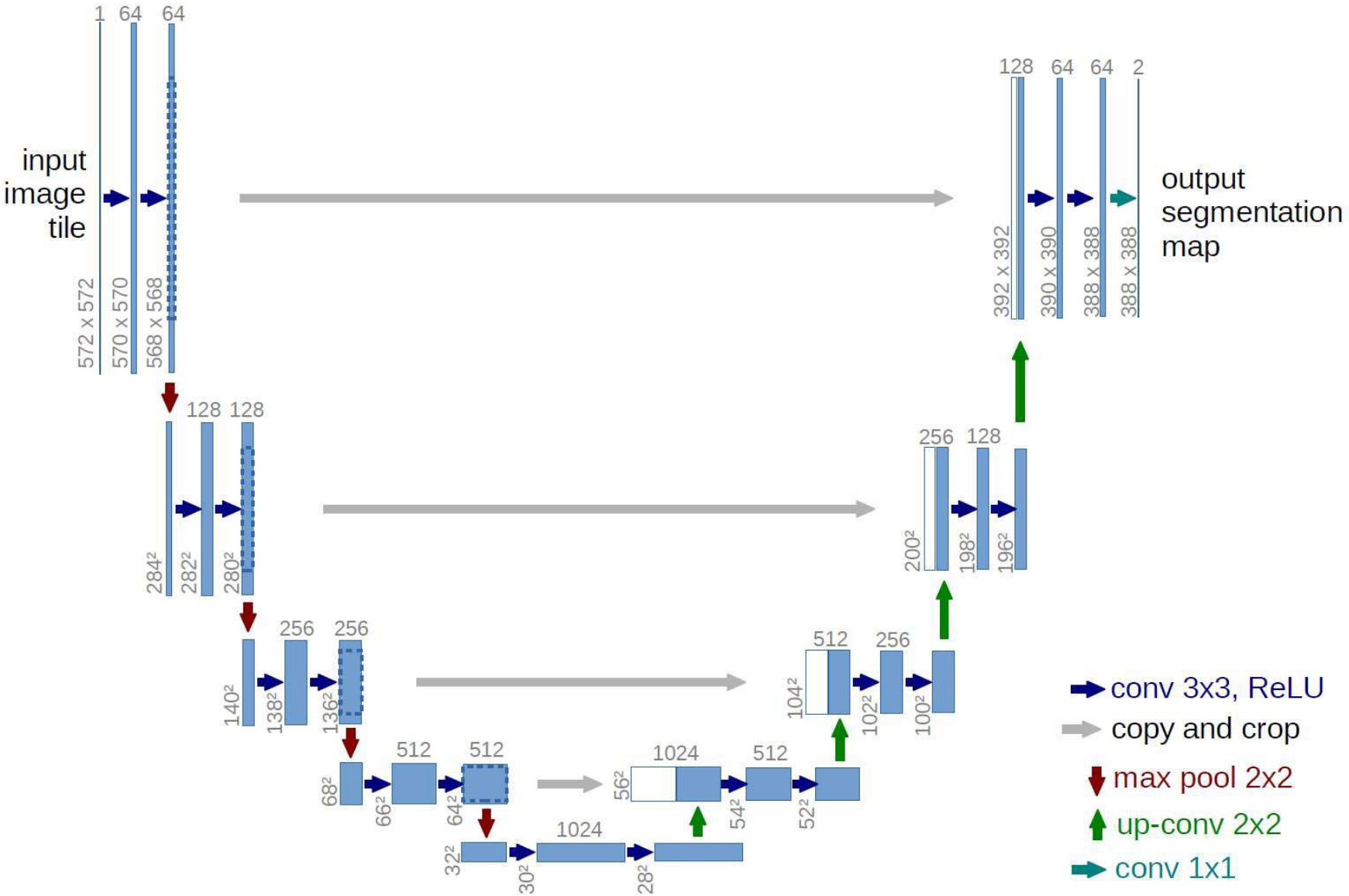


Often called "deconvolution", but misnomer.

