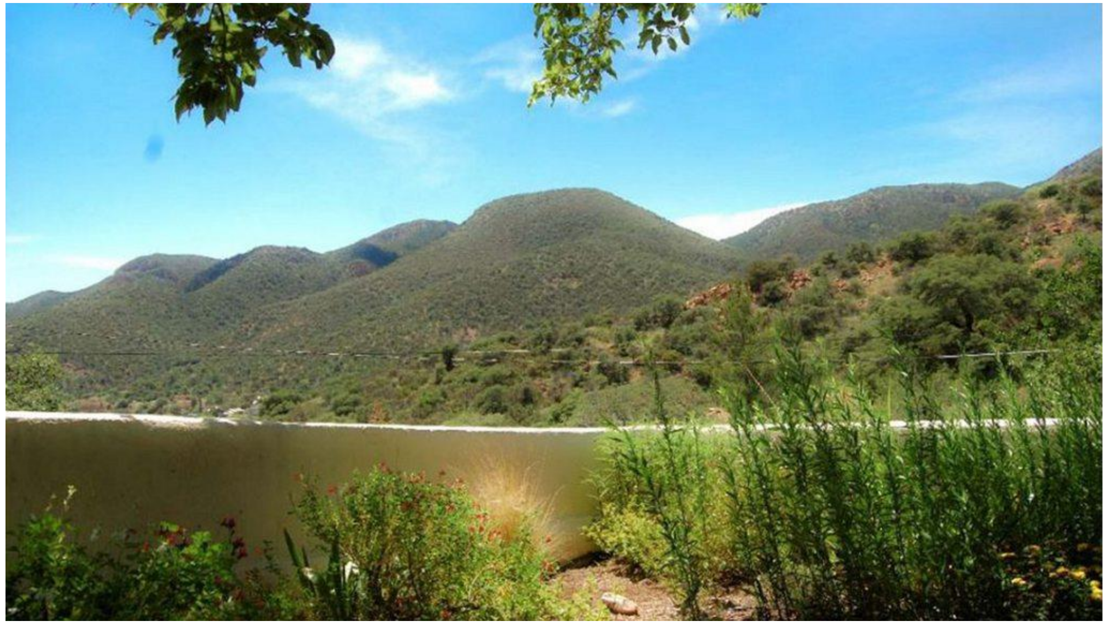
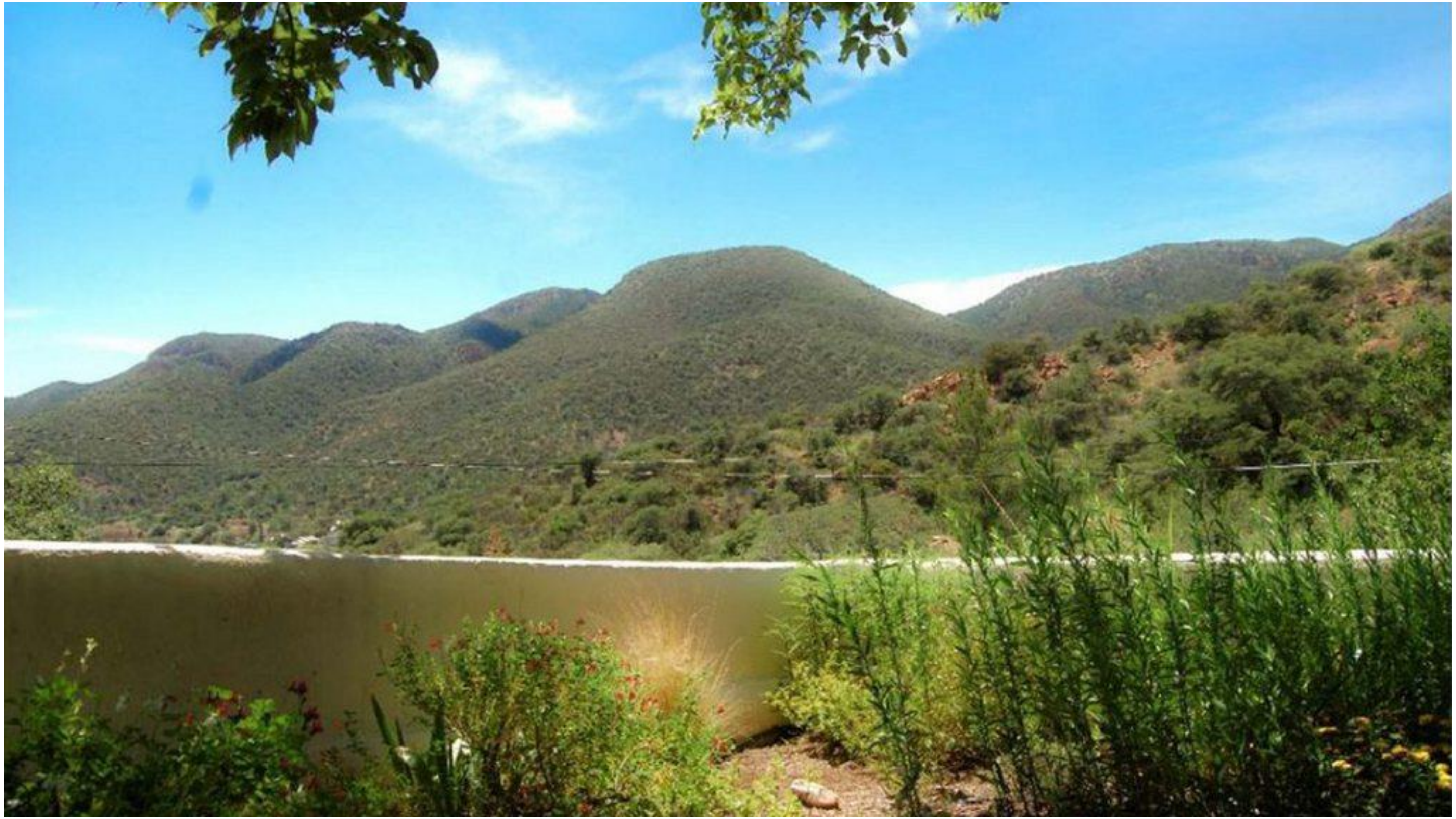
Let's look at some lakefront property



*actually fences / walls



Project 4: Scene Recognition with Deep Learning

CS 4476/6476

Fall 2022

Brief

- Due: Check Canvas for up to date information
- Project materials including report template: GitHub
- Hand-in: Gradescope
- Required files: <your_gt_username>.zip, <your_gt_username>_proj4.pdf

Overview

In this project, you will design and train deep convolutional networks for scene recognition. In Part 1, you will train a simple network from scratch. In Part 2, you will implement a few modifications on top of the base architecture from Part 1 to increase recognition accuracy to ∼55%. In Part 3, you will instead *fine-tune* a pre-trained deep network to achieve more than 80% accuracy on the task. We will use the pre-trained ResNet architecture which was not trained to recognize scenes at all. Finally, we will explore multi-label prediction of scene attributes in Part 4.

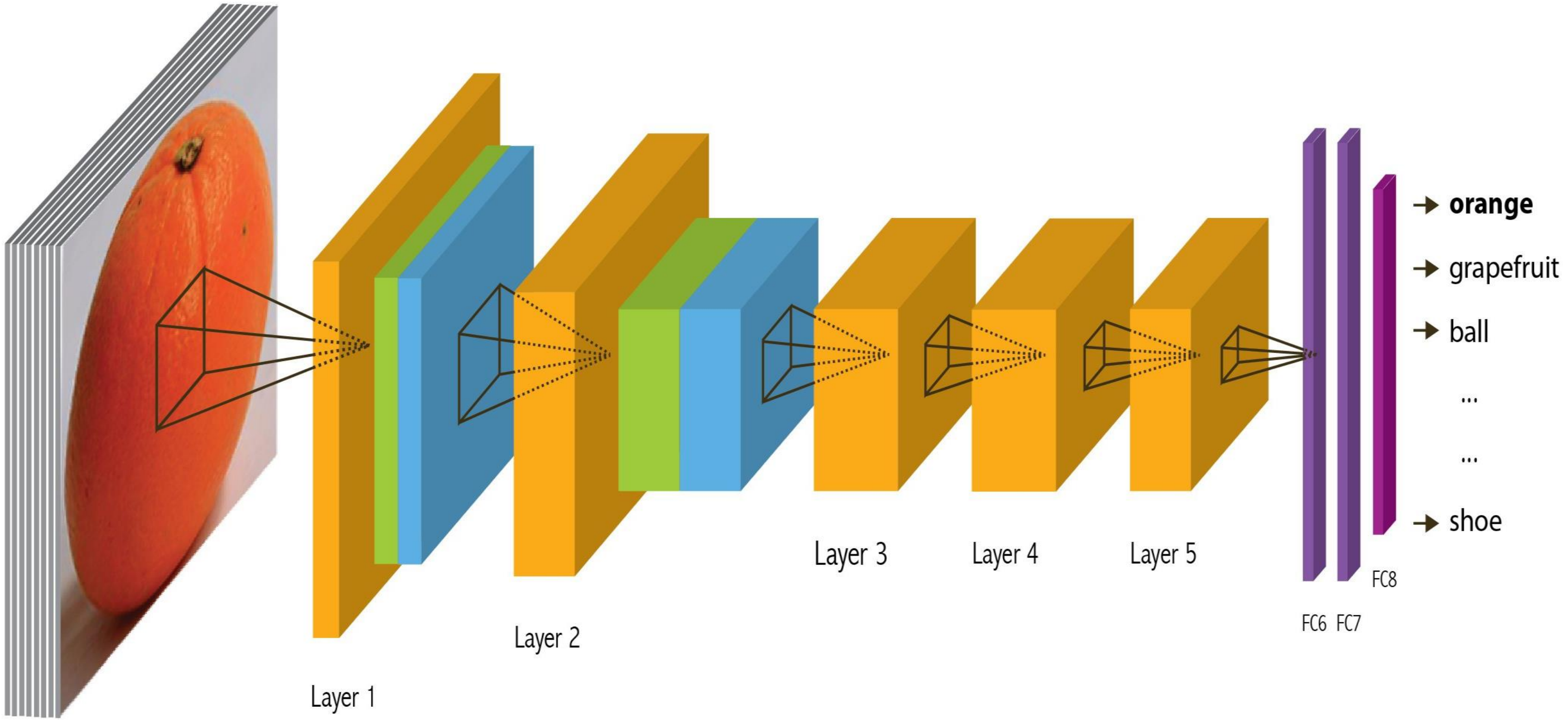
These different approaches (starting the training from scratch or fine-tuning) represent the most common approach to recognition problems in computer vision today–train a deep network from scratch if you have enough data (it's not always obvious whether or not you do), and if you cannot then fine-tune a pre-trained network instead. A GPU is not necessary for this project, but you can use Google Colab to help speed up training. Learn more about Colab here.

History of ideas in recognition

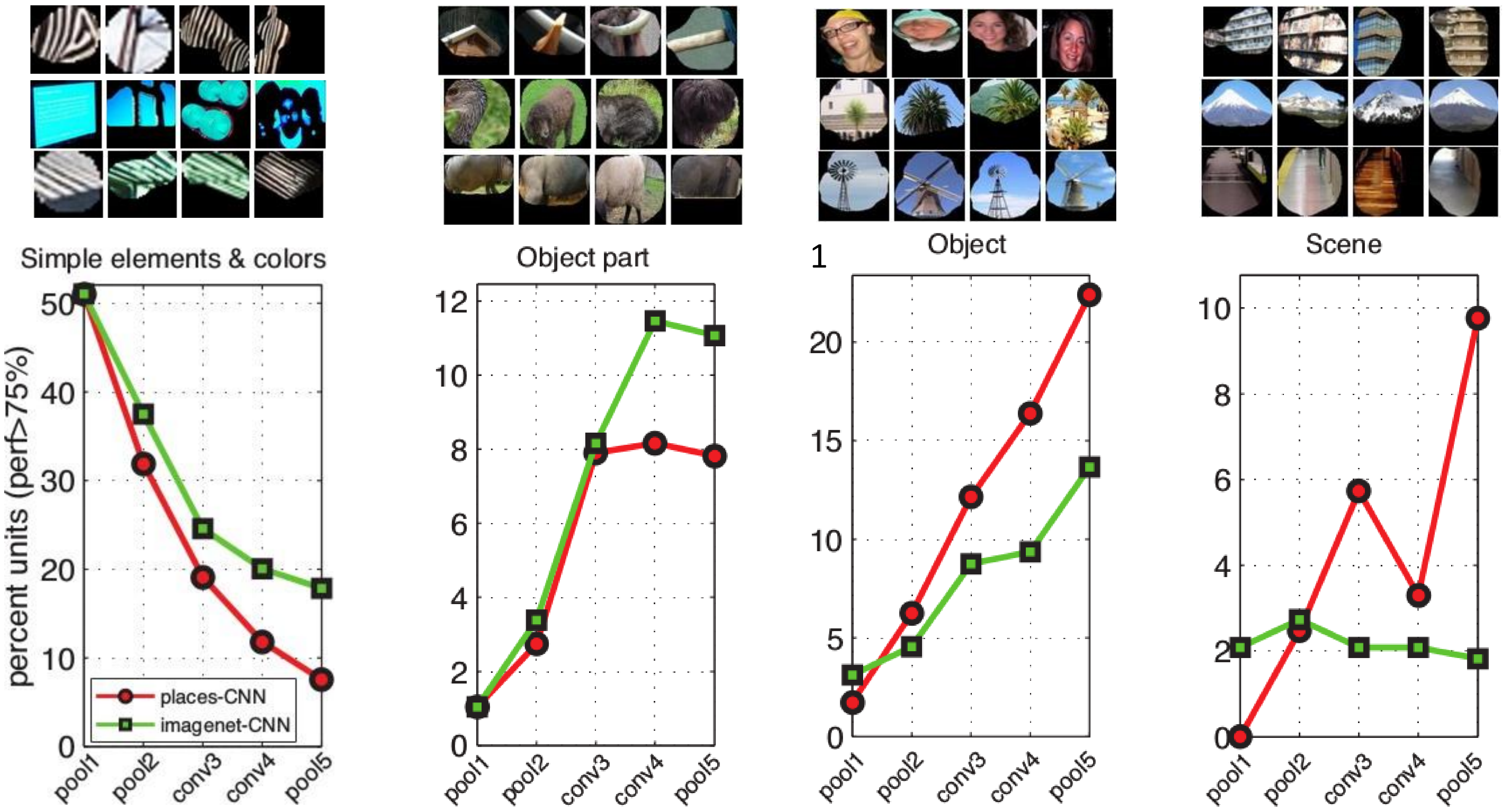
- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- Mid-1990s: sliding window approaches
- Late 1990s: local features
- Early 2000s: parts-and-shape models
- Mid-2000s: bags of features
- Present trends: combination of local and global methods, context, *deep learning*

