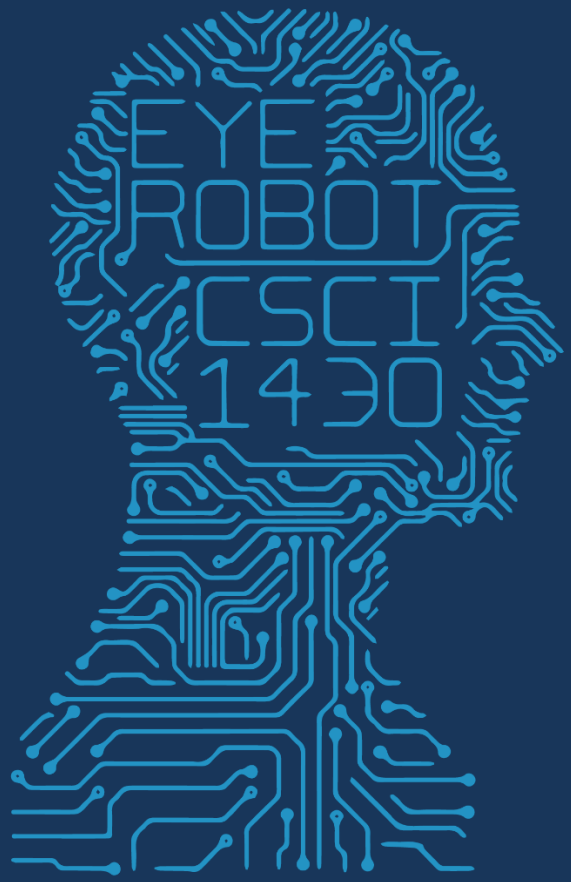
1950

Future Vision



2017 MWF 1pm 368

Computer Vision

FaceApp

- Learning-based face transformations

neural networks
FaceApp
algorithmic bias
algorithmic accountability
Artificial Intelligence

FaceApp apologizes for building a racist AI

Posted 45 minutes ago by ***Natasha Lomas*** *(@riptari)*

2017/04/25

If only all algorithmic bias were as easy to spot as this: FaceApp, a photo-editing app that uses a neural network for editing selfies in a photorealistic way, has apologized for building a racist algorithm.

The app lets users upload a selfie or a photo of a face, and offers a series of filters that can then be applied to the image to subtly or radically alter its appearance — its appearance-shifting effects include aging and even changing gender.

The problem is the app also included a so-called "hotness" filter, and this filter was racist. As users pointed out, the filter was lightening skin tones to achieve its mooted "beautifying" effect. You can see the filter pictured above in a before and after shot of President Obama.

In an emailed statement apologizing for the racist algorithm, FaceApp's founder and CEO Yaroslav Goncharov told us: "We are deeply sorry for this unquestionably serious issue. It is an unfortunate side-effect of the underlying neural network caused by the training set bias, not intended behaviour. To mitigate the issue, we have renamed the effect to exclude any positive connotation associated with it. We are also working on the complete fix that should arrive soon."

Popular Posts


Doug finds the best Amazon deals
4 days ago

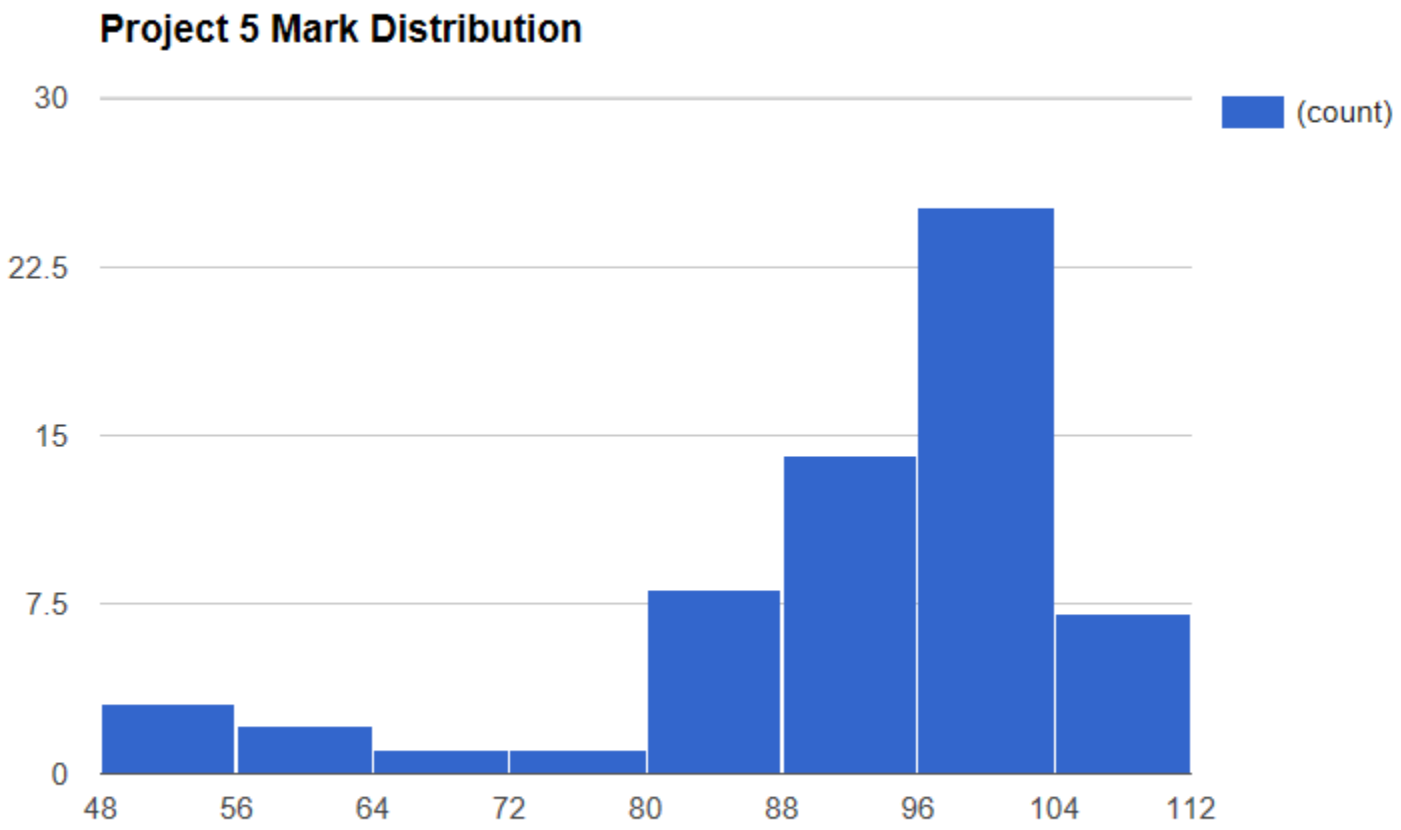

Elon Musk's Neuralink wants to turn cloud-based AI into an extension of our brains
4 days ago


Oculus co-founder Palmer Luckey donated $100,000 to Trump's inauguration
5 days ago