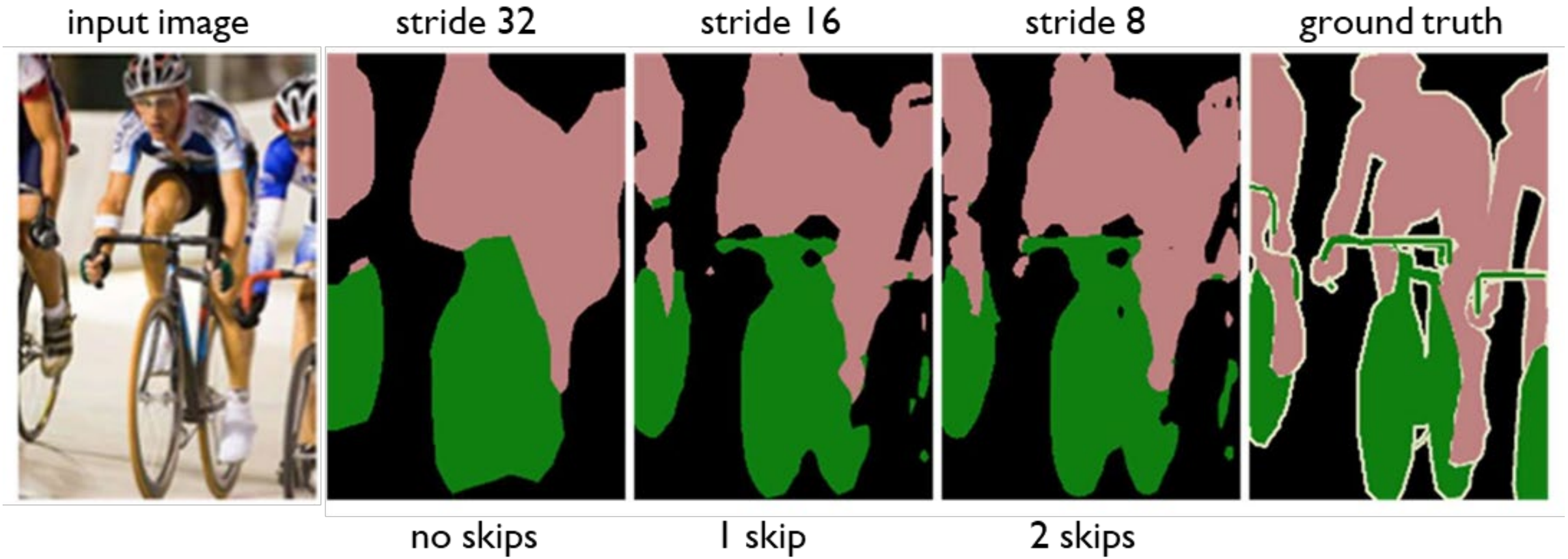
'Transposed convolution' is better.