Svetlana Lazebnik

Recap: Convolutional Network, AlexNet



Recap: Convolutional Network Interpretation



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

Beyond AlexNet

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan & Andrew Zisserman 2015

These are the “VGG” networks.
“Perceptual Loss” in generative deep learning refers to these networks

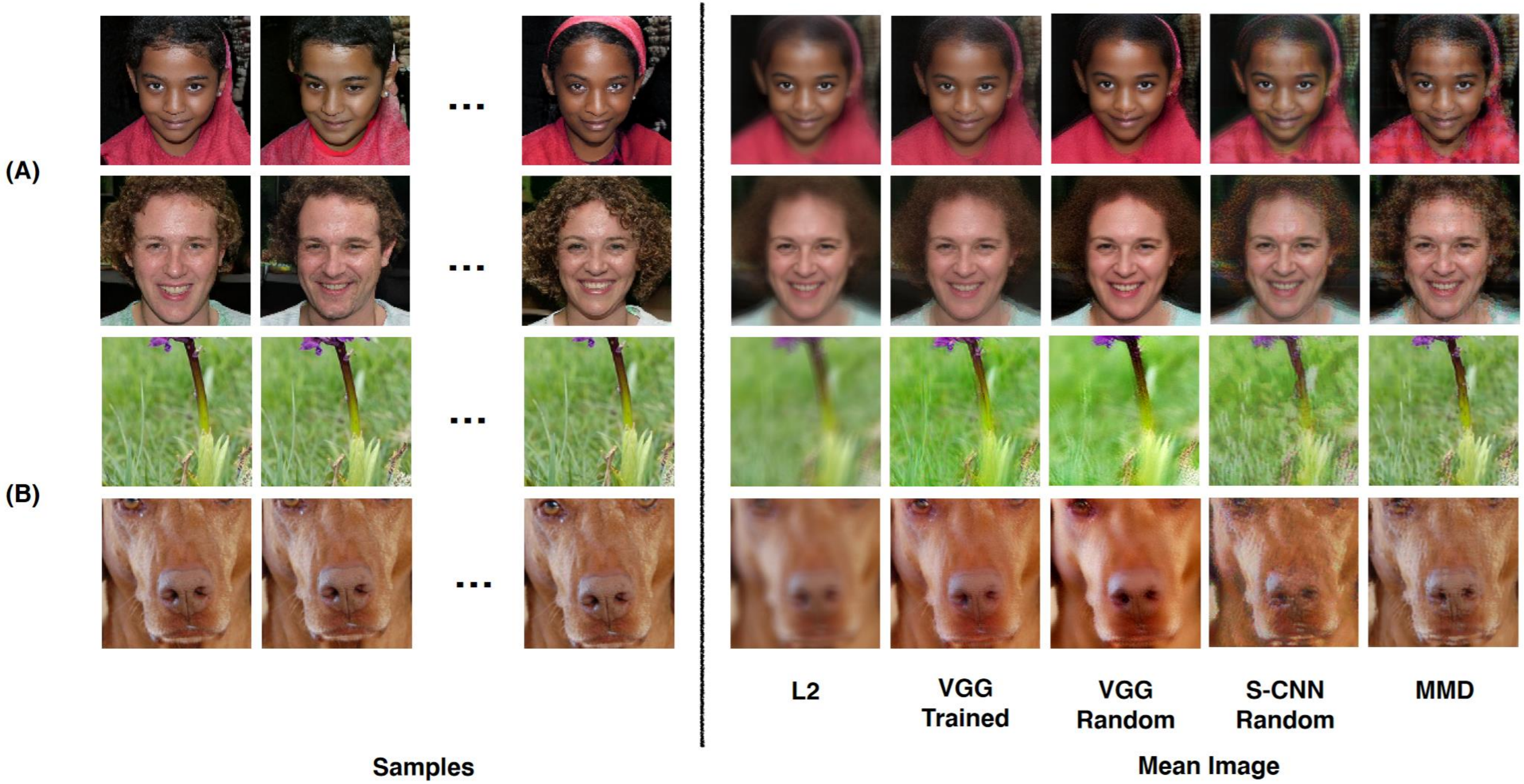
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	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**



"VGG" networks are commonly used as the basis for "Perceptual Loss".
The images on the right are as close as possible to all images on the left in various feature spaces.

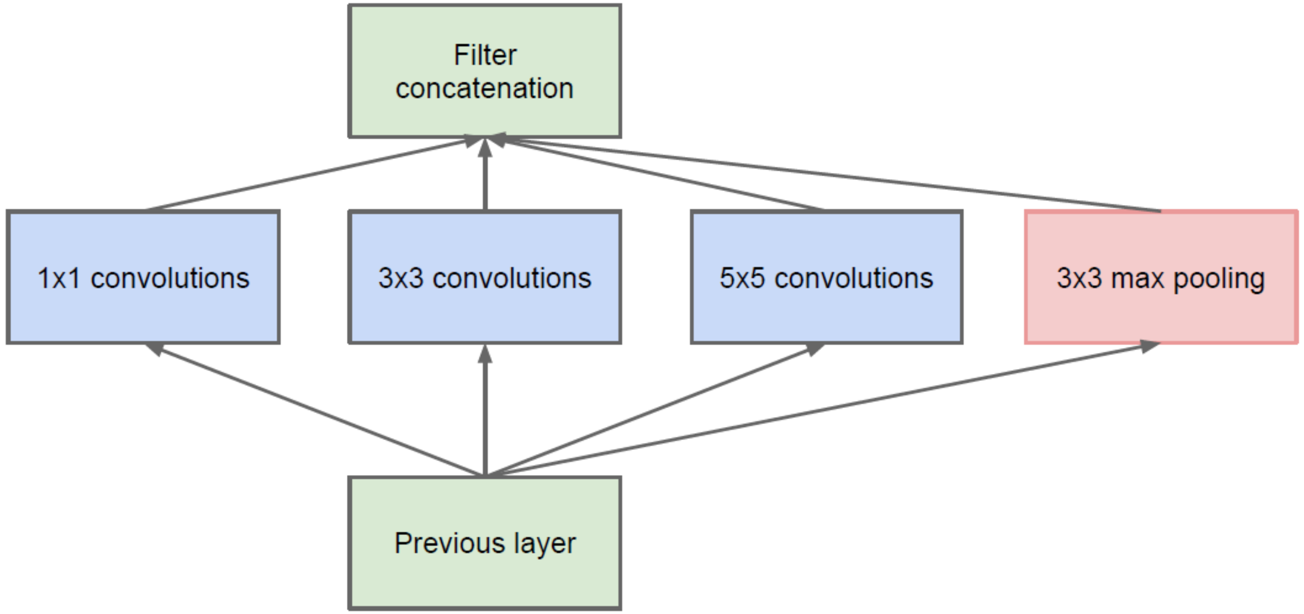
Understanding and Simplifying Perceptual Distances. Dan Amir and Yair Weiss. CVPR 2021

Going Deeper with Convolutions

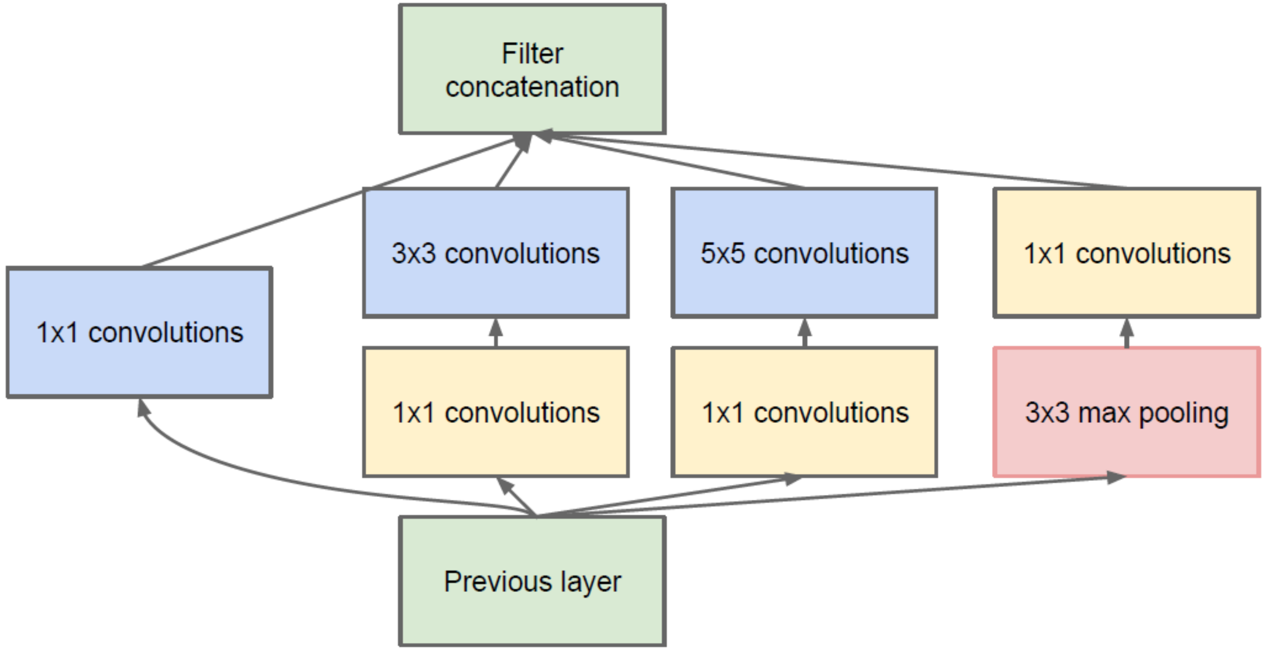
Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed,
Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich
2015

This is the “Inception” architecture or “GoogLeNet”

***The architecture blocks are called “Inception” modules**
and the collection of them into a particular net is “GoogLeNet”