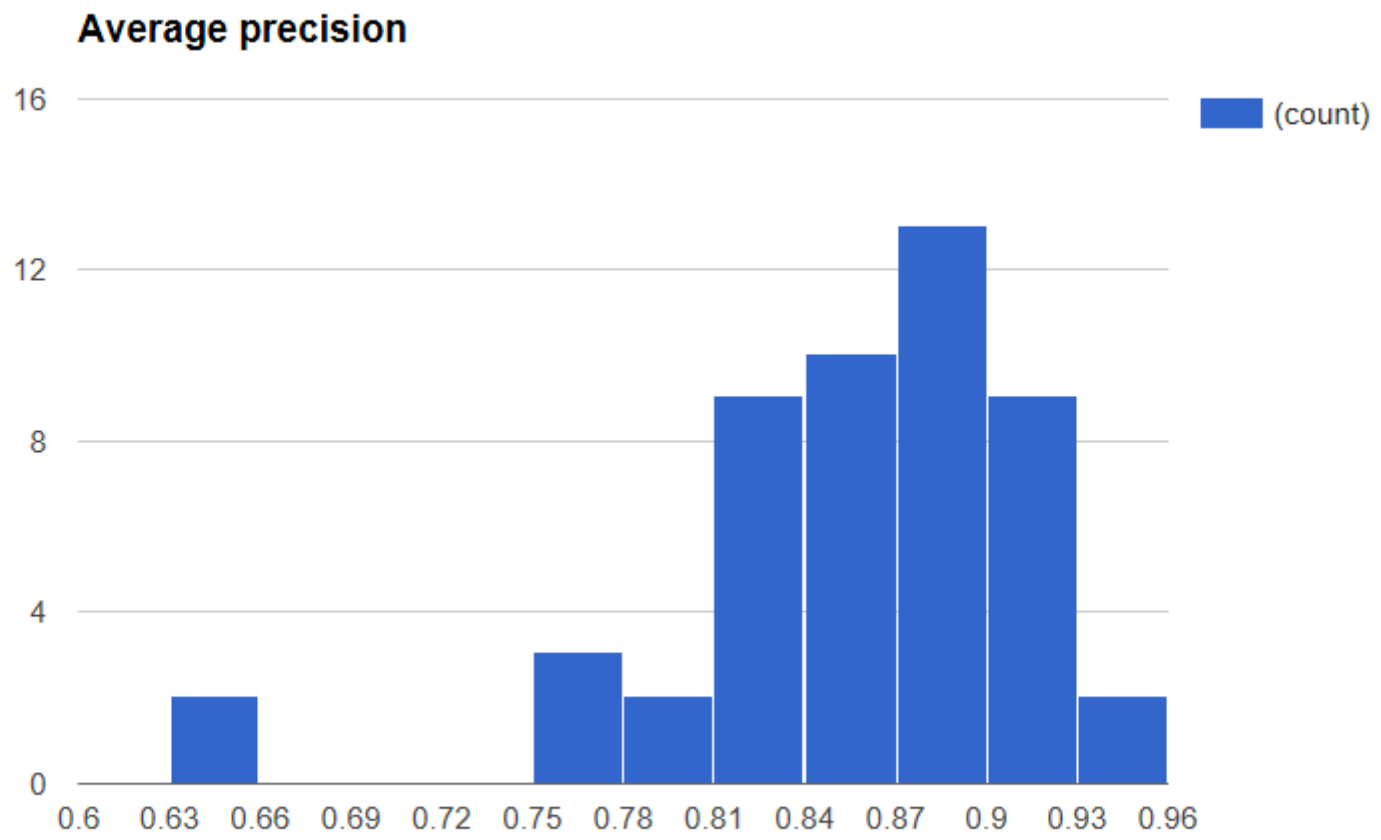

FTC tells 'influencers' to quit trying to hide the fact that they're shilling for brands
5 days ago


Uber gets sued over alleged 'Hell' program to track Lyft drivers
a day ago

Project 5 mark distribution



Project 5 avg. prec. distribution



Project 5 – well done!

“Finally, there will be extra credit and recognition for the students who achieve the highest average precision.”

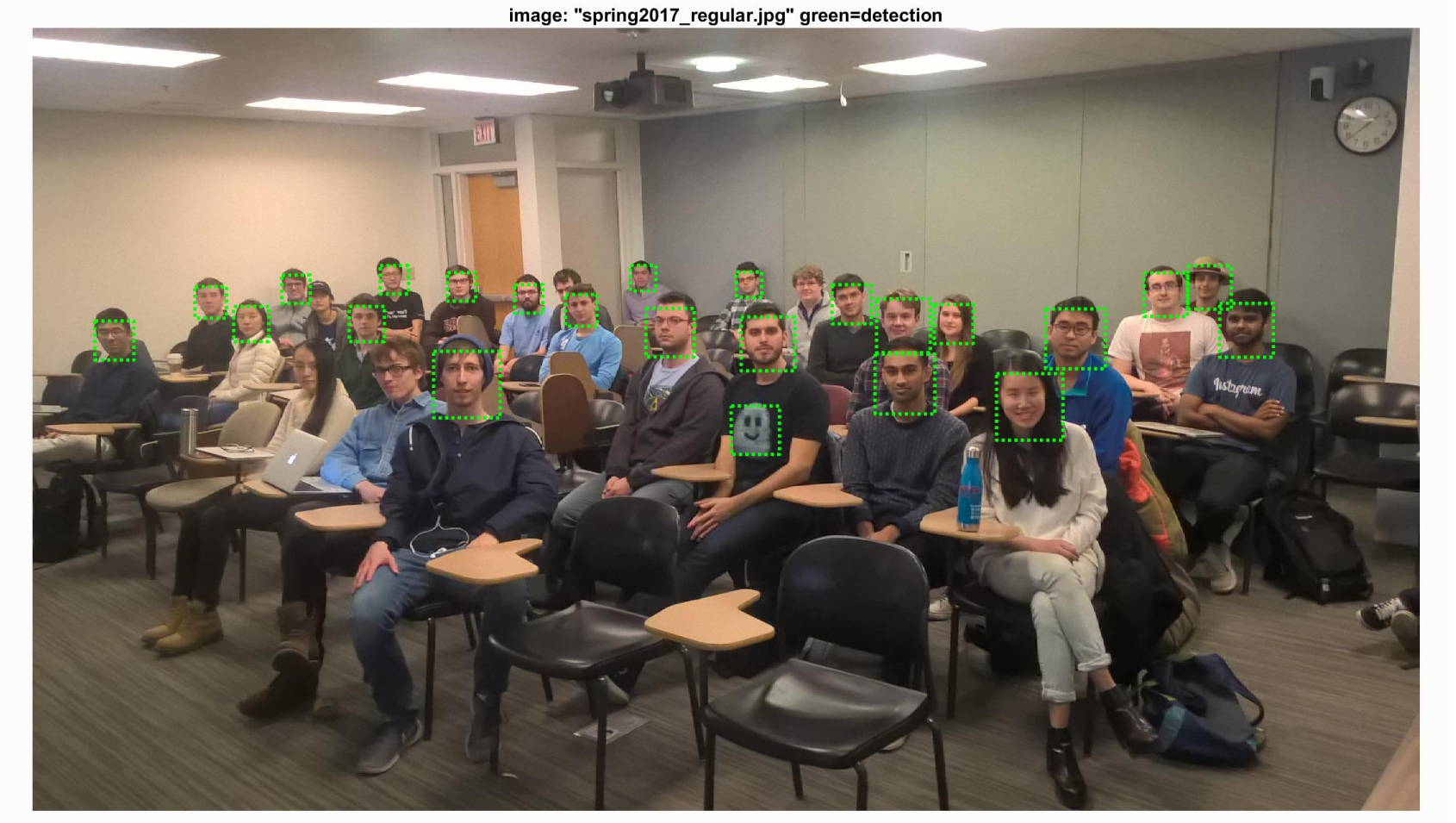
Drumroll please…

*. 100% - Multi-scale squares at every position

1. 93.9% - Kyle Myerson
2. 93.7% - Tiffany Chen
3. 92.7% - Qikun (Tim) Guo
4. 92.4% - Lucas Lehnert
5. 92.1% - Katya Schwiegershausen

Results on our photos

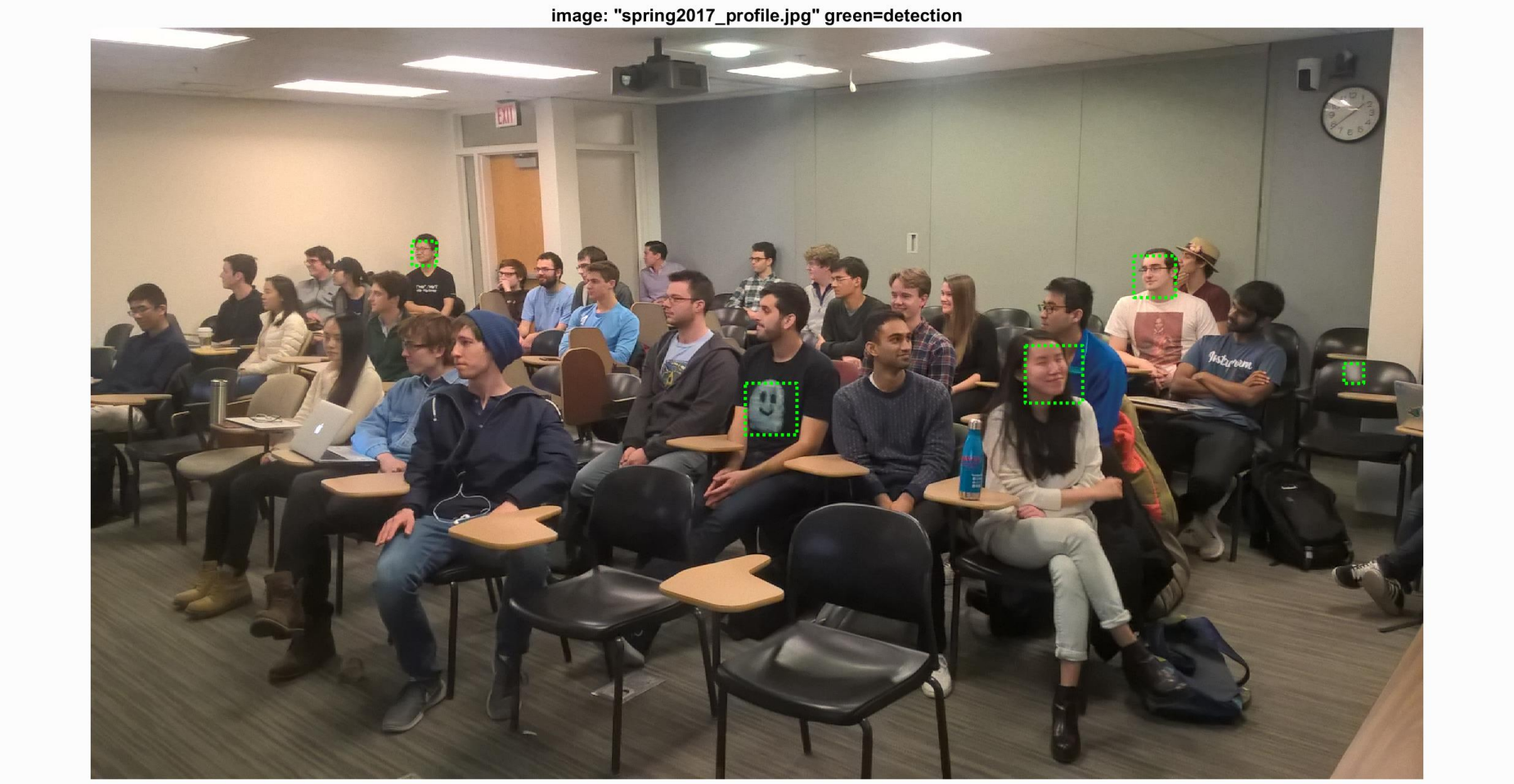
Results on our photos



[Kyle Myerson]

Profile?