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

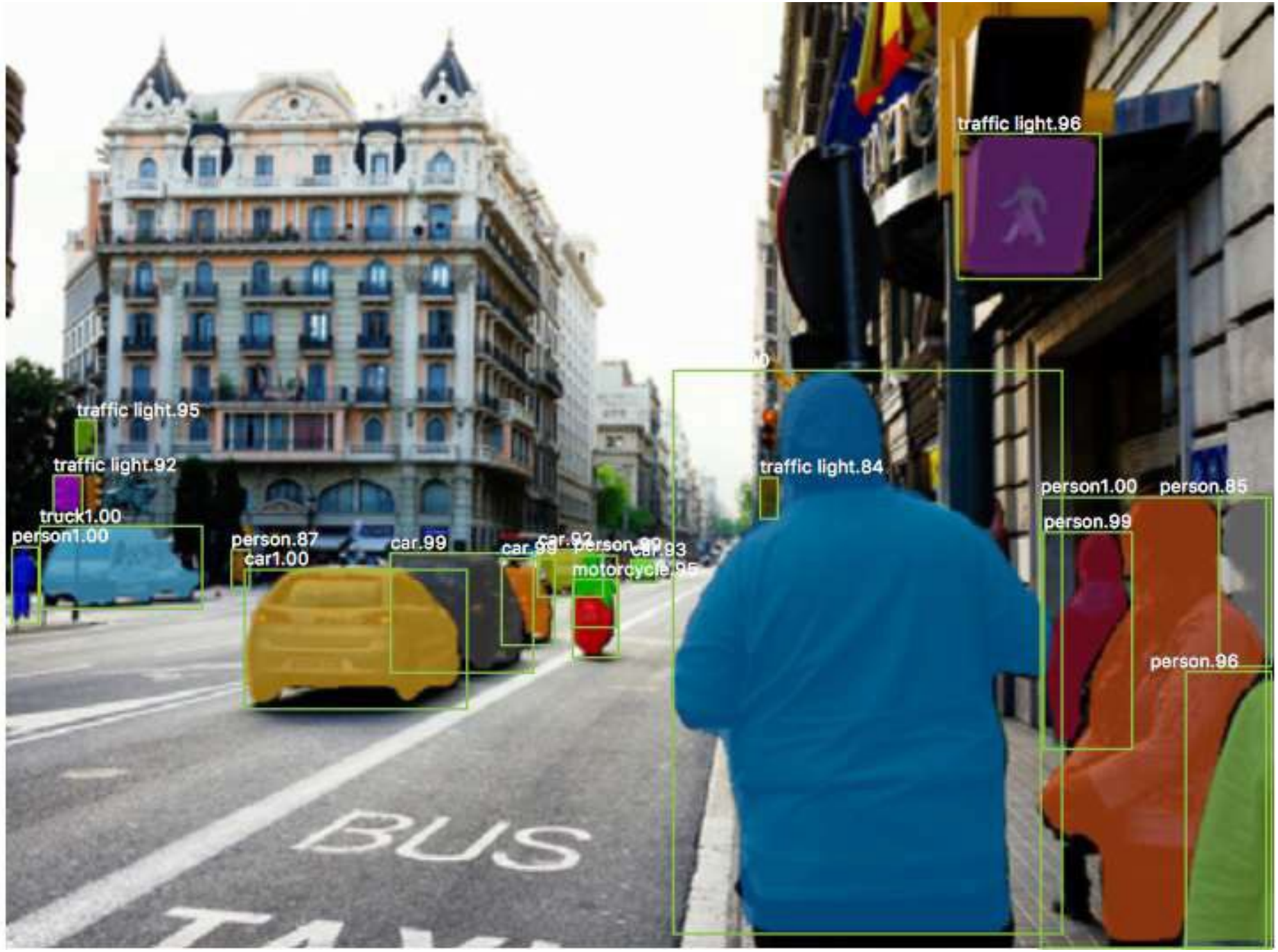
UNet [Ronneberger et al., 2015]



Skip Layer Refinement



[Long et al.]


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


person1.00 person.99 person.99 person1.00 person1.00 skateboard.91 handbag.81 skateboard.83
umbrella.97 umbrella.97 umbrella1.00 umbrella.96 umbrella.99 person1.00 handbag.97 umbrella.98 person1.00 backpack.98 backpack.95 backpack.96 handbag.85 bicycle.93
person1.00 person1.00 person1.00 person1.00 baseball bat.99 baseball bat.98 dog1.00 baseball bat.85
person1.00 person1.00 person1.00 surfboard1.00 surfboard1.00 surfboard.98 surfboard1.00 person.74
zebra.99 zebra.99 zebra.88 zebra1.00 zebra1.00
person.99 person.99 person1.00 person.82 person1.00 frisbee1.00 person.80
person.95 person.72 person.87 person.99 person.77 chair.98 chair.96 dining table.75 dining table.78 chair.95 wine glass.80 chair.92 chair.85 chair.87 chair.97 chair.94 wine glass.91 wine glass.93 wine glass.94 wine glass.83 cup.96 cup.91
kite.80 kite.81 kite.98 kite1.00 kite.82 kite.97 kite.99 kite.84 kite.93 kite.86 kite.88 kite.95 person1.00 handbag.80
elephant1.00 elephant1.00 elephant1.00 elephant.97 elephant.99
traffic light.99 traffic light.71 person.99 person1.00 tie.85 handbag.88 handbag.88 handbag.99
person1.00 person.88 tv.98 tv.84 person1.00 bottle.97 wine glass.99 dining table.95 wine glass1.00 wine glass1.00
potted plant.92 person1.00 person.98 bird.93 bench.97 handbag.73 cell phone.77