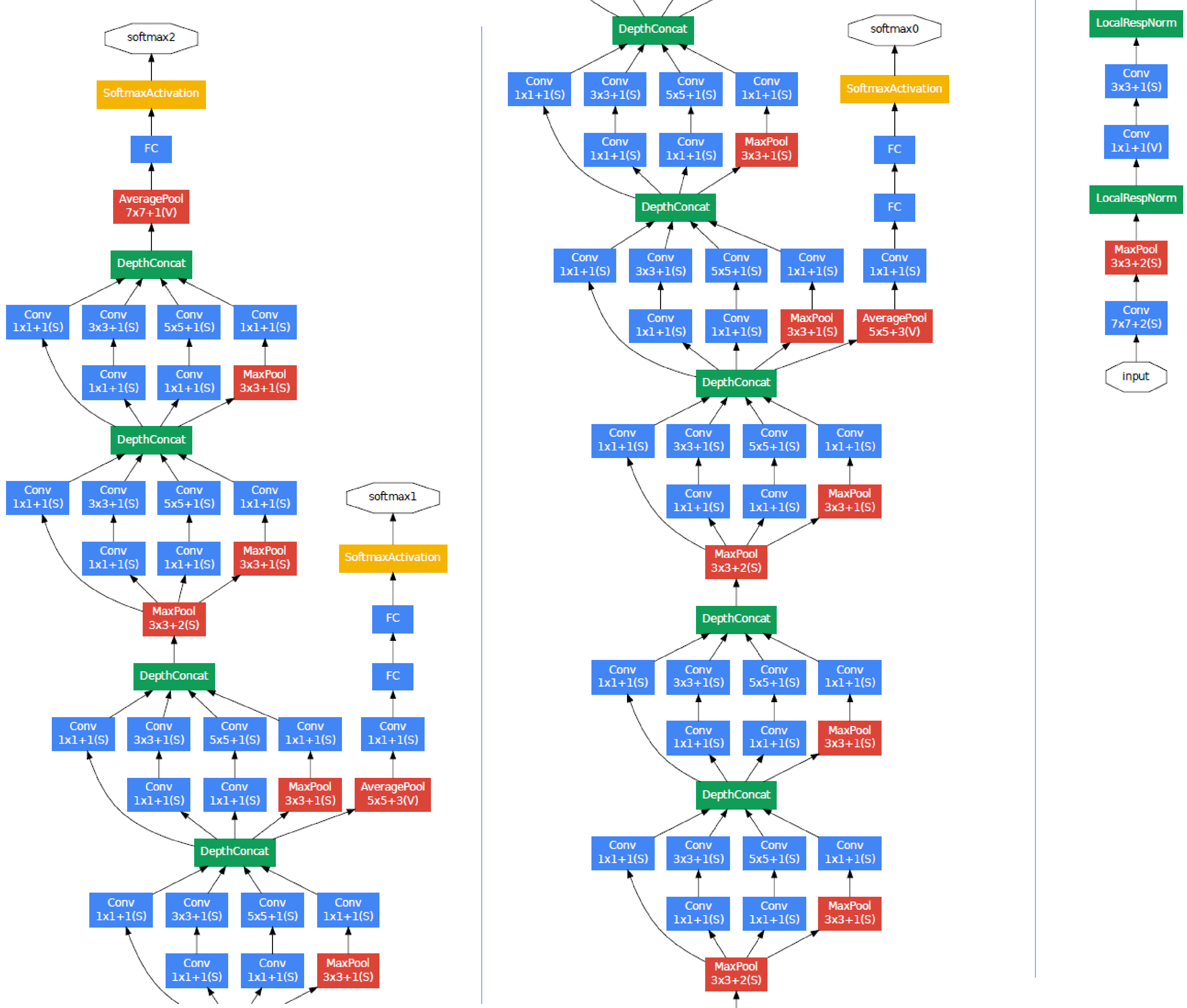
(a) Inception module, naïve version



(b) Inception module with dimensionality reduction

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Only 6.8 million parameters. AlexNet ~60 million, VGG up to 138 million

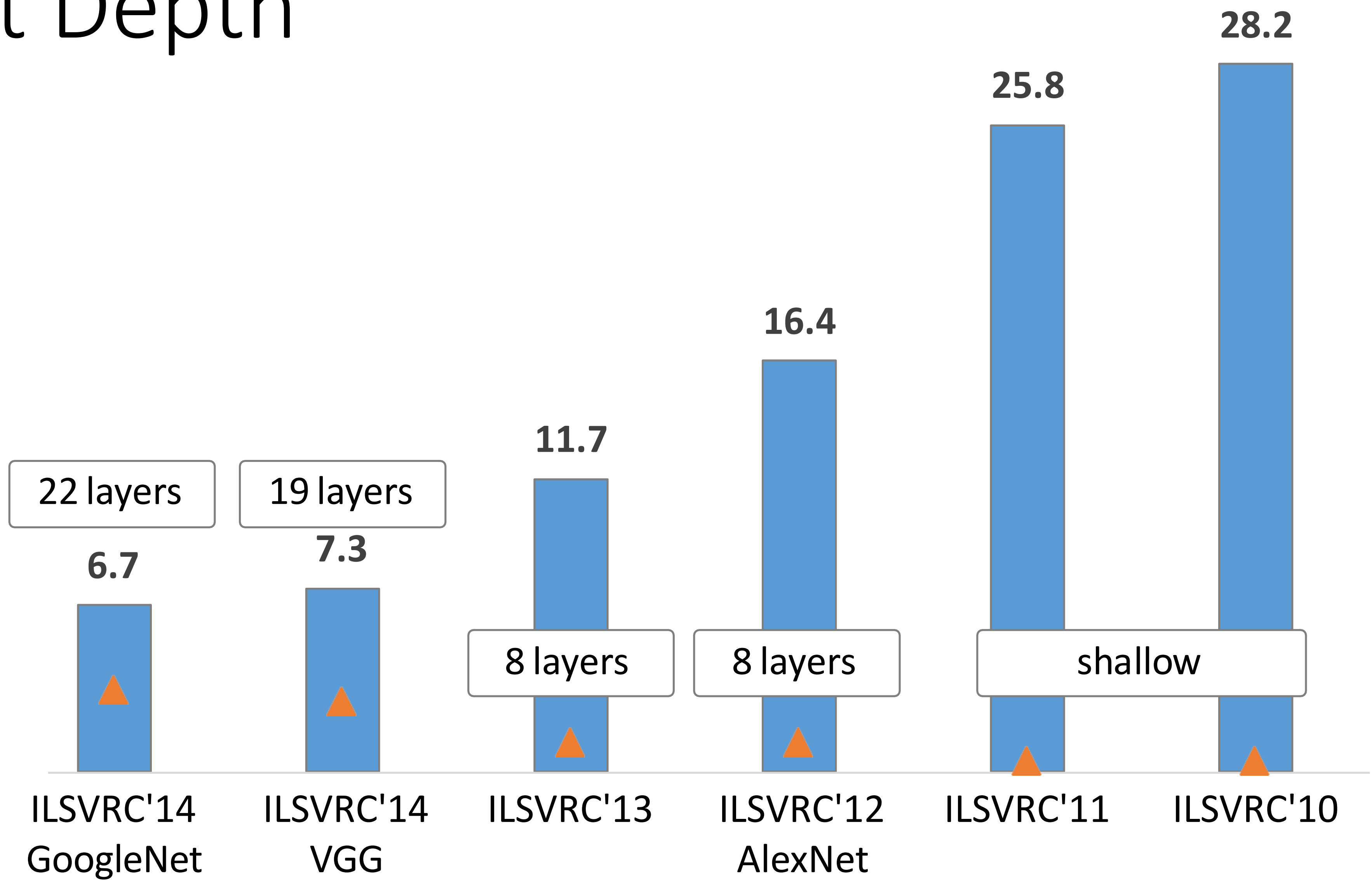

softmax2
SoftmaxActivation
FC
AveragePool 7x7+1(V)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
MaxPool 3x3+2(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
softmax1
SoftmaxActivation
FC
FC
Conv 1x1+1(S)
AveragePool 5x5+3(V)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
softmax0
SoftmaxActivation
FC
FC
Conv 1x1+1(S)
AveragePool 5x5+3(V)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
MaxPool 3x3+2(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
MaxPool 3x3+2(S)
LocalRespNorm
Conv 3x3+1(S)
Conv 1x1+1(V)
LocalRespNorm
MaxPool 3x3+2(S)
Conv 7x7+2(S)
input

Team	**Year**	**Place**	**Error (top-5)**	**Uses external data**
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

Table 2: Classification performance.

Number of models	**Number of Crops**	**Cost**	**Top-5 error**	**compared to base**
1	1	1	10.07%	base
1	10	10	9.15%	-0.92%
1	144	144	7.89%	-2.18%
7	1	7	8.09%	-1.98%
7	10	70	7.62%	-2.45%
7	144	1008	6.67%	-3.45%

ConvNet Depth



ImageNet Classification top-5 error (%)

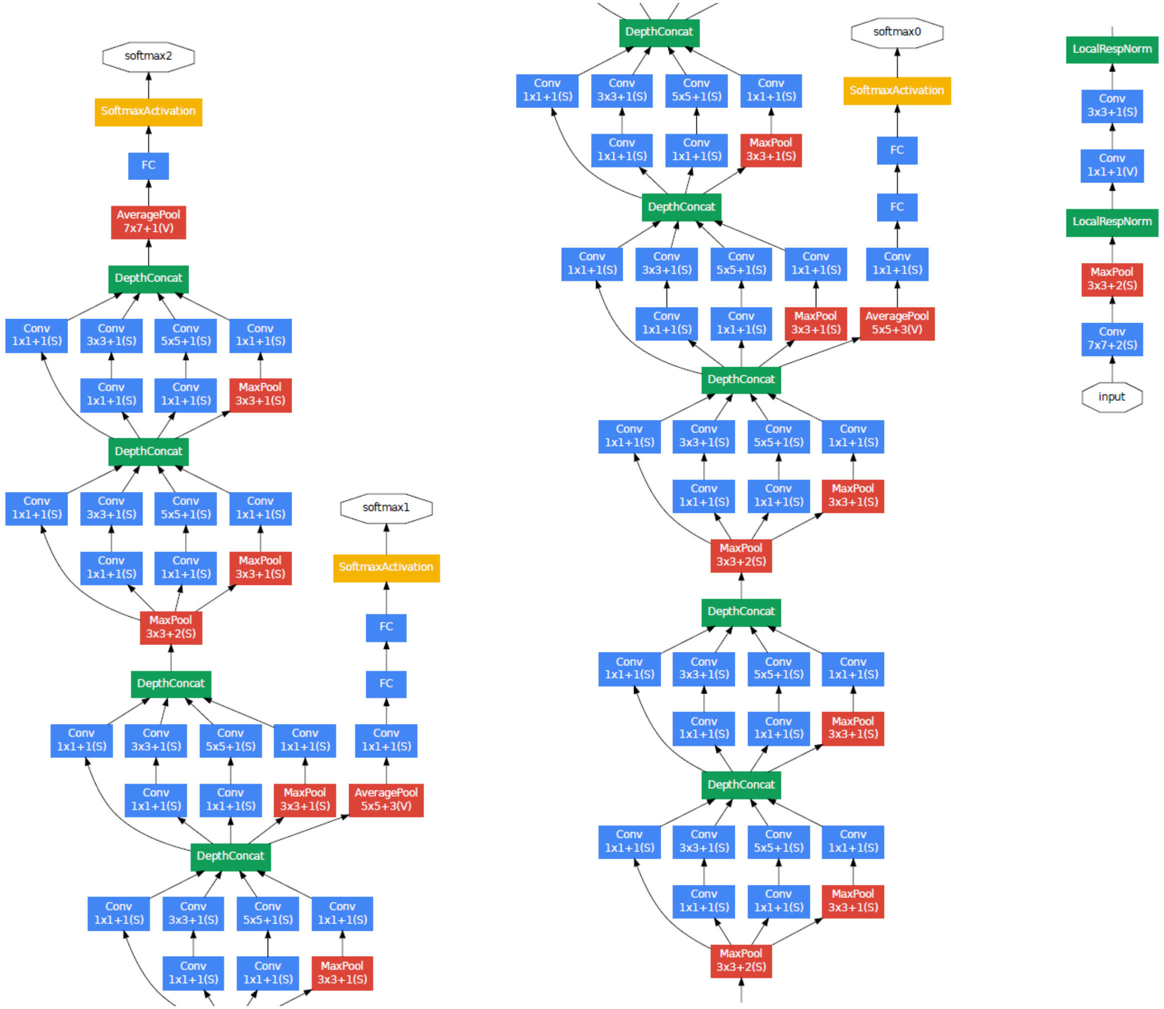
Recap: Beyond AlexNet

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

VGG



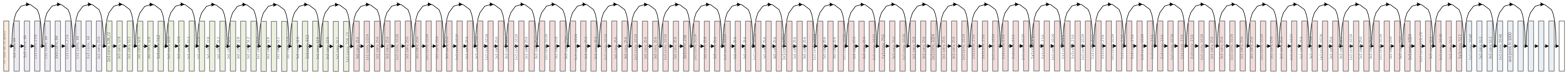
GoogLeNet

Deep Residual Learning for Image Recognition

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

work done at

Microsoft Research Asia



Cited 136,837 times as of 10/27/2022.