Profile?



[Kyle Myerson]

Are there any faces here?

No faces…

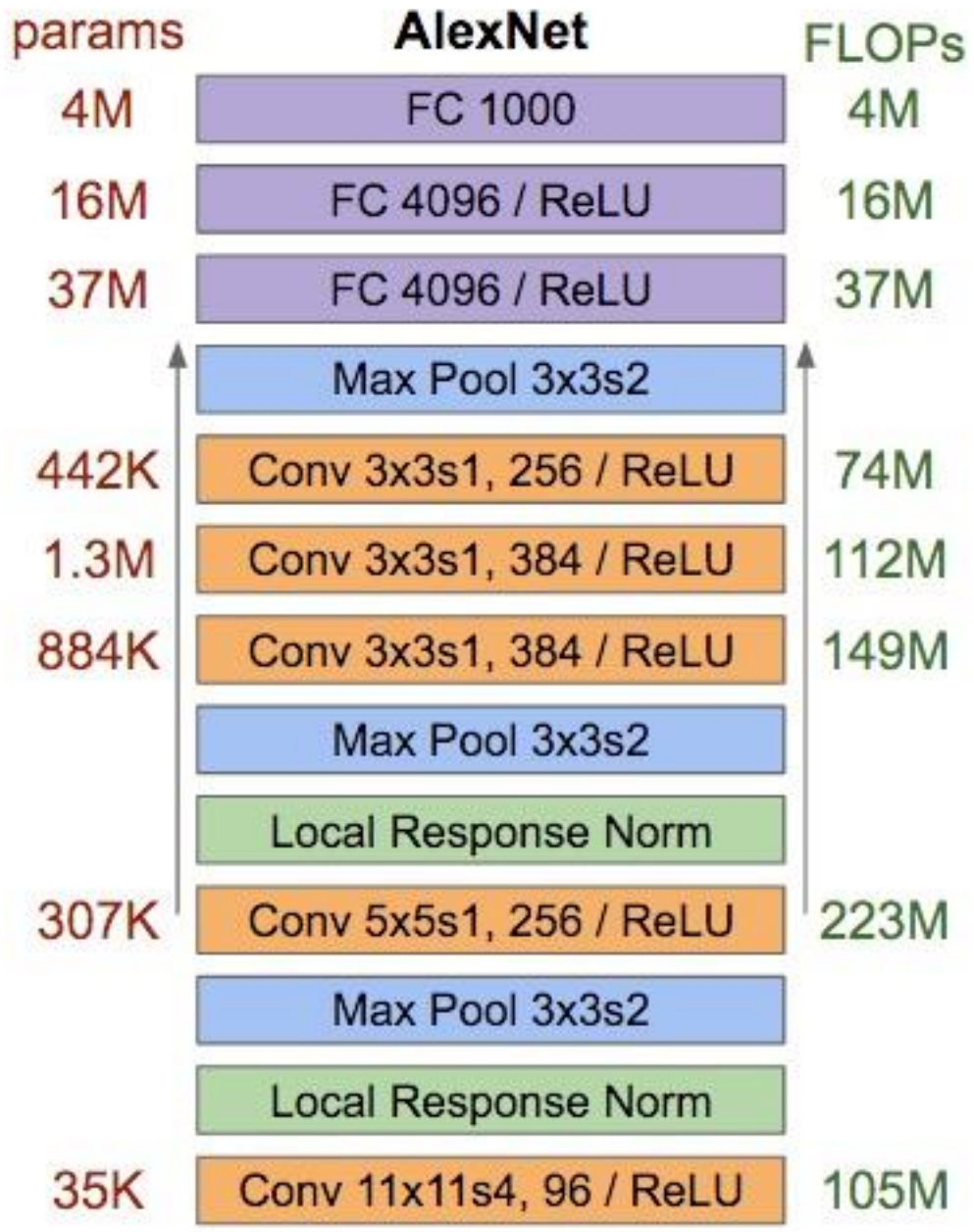
[Kyle Myerson]

More TAs!

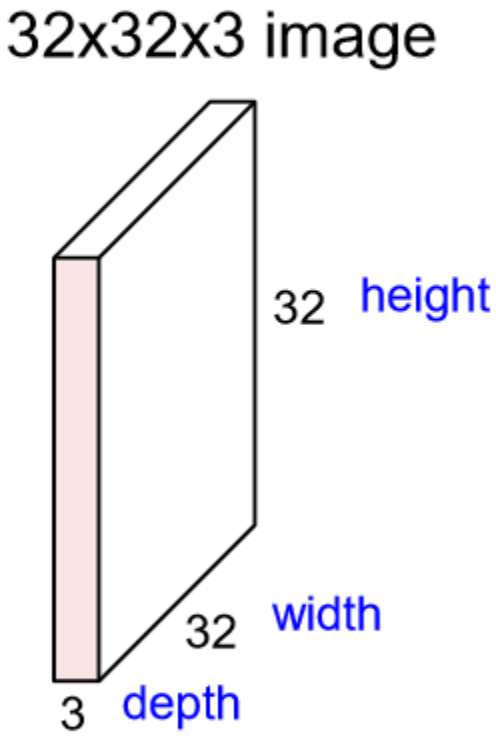
- 70 in class now (+ 70 on waitlist)
- 107 pre-registered for fall 2017