	Publication	h5-index	h5-median
1.	Nature	444	667
2.	The New England Journal of Medicine	432	780
3.	Science	401	614
4.	IEEE/CVF Conference on Computer Vision and Pattern Recognition	389	627
5.	The Lancet	354	635
6.	Advanced Materials	312	418
7.	Nature Communications	307	428
8.	Cell	300	505
9.	International Conference on Learning Representations	286	533
10.	Neural Information Processing Systems	278	436
11.	JAMA	267	425
12.	Chemical Reviews	265	444
13.	Proceedings of the National Academy of Sciences	256	364
14.	Angewandte Chemie	245	332
15.	Chemical Society Reviews	244	386
16.	Journal of the American Chemical Society	242	344
17.	IEEE/CVF International Conference on Computer Vision	239	415
18.	Nucleic Acids Research	238	550
19.	International Conference on Machine Learning	237	421

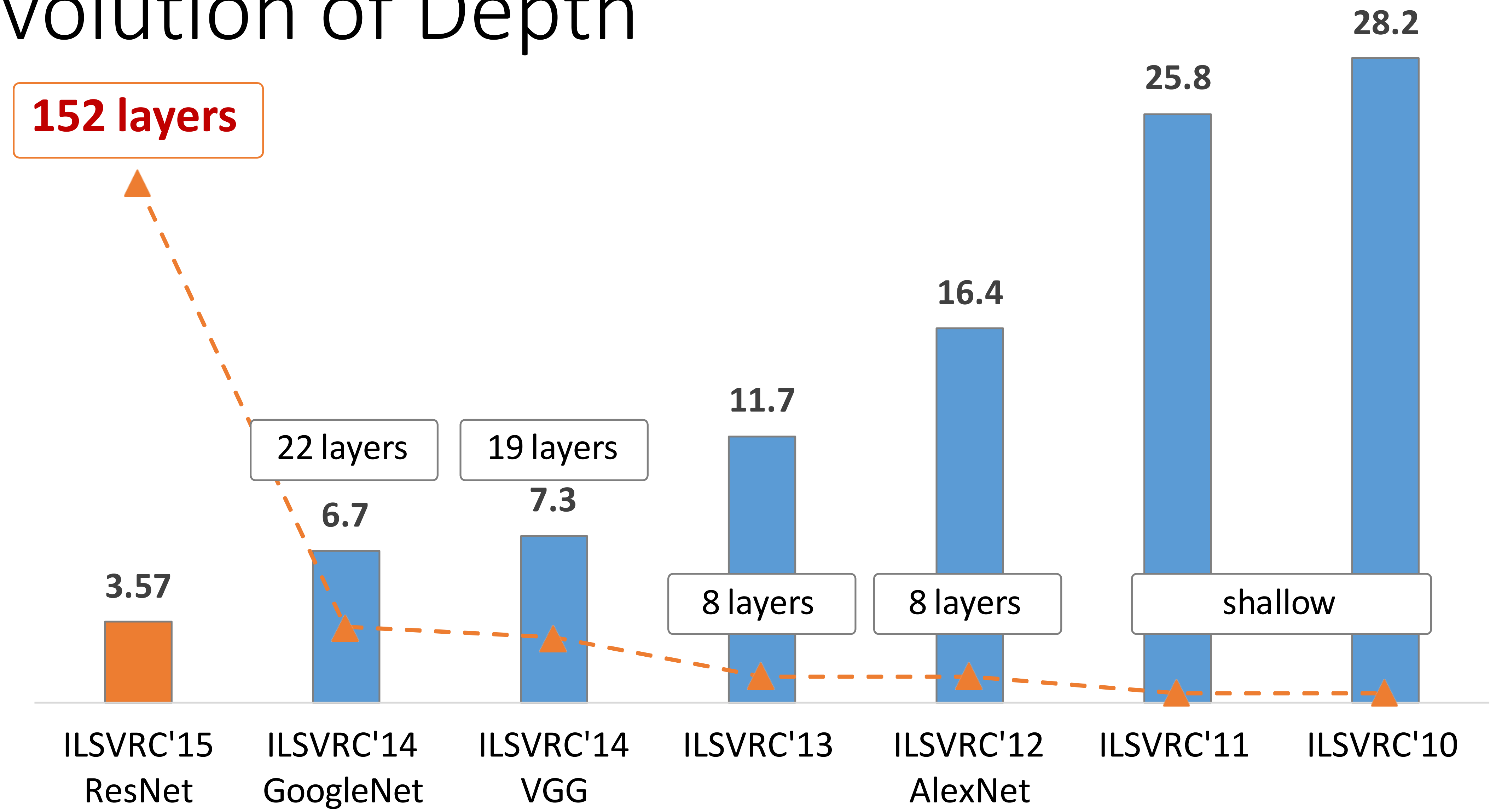
ResNet @ ILSVRC & COCO 2015 Competitions

1st places in all five main tracks

- ImageNet Classification: "*Ultra-deep*" 152-layer nets
- ImageNet Detection: 16% better than 2nd
- ImageNet Localization: 27% better than 2nd
- COCO Detection: 11% better than 2nd
- COCO Segmentation: 12% better than 2nd

*improvements are relative numbers

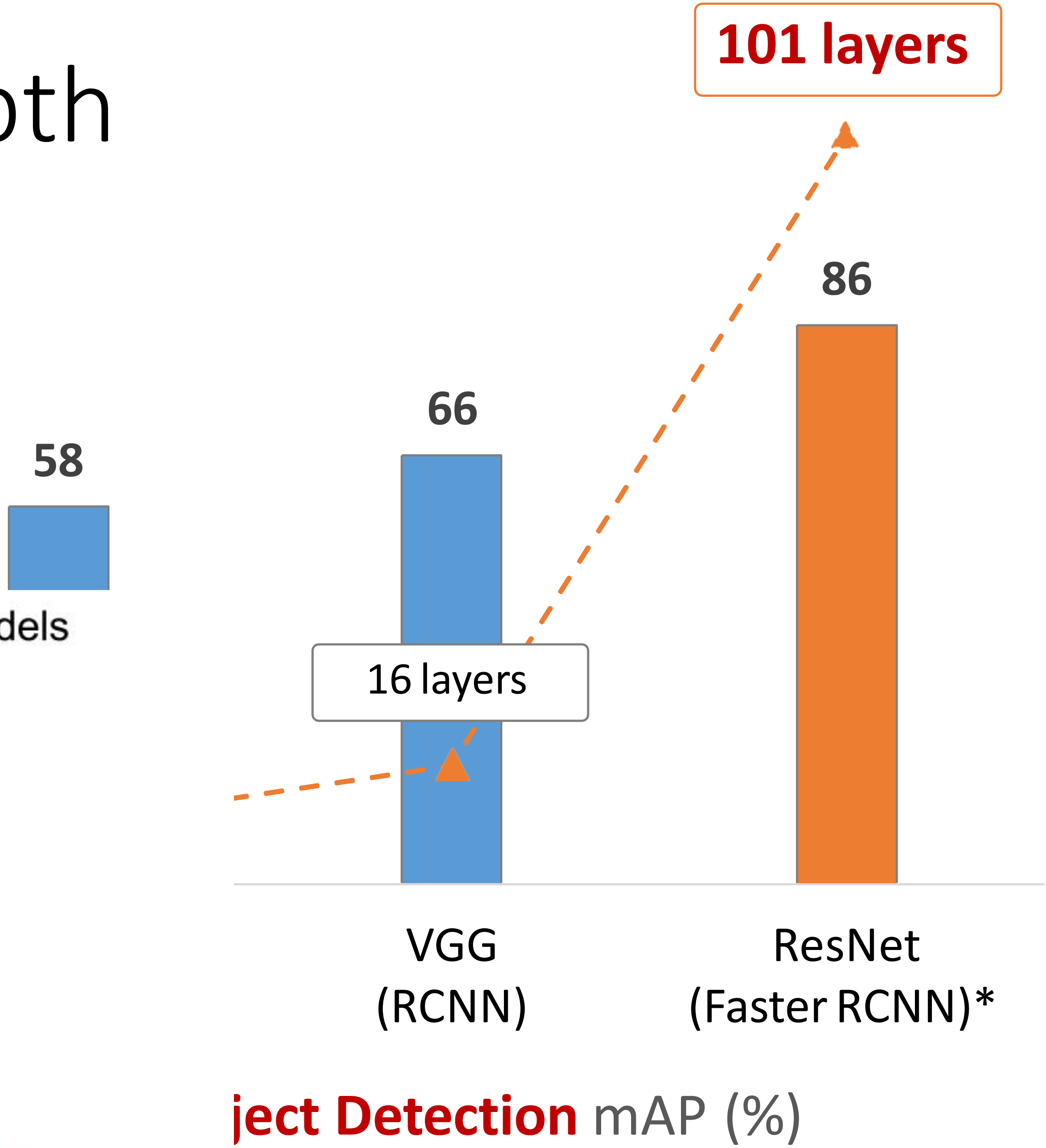
Revolution of Depth



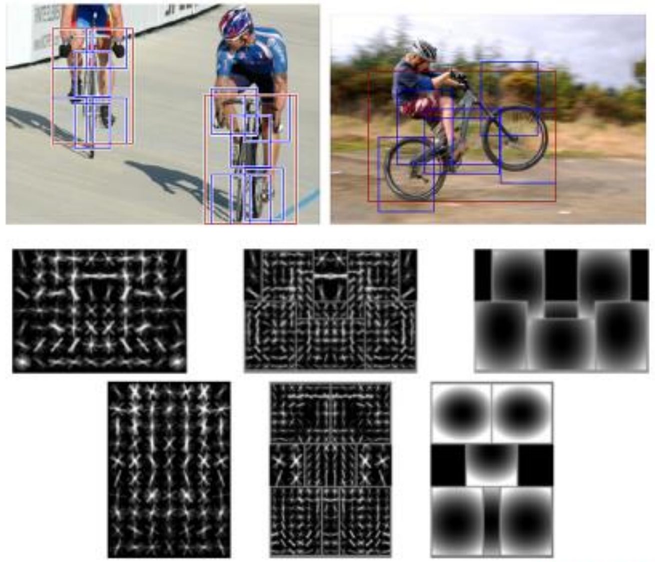
ImageNet Classification top-5 error (%)

Revolution of Depth

Engines of
visual recognition



Discriminatively trained part-based models

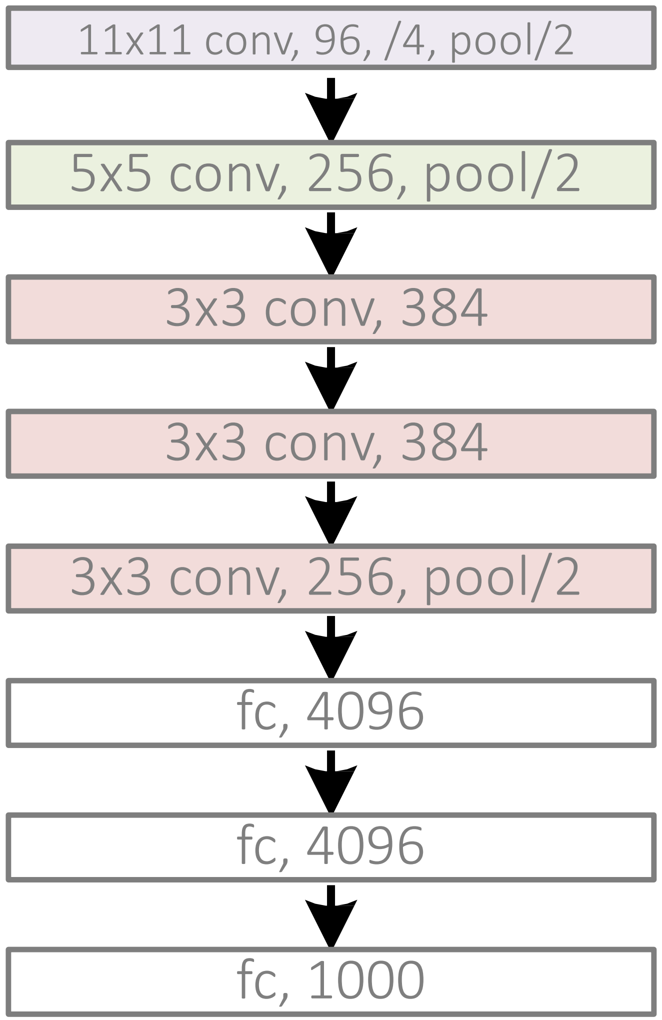


P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," PAMI 2009

*w/ other improvements & more data

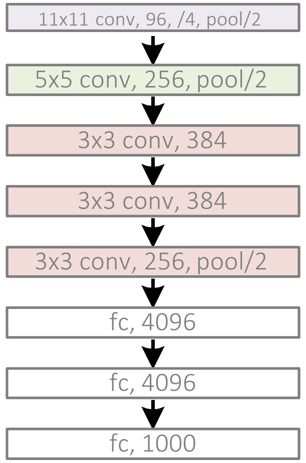
Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

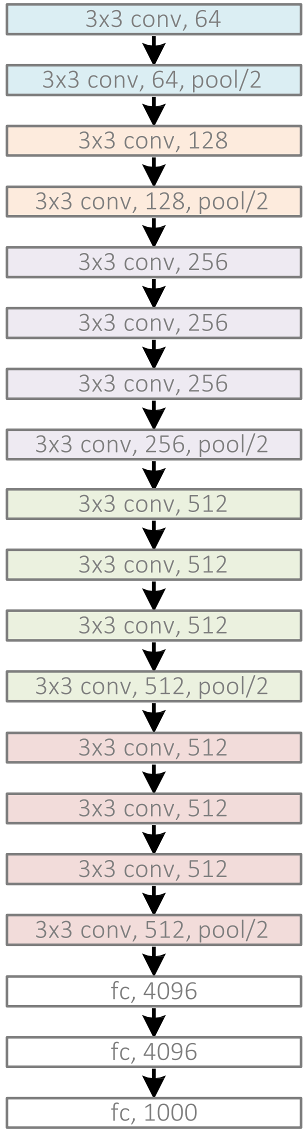


Revolution of Depth

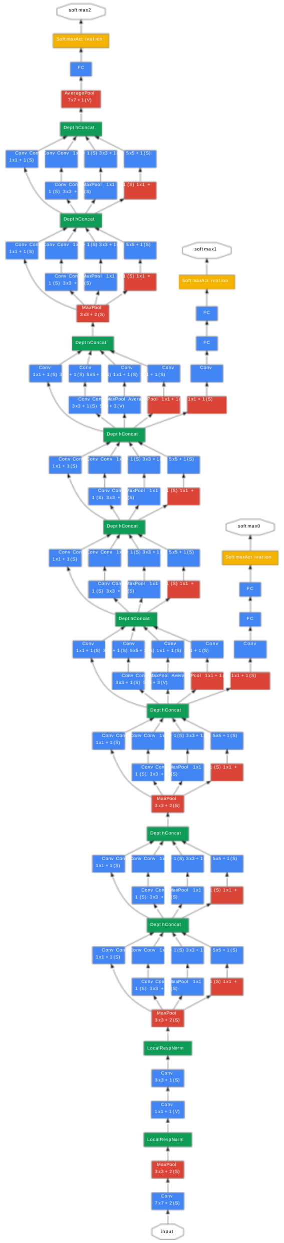
AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