Reading architecture diagrams

Layers
- Kernel sizes
- Strides
- # channels
- # kernels
- Max pooling

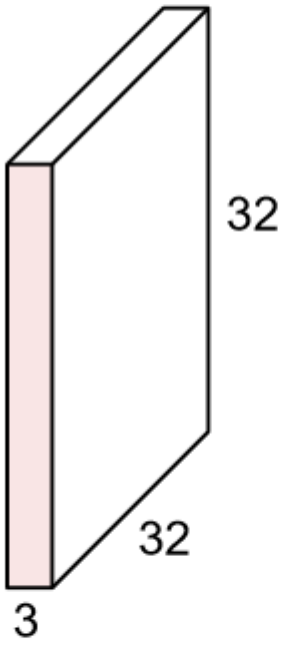


Convolutions: More detail



Convolutions: More detail

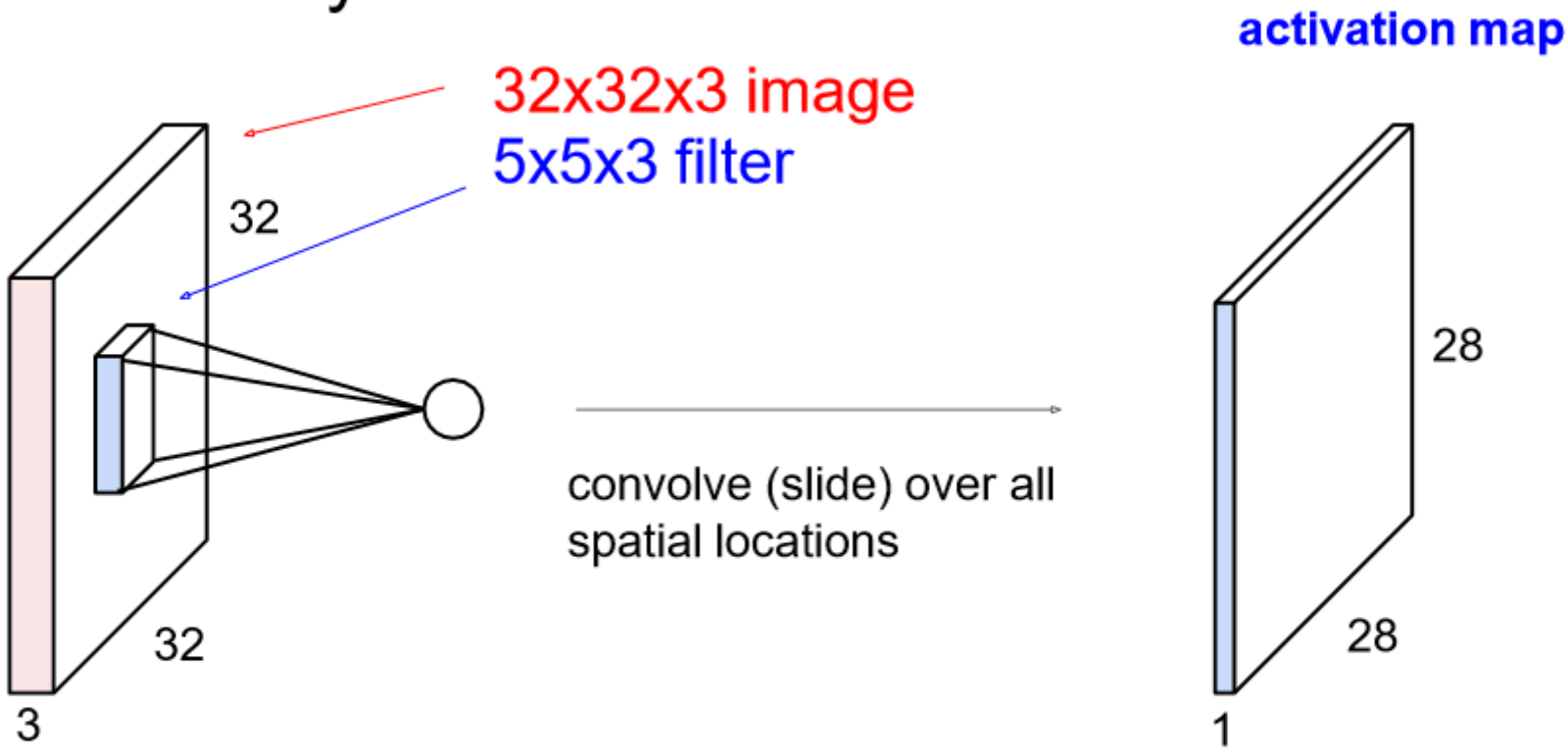
32x32x3 image



5x5x3 filter

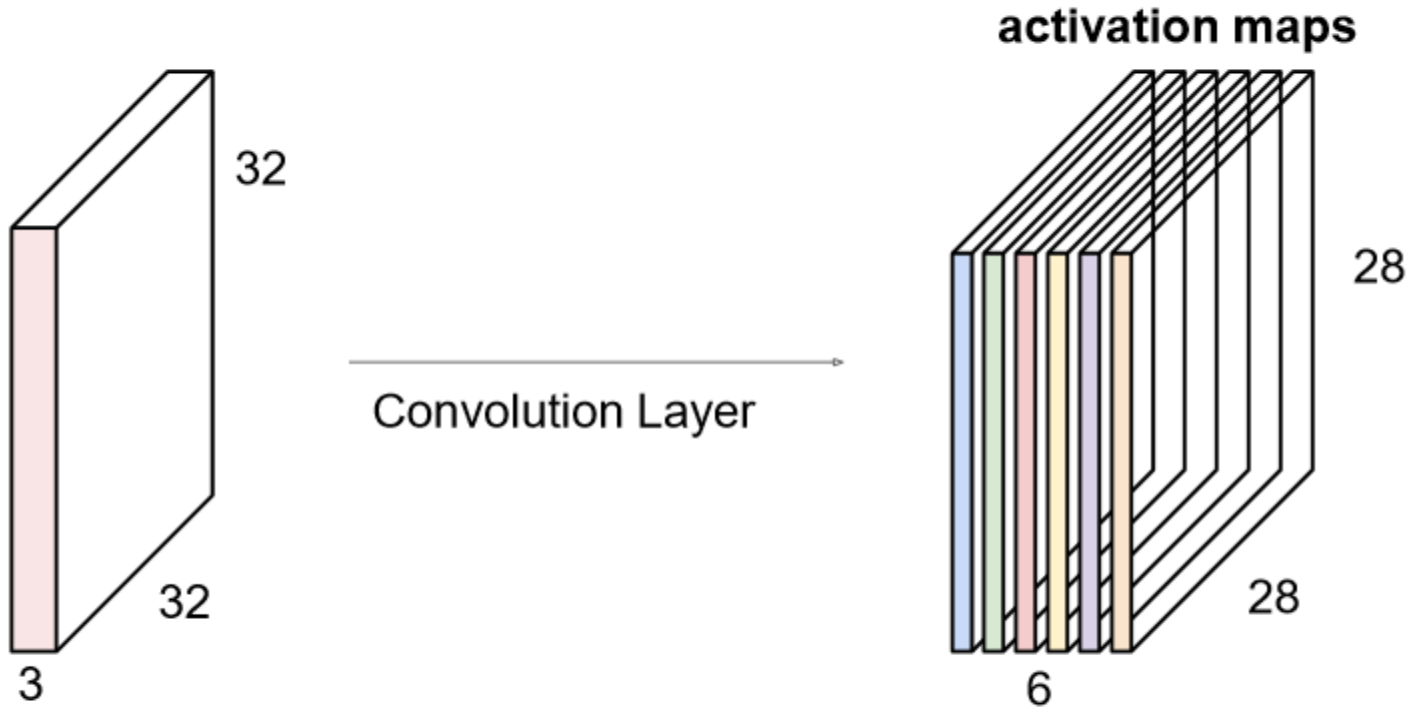
Convolutions: More detail

Convolution Layer



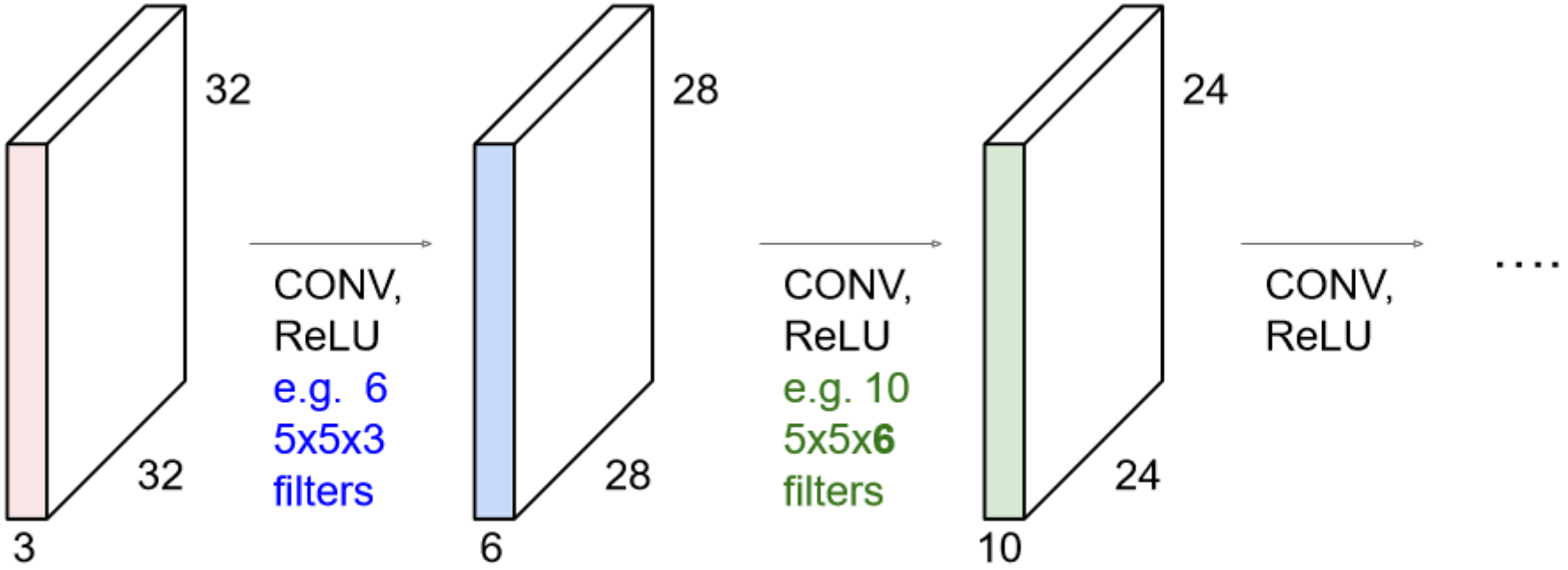
Convolutions: More detail

For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:

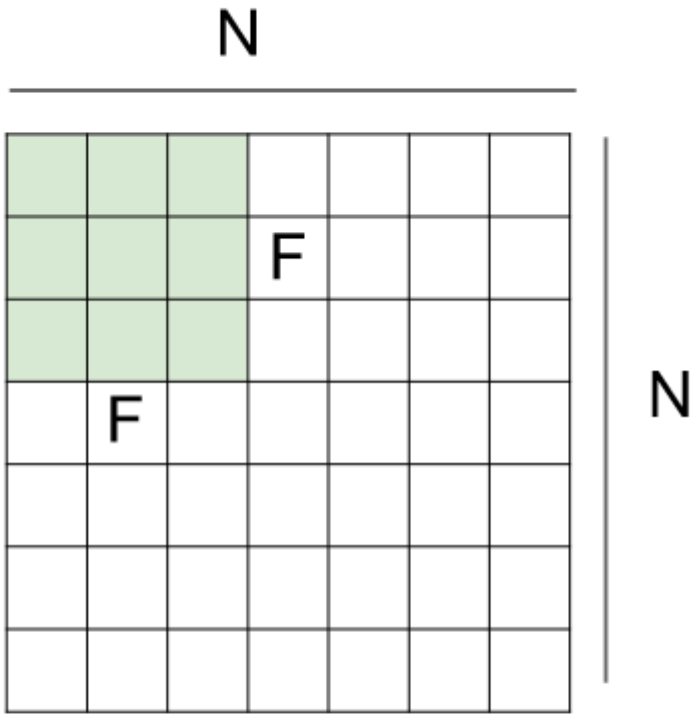


We stack these up to get a "new image" of size 28x28x6!

Convolutions: More detail



Convolutions: More detail

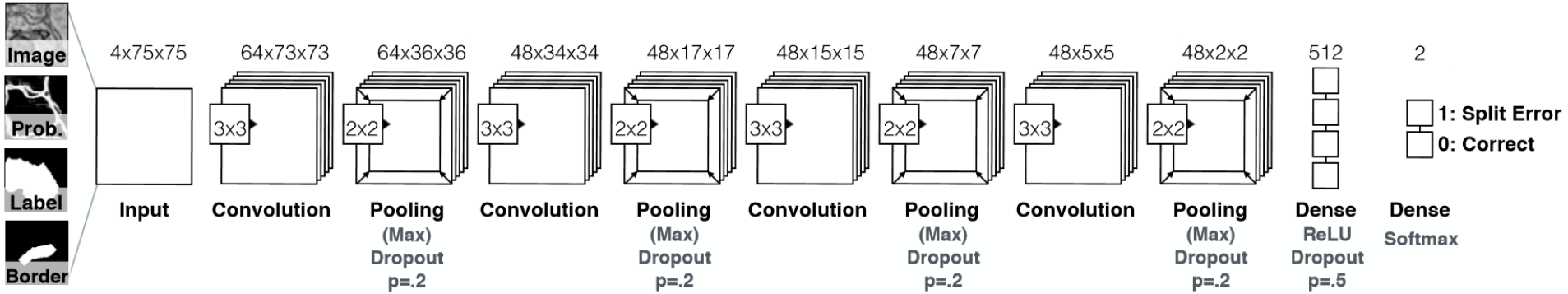


Output size:
(N - F) / stride + 1

Our connectomics diagram

Auto-generated from network declaration by *nolearn* (for Lasagne / Theano)

Input
75x75x4



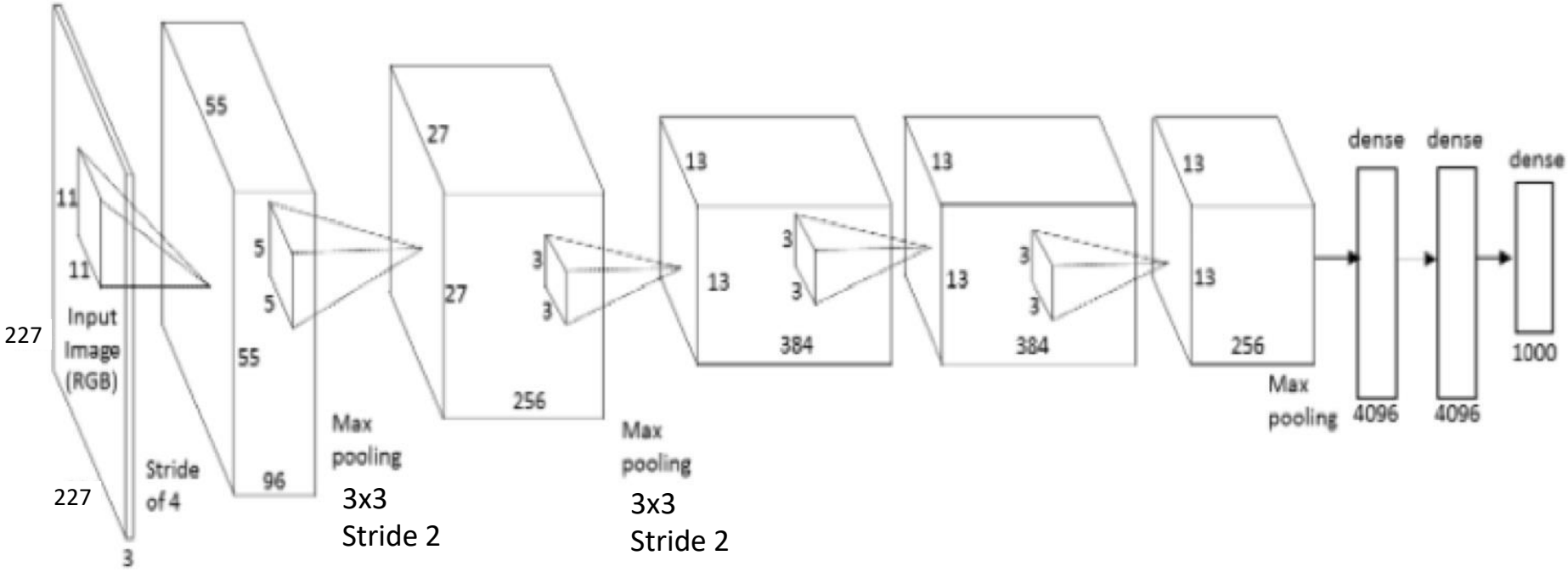
Conv 1	Conv 2	Conv 3	Conv 4
3x3x4	3x3x64	3x3x48	3x3x48
64 filters	48 filters	48 filters	48 filters

Max pooling	Max pooling	Max pooling	Max pooling
2x2 per filter	2x2 per filter	2x2 per filter	2x2 per filter

[Krizhevsky et al. 2012]

AlexNet diagram (simplified)

Input size
227 x 227 x 3



Conv 1
11 x 11 x 3
Stride 4
96 filters

Conv 2
5 x 5 x 48
Stride 1
256 filters

Conv 3
3 x 3 x 256
Stride 1
384 filters

Conv 4
3 x 3 x 192
Stride 1
384 filters

Conv 4
3 x 3 x 192
Stride 1
256 filters

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 6

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					

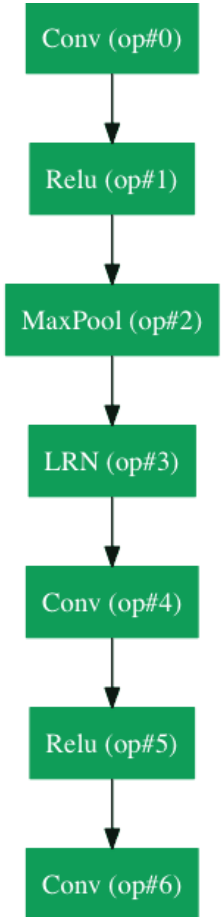
Table 2: **Number of parameters** (in millions).

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**

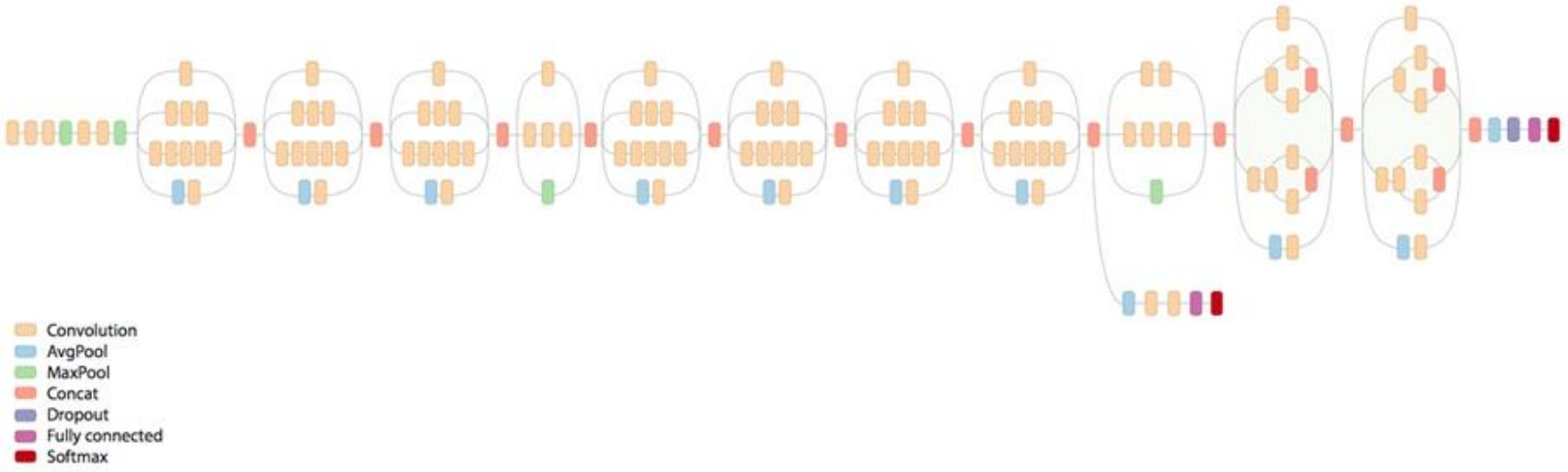
Google LeNet (2014)