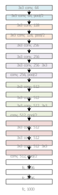
GoogleNet, 22 layers
(ILSVRC 2014)



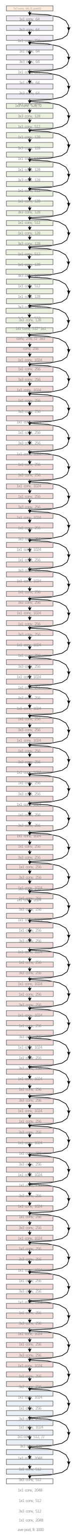
Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

VGG, 19 layers
(ILSVRC 2014)

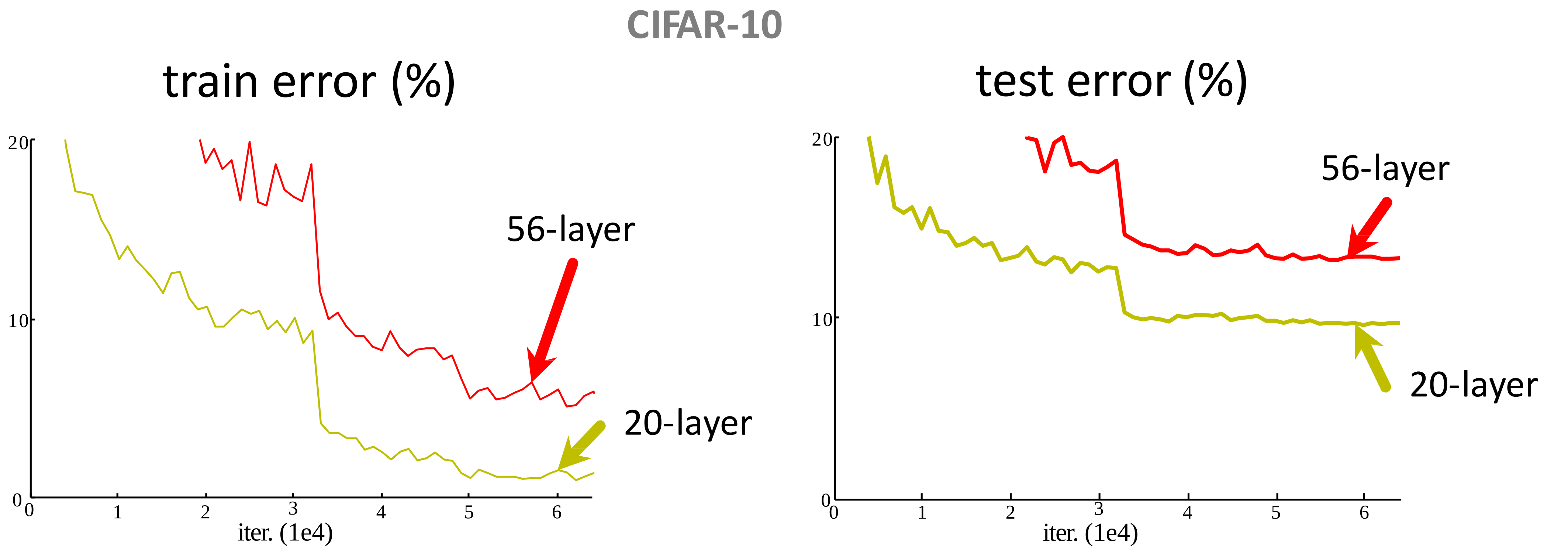


ResNet, 152 layers
(ILSVRC 2015)



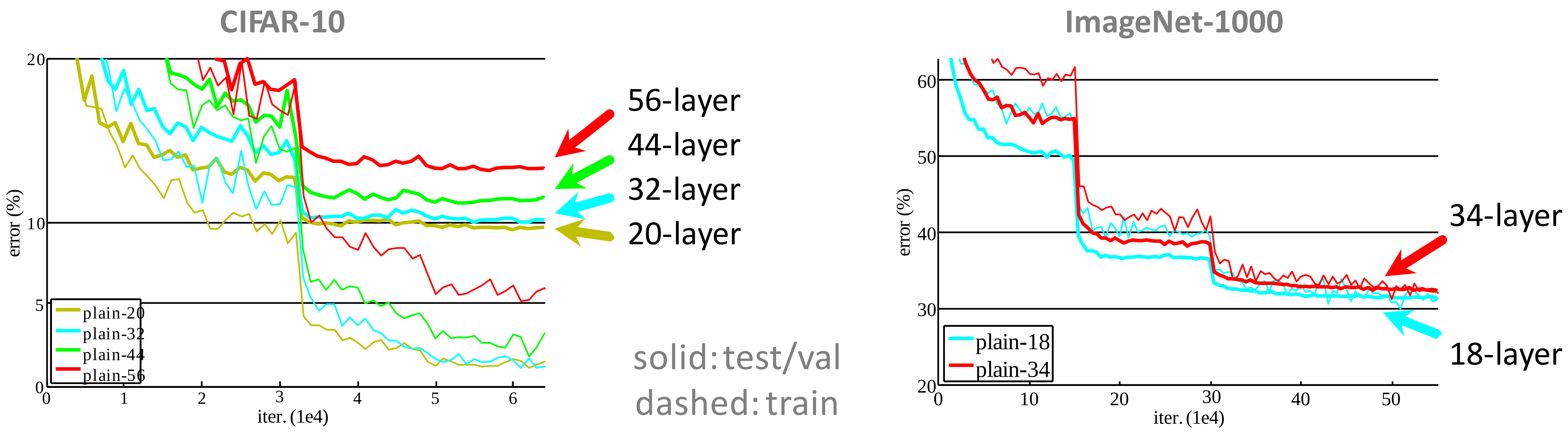
Is learning better networks
as simple as stacking more layers?

Simply stacking layers?



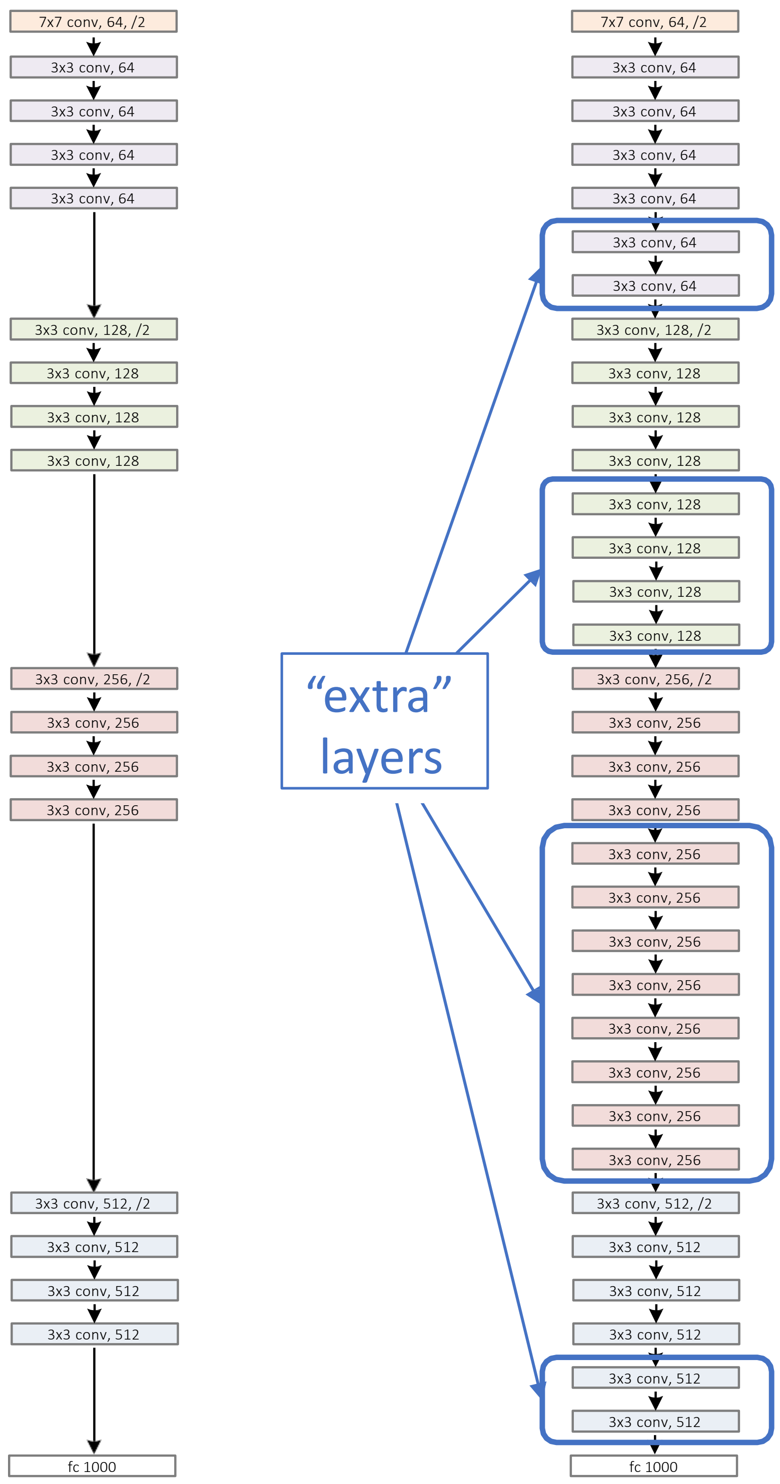
- *Plain* nets: stacking 3x3 conv layers...
- 56-layer net has **higher training error** and test error than 20-layer net

Simply stacking layers?



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

a shallower model (18 layers)

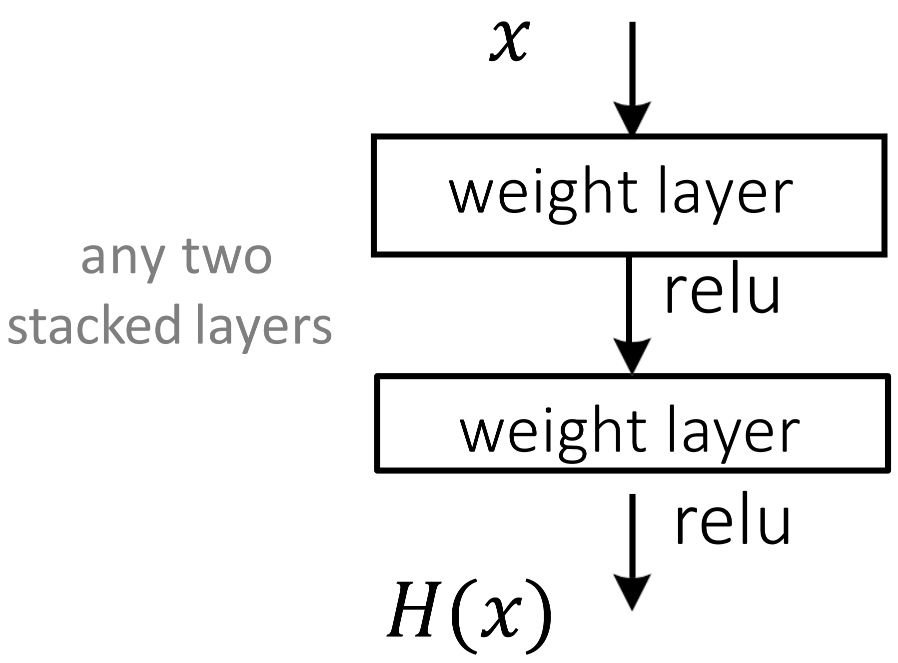


a deeper counterpart (34 layers)

- Richer solution space
- A deeper model should not have **higher training error**
- A solution *by construction*:
  - original layers: copied from a learned shallower model
  - extra layers: set as identity
  - at least the same training error
- Optimization difficulties: solvers cannot find the solution when going deeper…