22 layers

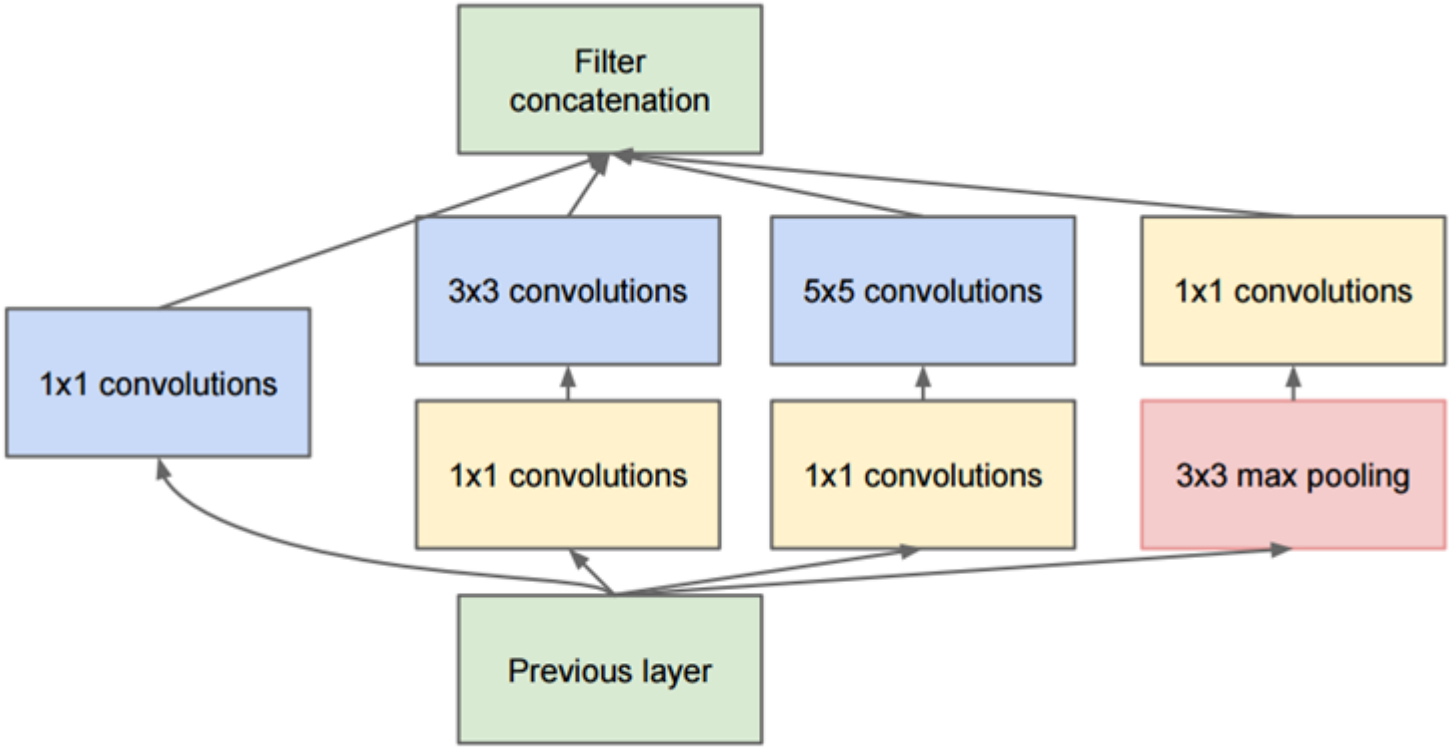
6.67% error
ImageNet top 5

Inception!



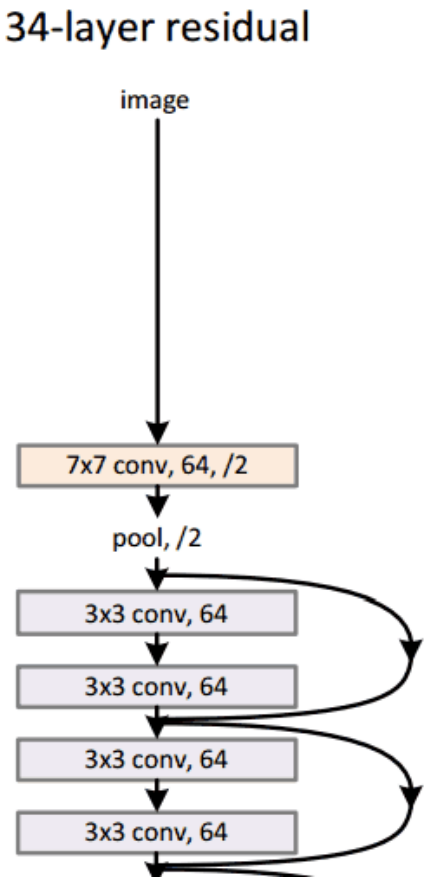
Another view of GoogLeNet's architecture.

Parallel layers



Full Inception module

ResNet (He et al., 2015)

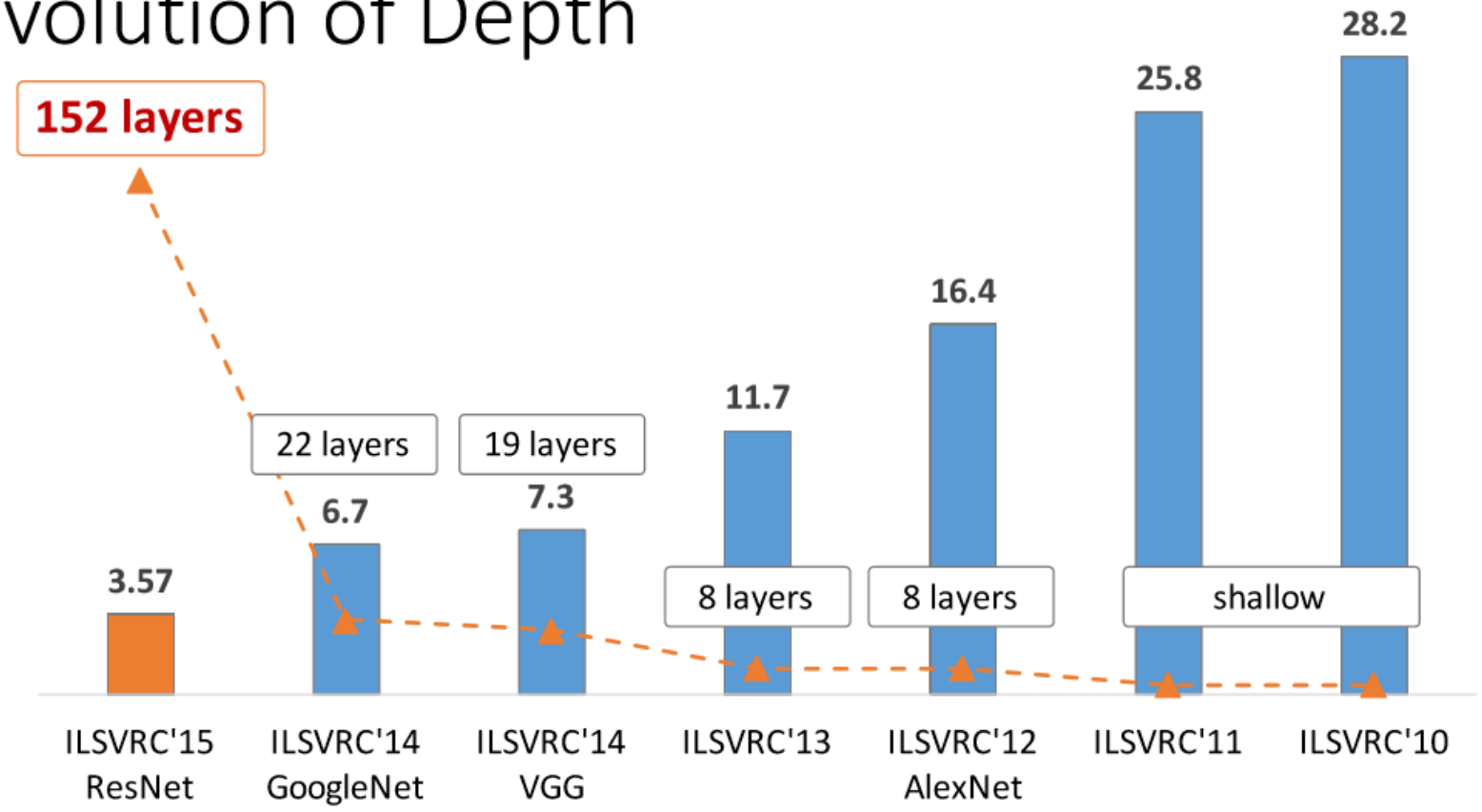


ResNet won ILSVRC 2015 with a top-5 error rate of 3.6%

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

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)

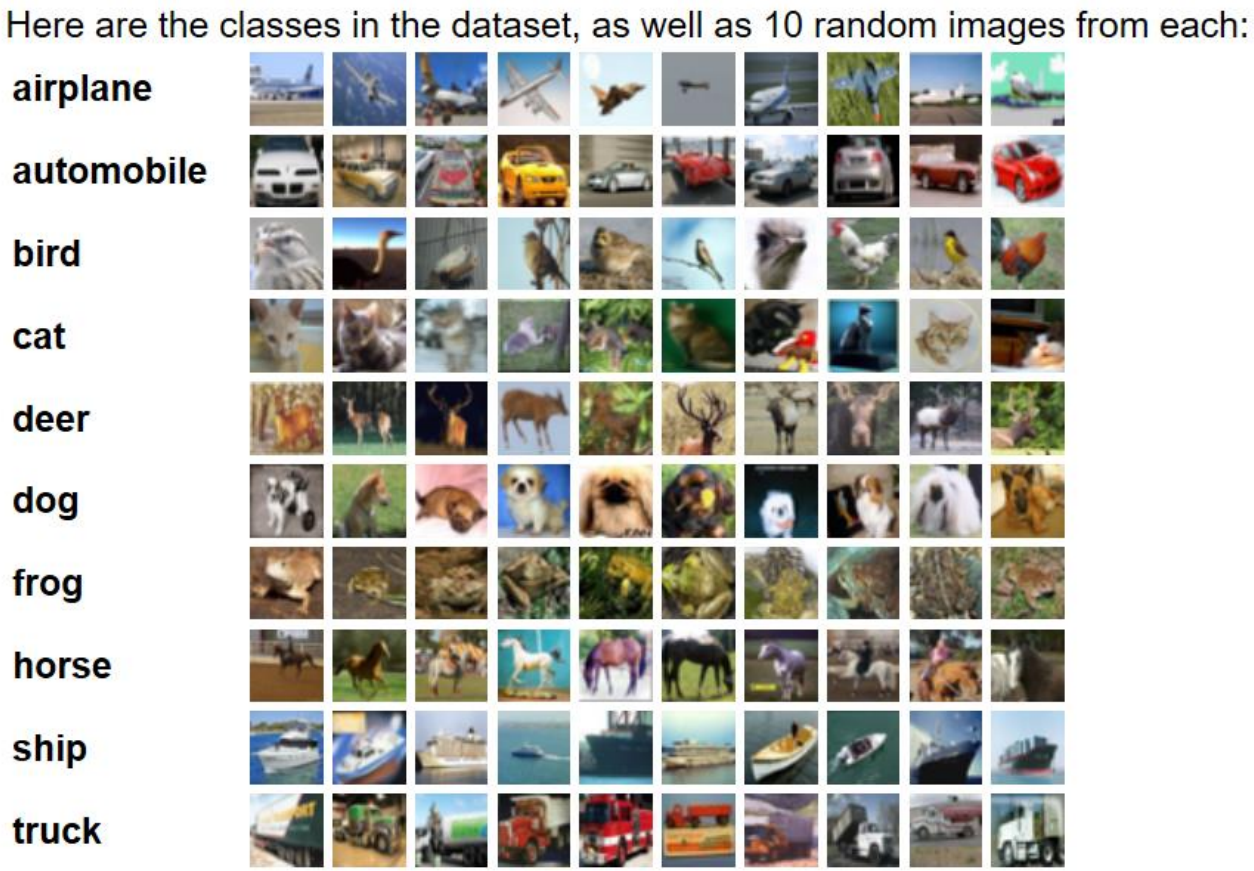
VGG, 19 layers
(ILSVRC 2014)

ResNet, 152 layers
(ILSVRC 2015)

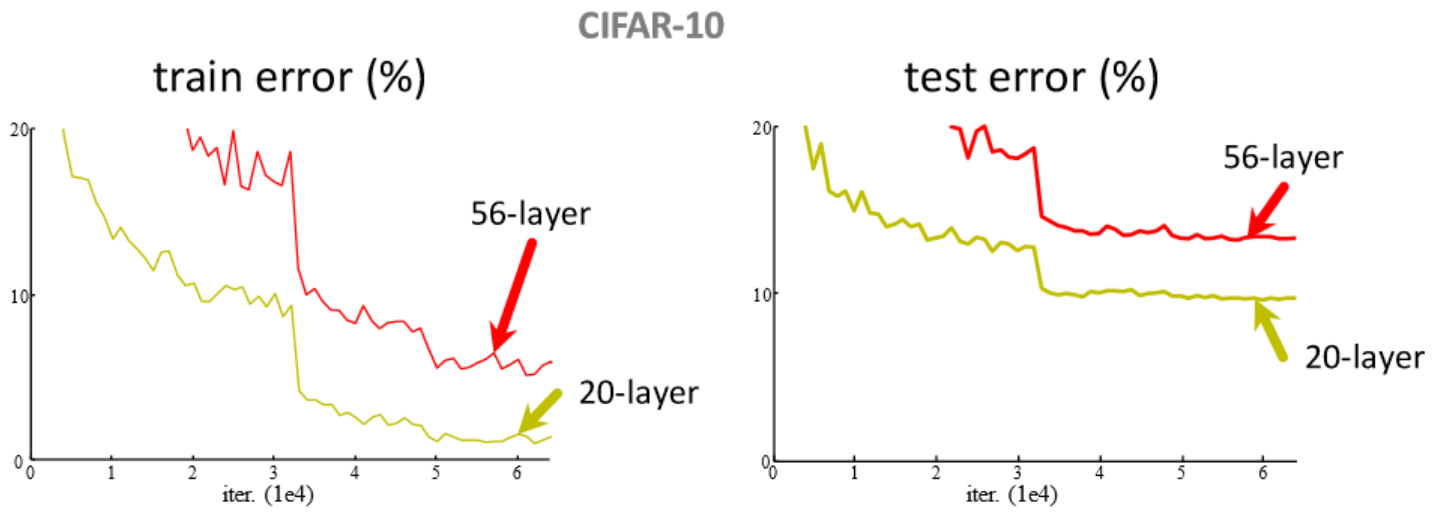
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

CIFAR-10

- 60,000 32x32 color images, 10 classes



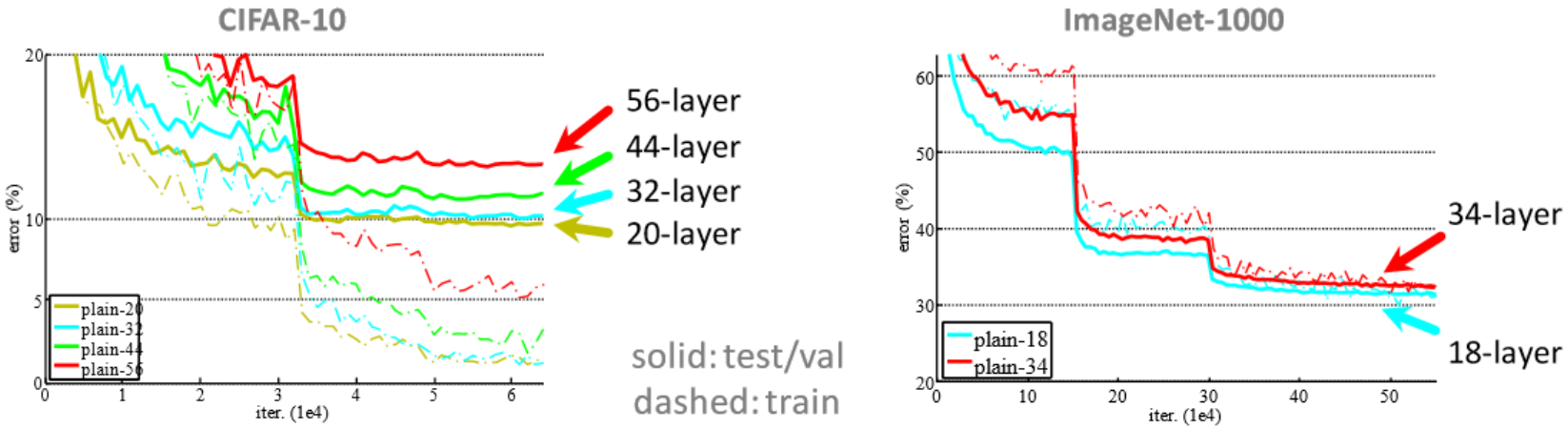
Simply stacking layers?



- *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.

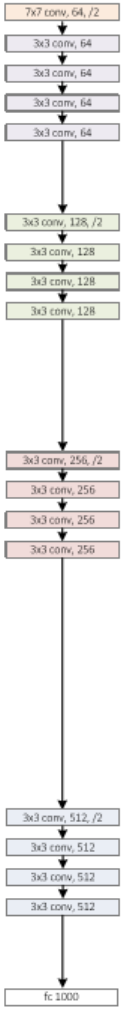
Simply stacking layers?

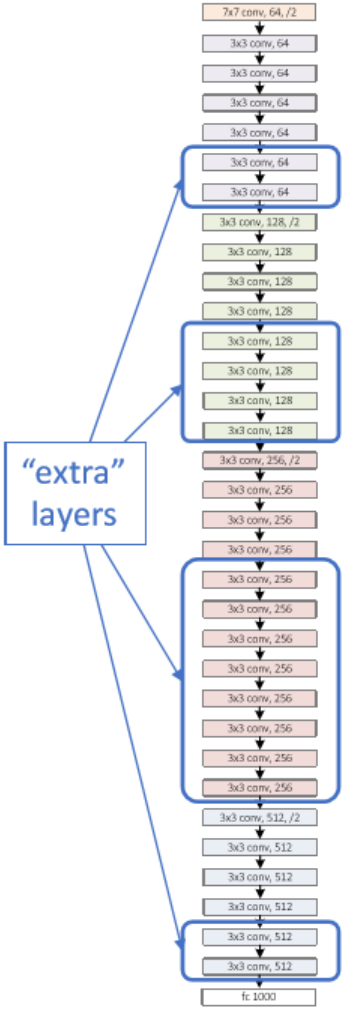


- "Overly deep" plain nets have **higher training error**
- A general phenomenon, observed in many datasets

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

a shallower model (18 layers)





a deeper counterpart (34 layers)

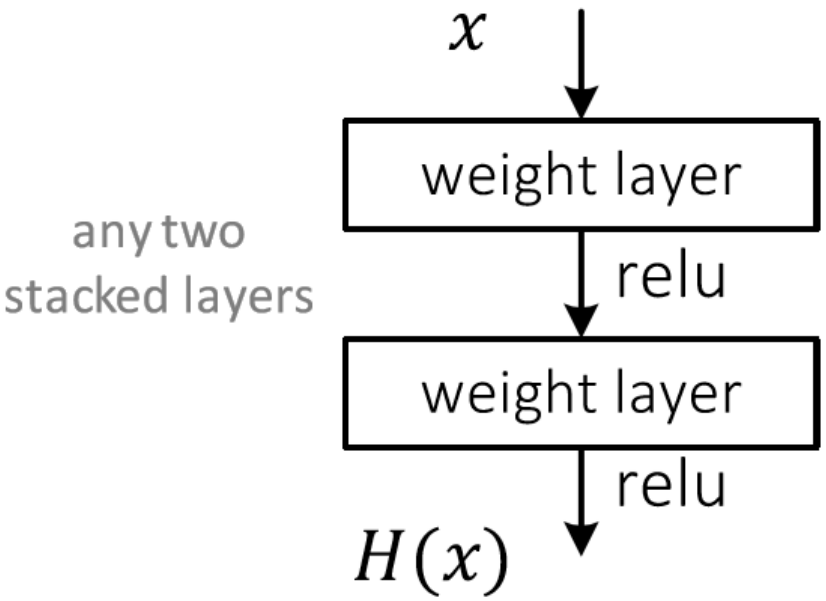
- Richer solution space
- A deeper model should not have **higher training error**