Deep Residual Learning

- Plain net

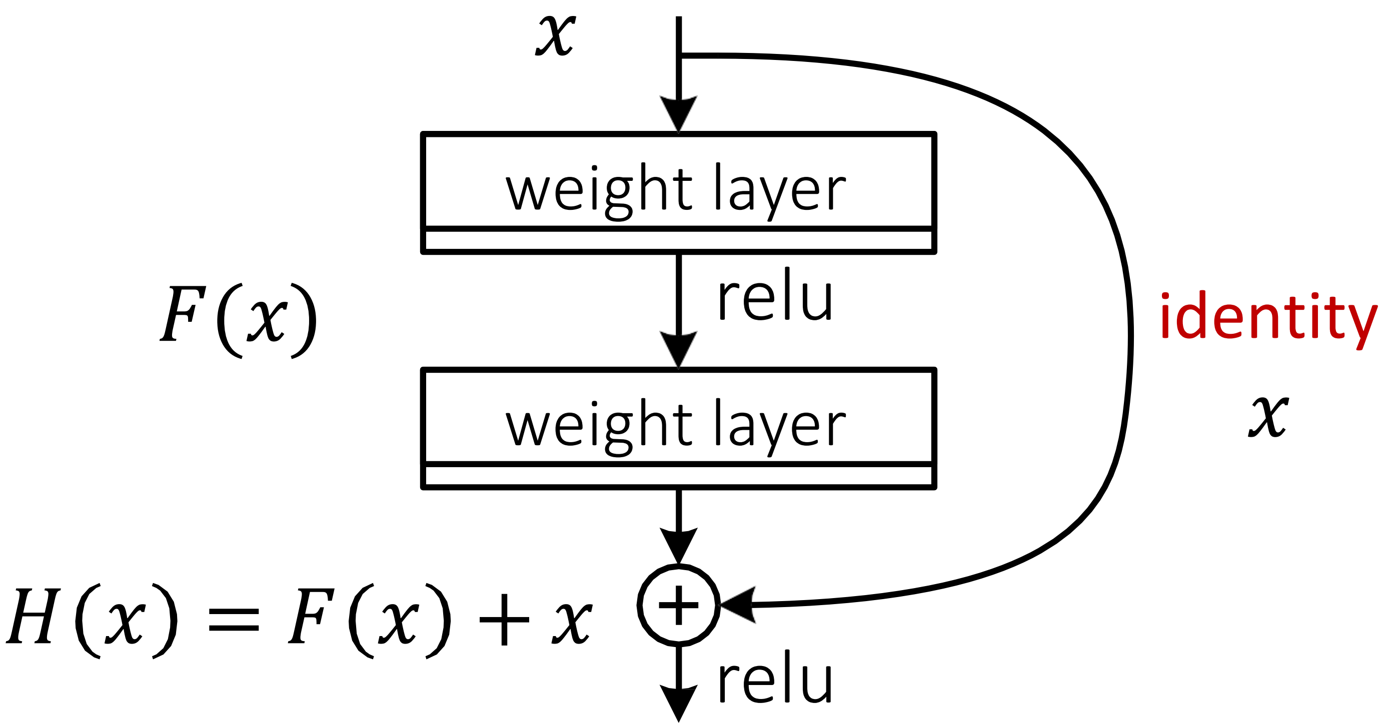


$H(x)$ is any desired mapping,

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

Deep Residual Learning

- Residual net



$H(x)$ is any desired mapping,

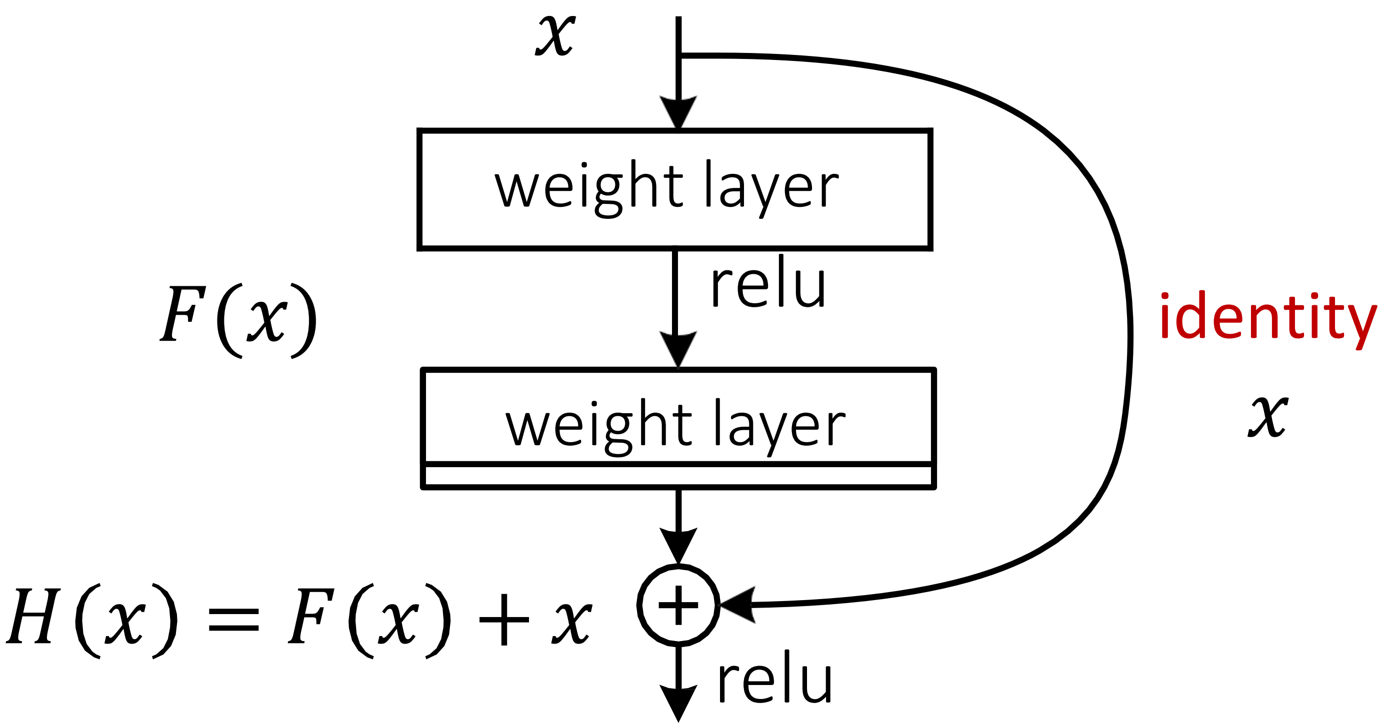
~~hope the 2 weight layers fit $H(x)$~~

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

let $H(x) = F(x) + x$

Deep Residual Learning

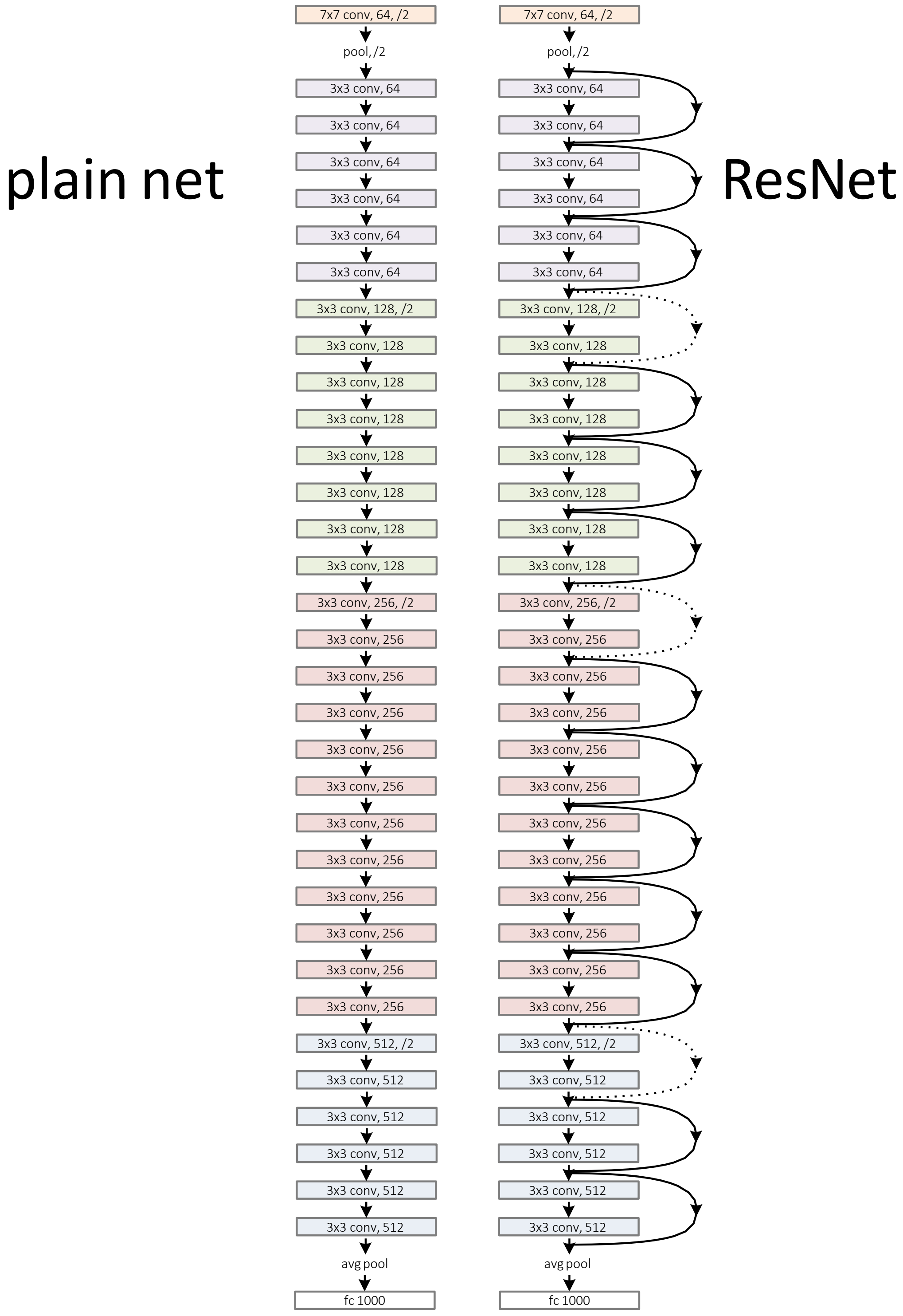
- $F(x)$ is a residual mapping w.r.t. identity



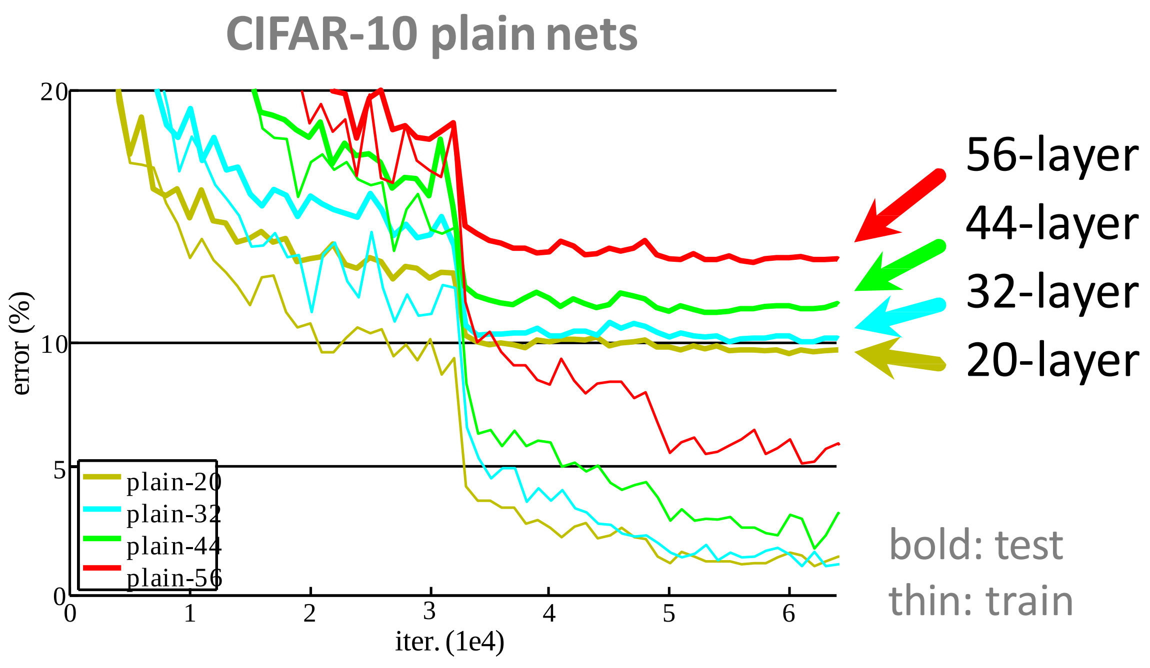
- If identity were optimal, easy to set weights as 0
- If optimal mapping is closer to identity, easier to find small fluctuations