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

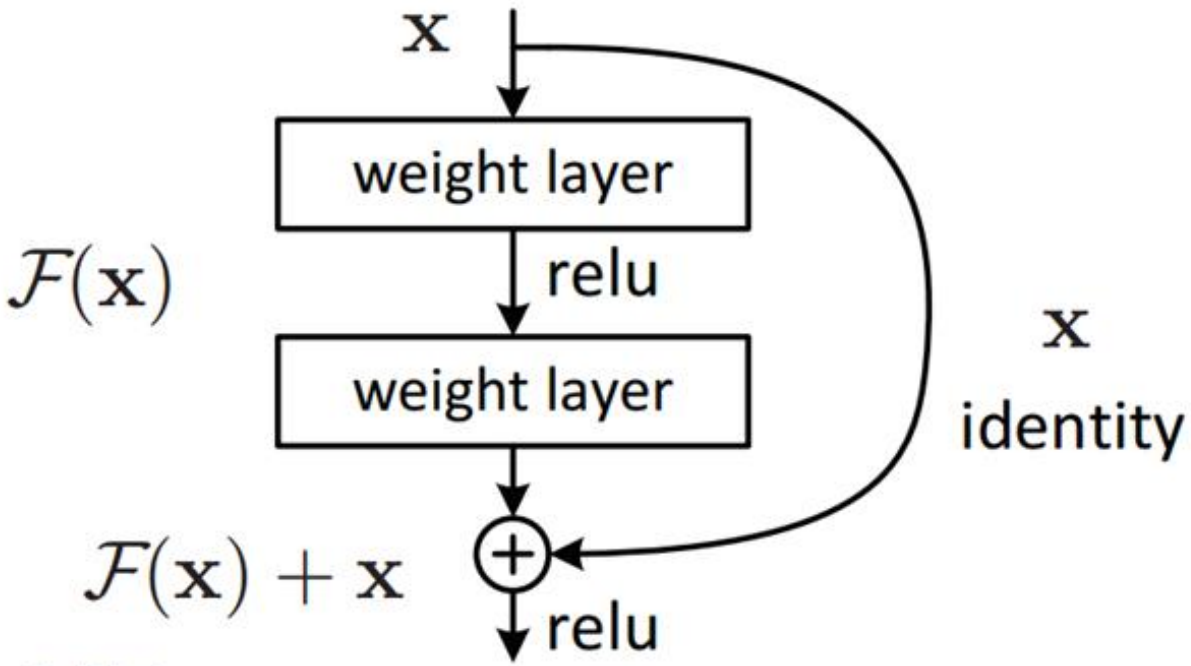
Regular net

$H(x)$ is any desired mapping,

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



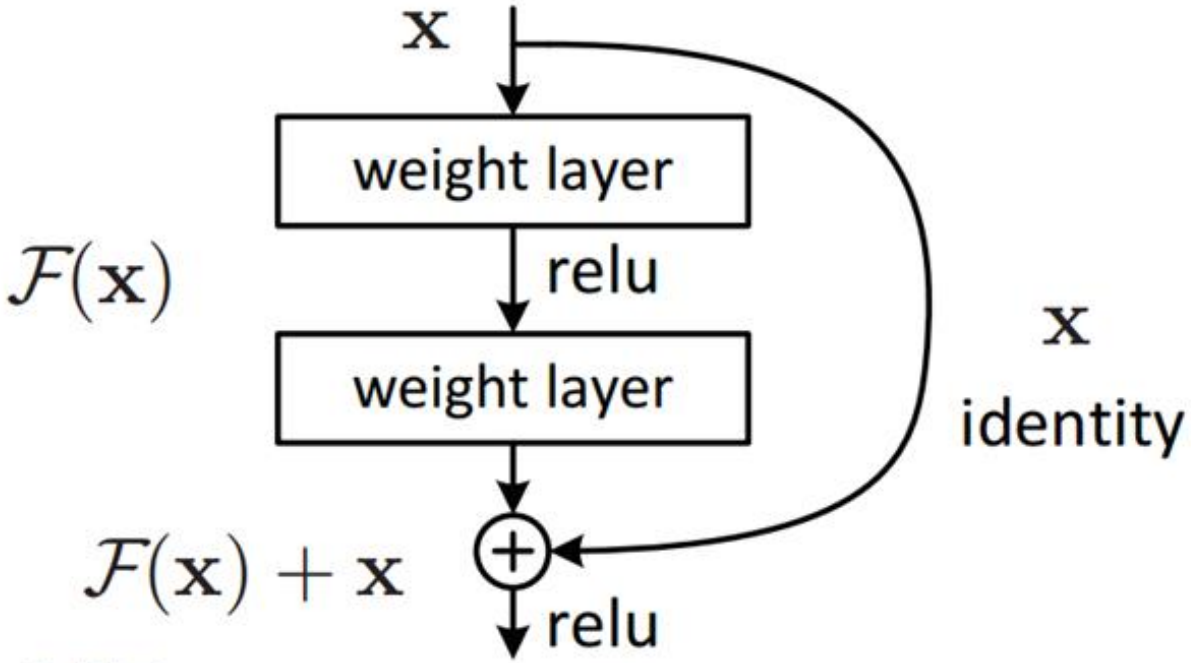
Residual Unit



A residual block

Residual Unit

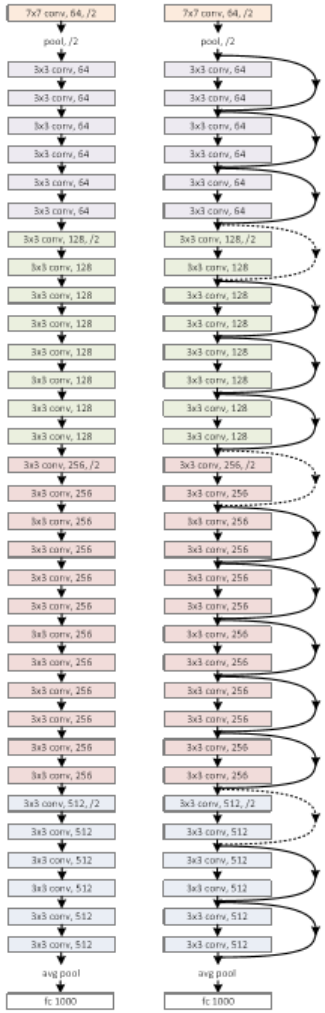
The inputs of a lower layer is made available to a node in a higher layer.



A residual block

Network “Design”

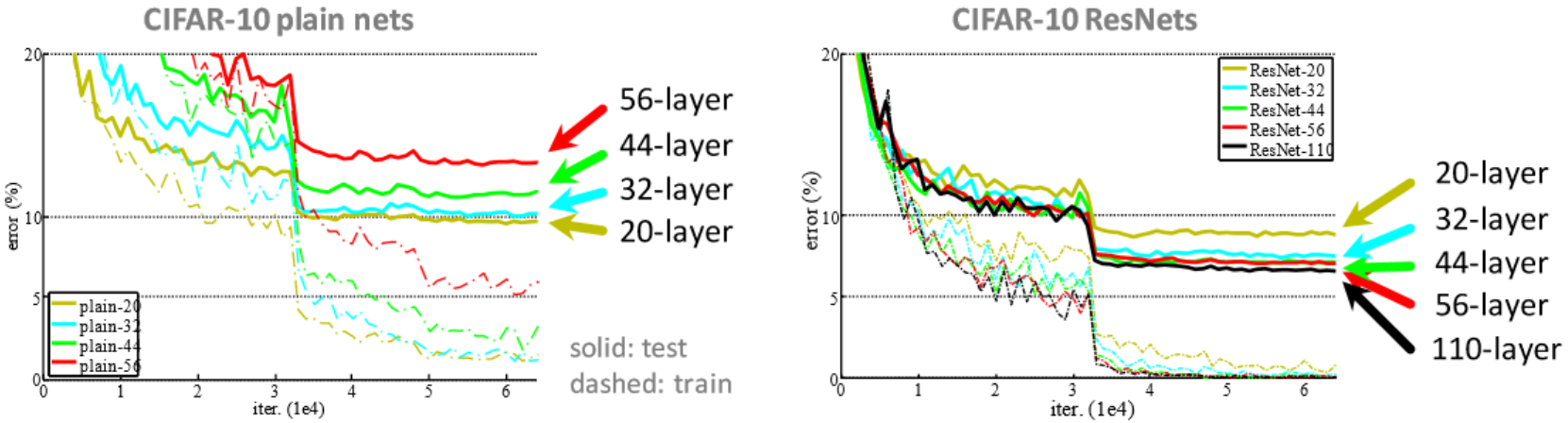
plain net



ResNet

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

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.