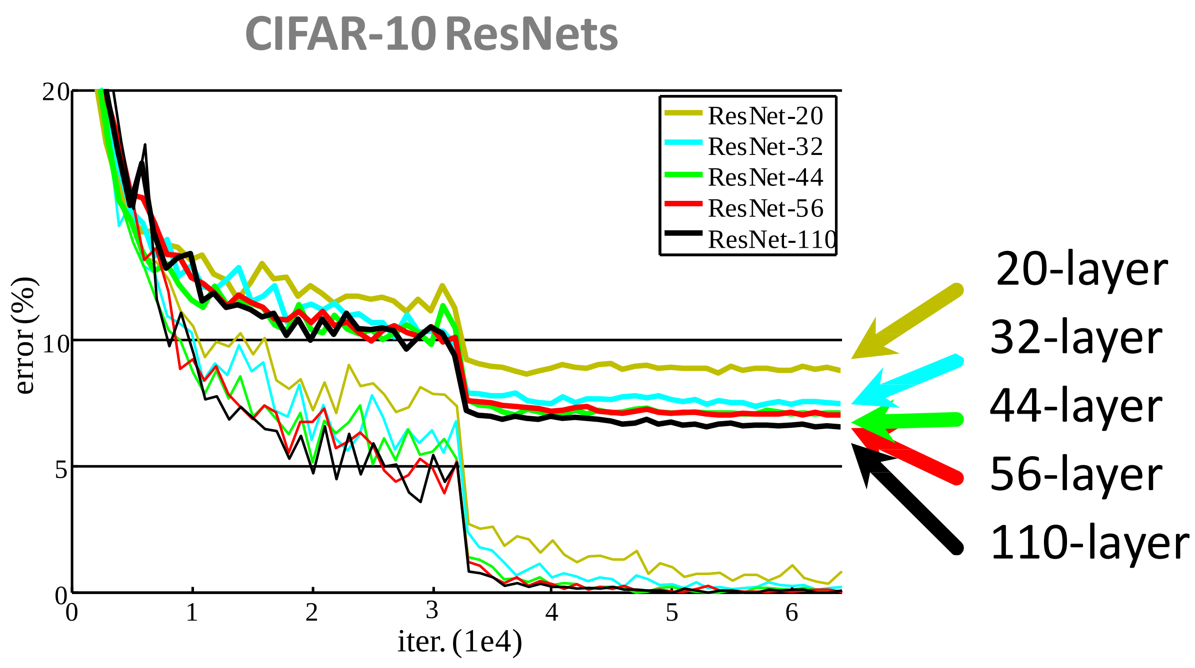
Network “Design”

- Keep it simple
- Our basic design (VGG-style)
    - all 3x3 conv (almost)
    - spatial size /2 => # filters x2
    - Simple design; just deep!



CIFAR-10 experiments

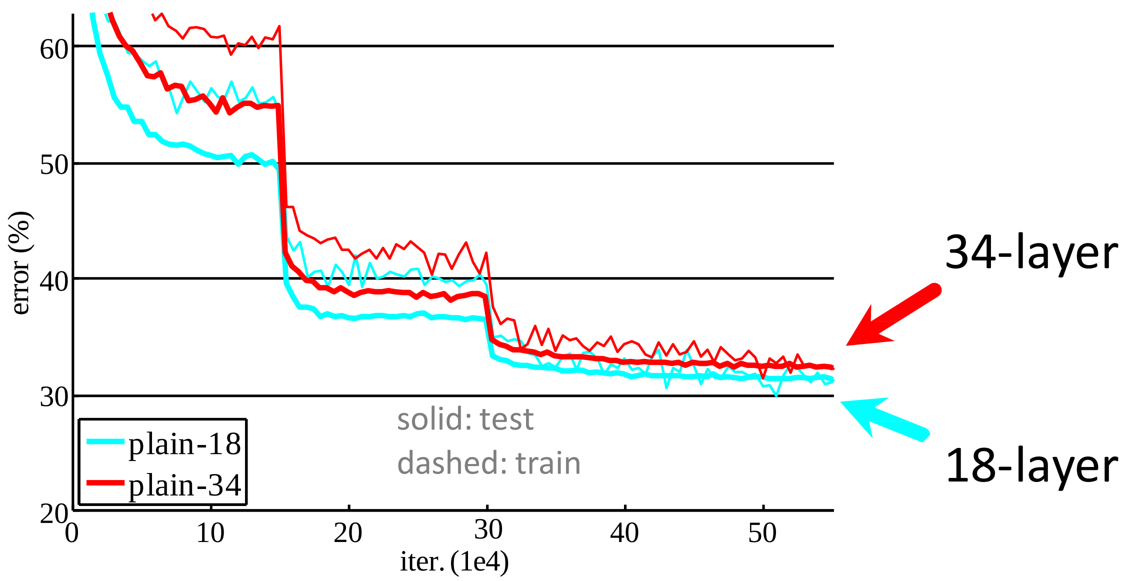




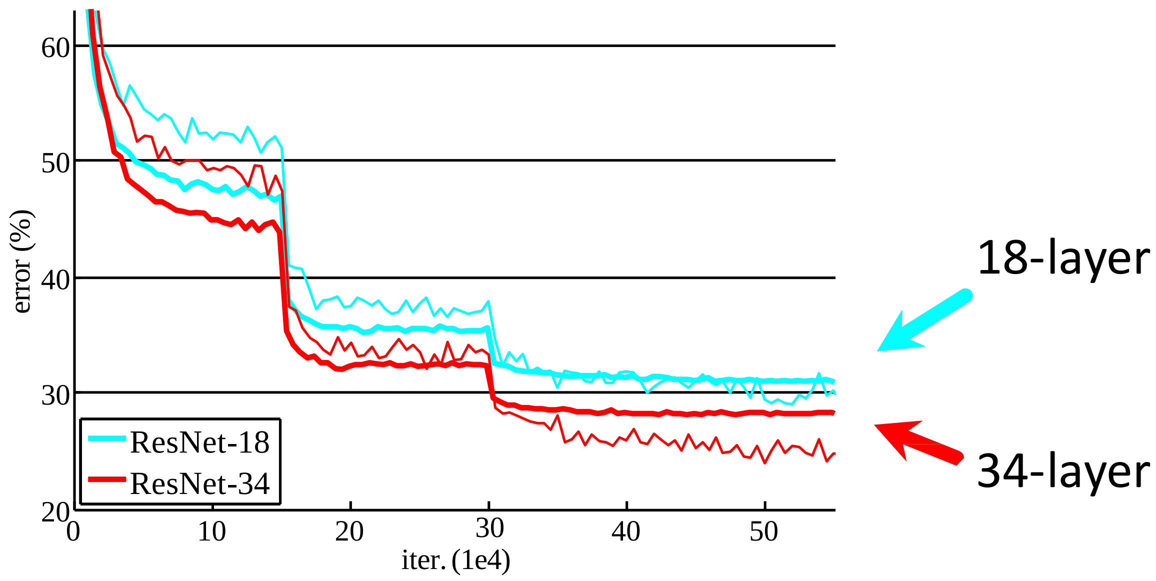
- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

ImageNet experiments

ImageNet plain nets



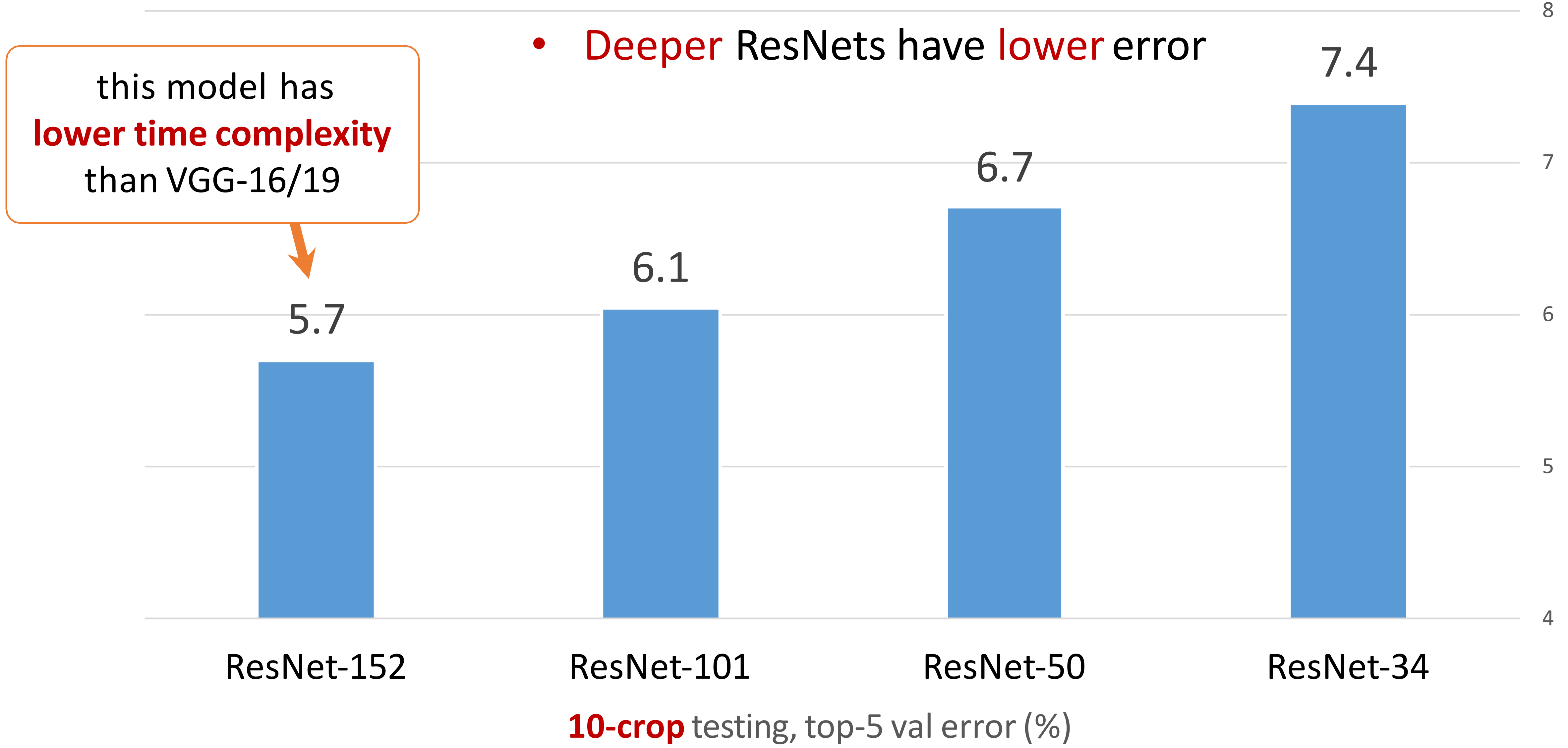
ImageNet ResNets



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

ImageNet experiments

- **Deeper** ResNets have **lower** error



Beyond classification

A treasure from ImageNet is on learning features.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015.

"Features matter." (quote [Girshick et al. 2014], the R-CNN paper)

task	2nd-place winner	ResNets	margin (relative)
ImageNet Localization (top-5 error)	12.0	9.0	**27%**
ImageNet Detection (mAP@.5)	53.6	62.1	**16%**
COCO Detection (mAP@.5:.95)	33.5	37.3	**11%**
COCO Segmentation (mAP@.5:.95)	25.1	28.2	**12%**

absolute 8.5% better!

- Our results are all based on ResNet-101
- Our features are well transferrable

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

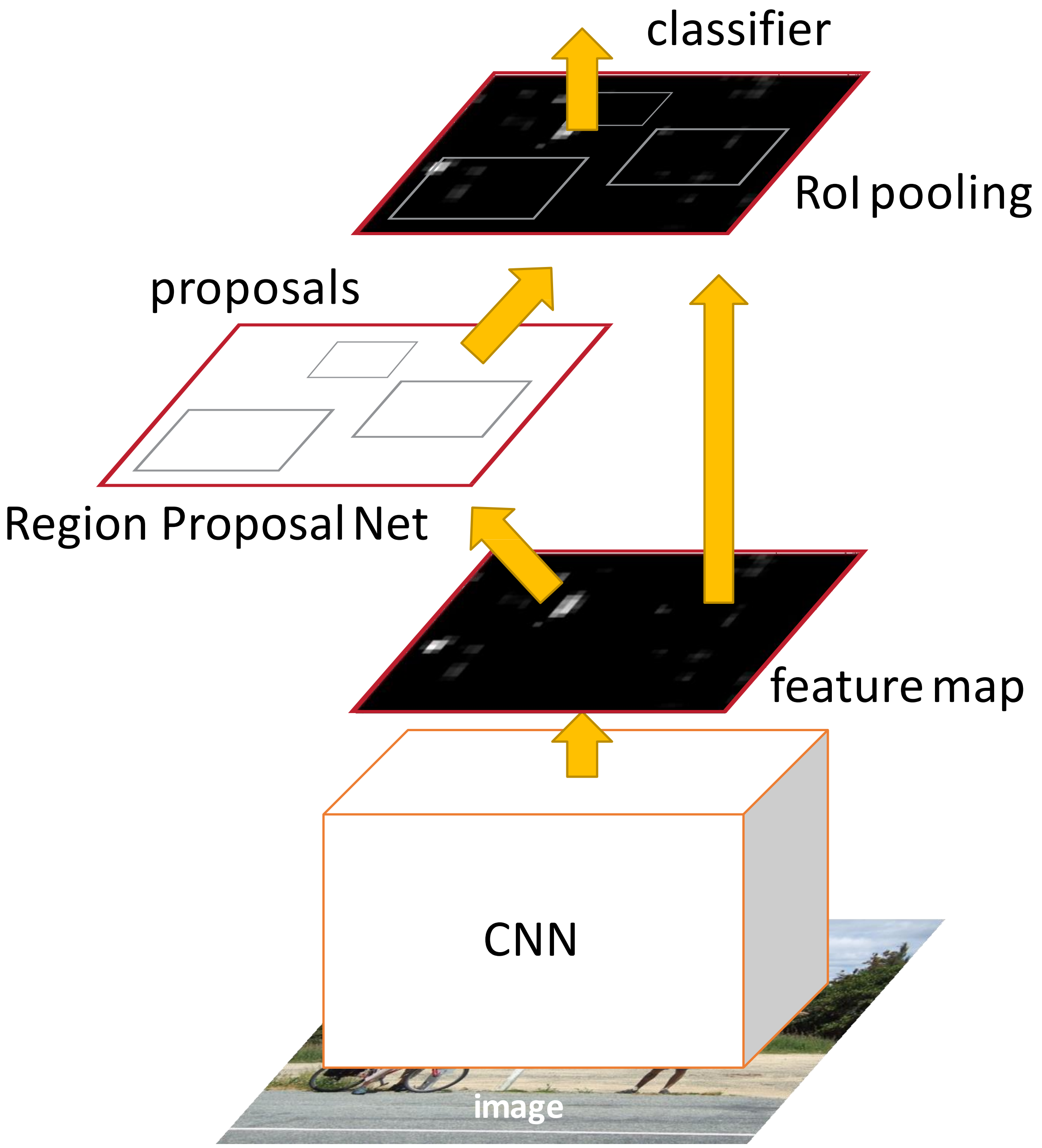
Object Detection (brief)

- Simply “Faster R-CNN + ResNet”

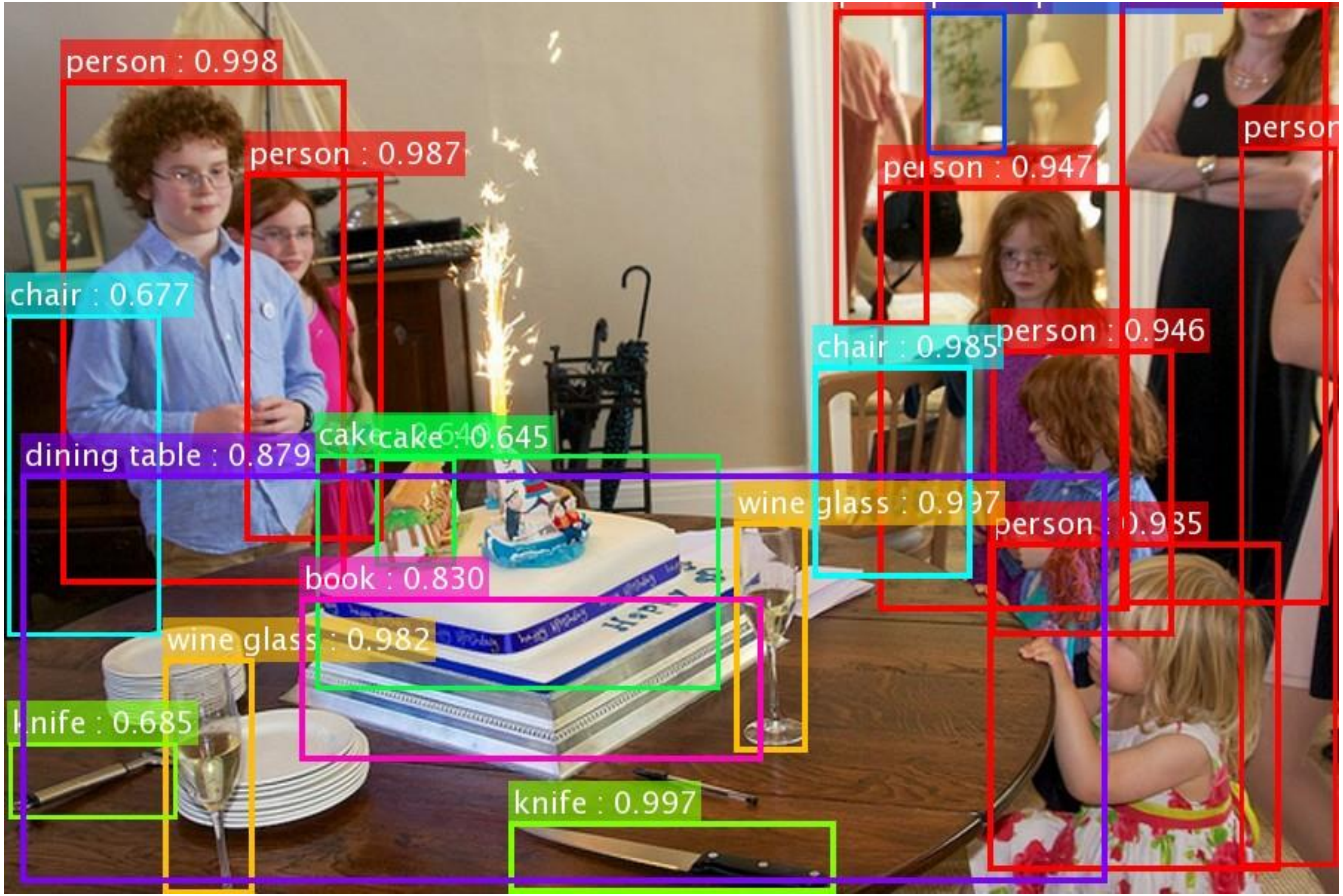
Faster R-CNN baseline	mAP@.5	mAP@.5:.95
VGG-16	41.5	21.5
ResNet-101	**48.4**	**27.2**

COCO detection results

(ResNet has 28% relative gain)



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. NIPS 2015.



Our results on MS COCO

*the original image is from the COCO dataset

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

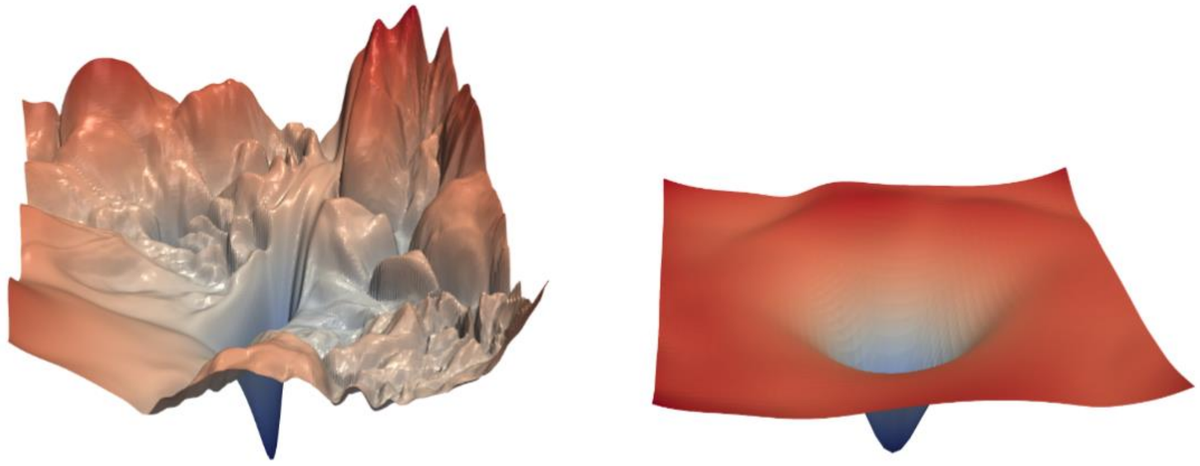
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. NIPS 2015.

Why does ResNet work so well?

- The architecture is somehow easier to optimize.
- The authors argue it probably isn't because it solves the "vanishing gradient" problem.
- While the gradients might not be "vanishing" in "plain" nets, they don't seem as stable and trustworthy, according to follow up work, e.g.

  Visualizing the Loss Landscape of Neural Nets. Hao Li, Zheng Xu , Gavin Taylor, Christoph Studer, Tom Goldstein. NeurIPS 2018.

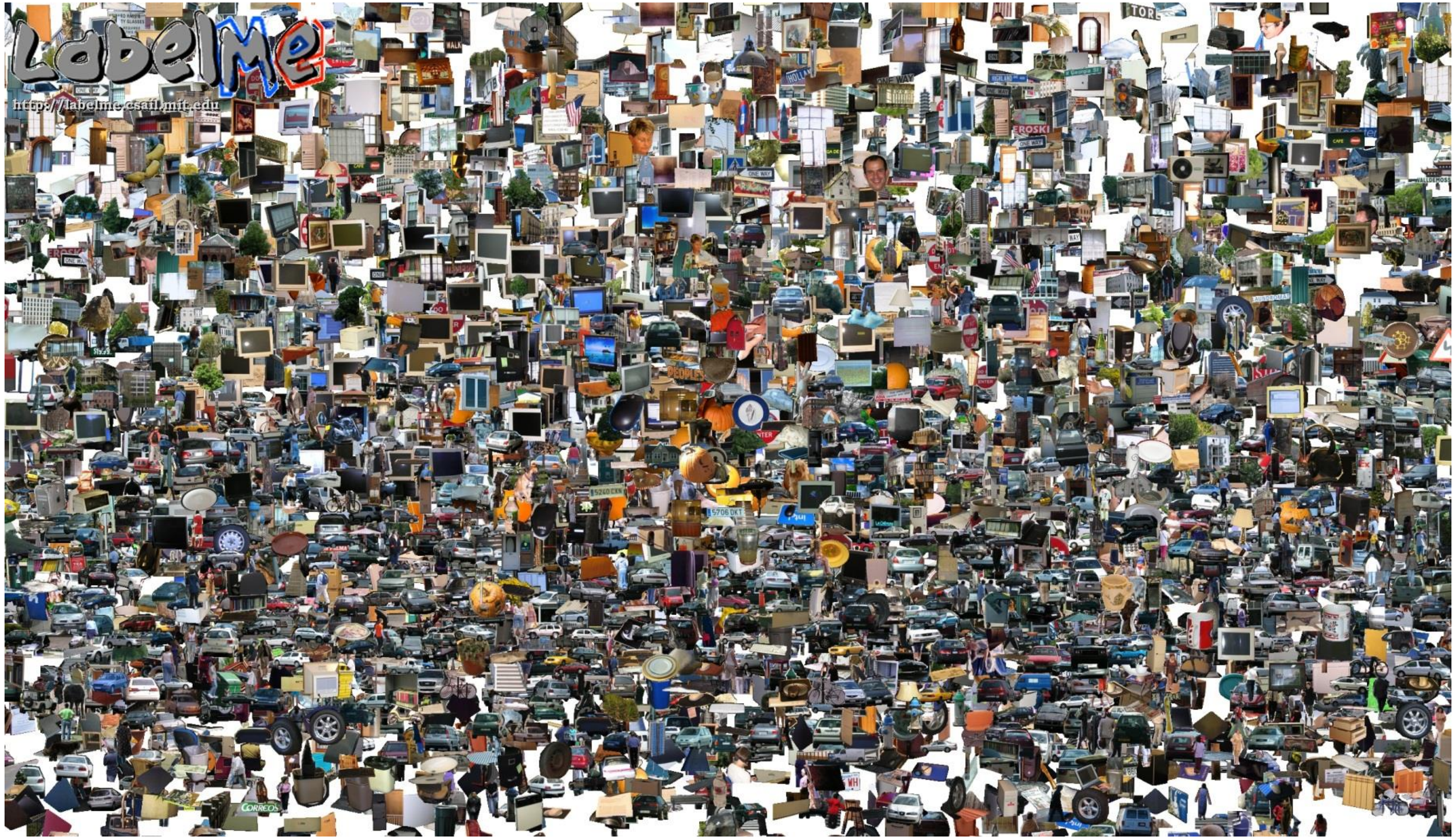
We argue that this optimization difficulty is *unlikely* to be caused by vanishing gradients. These plain networks are trained with BN [16], which ensures forward propagated signals to have non-zero variances. We also verify that the backward propagated gradients exhibit healthy norms with BN. So neither forward nor backward signals vanish. In



(a) without skip connections (b) with skip connections

Figure 1: The loss surfaces of ResNet-56 with/without skip connections. The proposed filter normalization scheme is used to enable comparisons of sharpness/flatness between the two figures.

Opportunities of Scale



Computer Vision

James Hays

Many slides from James Hays, Alyosha Efros, and Derek Hoiem

Graphic from Antonio Torralba

Outline

Opportunities of Scale: Data-driven methods

- The Unreasonable Effectiveness of Data
- Scene Completion
- Im2gps
- Recognition via Tiny Images

Computer Vision Class so far

- The geometry of image formation
  - Ancient / Renaissance
- Signal processing / Convolution
  - 1800, but really the 50's and 60's
- Hand-designed Features for recognition, either instance-level or categorical
  - 1999 (SIFT), 2003 (Video Google), 2005 (Dalal-Triggs), 2006 (spatial pyramid bag of words)
- Learning from Data
  - 1991 (EigenFaces) but late 90's to now especially

What has changed in the last 15 years?

- The Internet
- Crowdsourcing
- Learning representations from the data these sources provide (deep learning)

To be continued