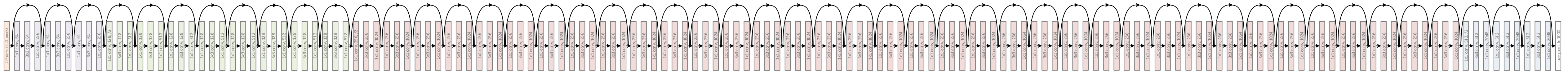
Deep Learning Gets Way Deeper

Recent Advances of Deep Learning for Computer Vision

Kaiming He

Research Scientist

Facebook AI Research (FAIR)

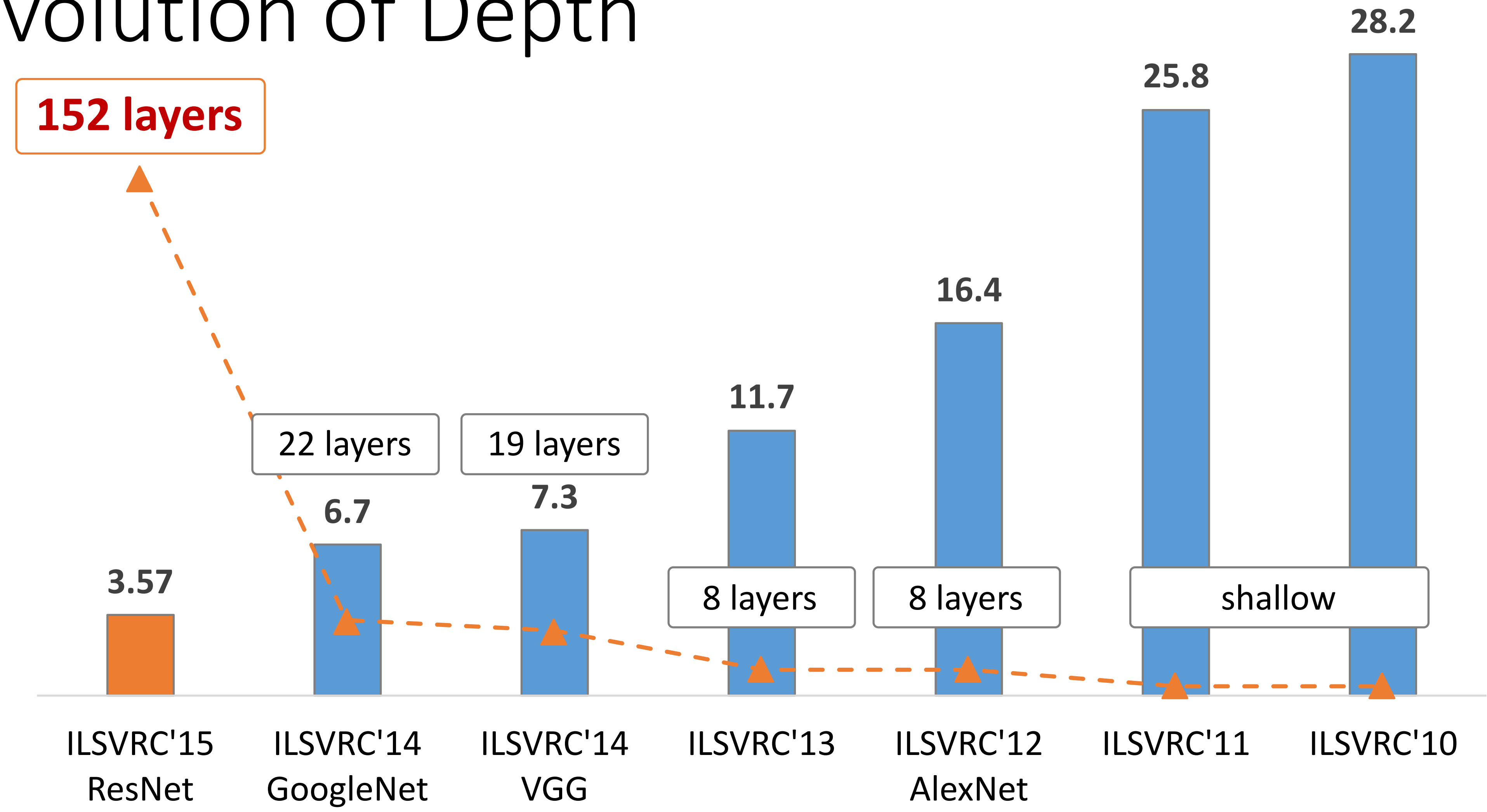


Overview

- Introduction
  - Look at some recent progress of deep learning for computer vision
- From Shallow Models to 100+ Layers
  - Advances and challenges of getting way deeper
- From Classification to Detection
  - Deep learning for complex recognition applications

Introduction

Revolution of Depth

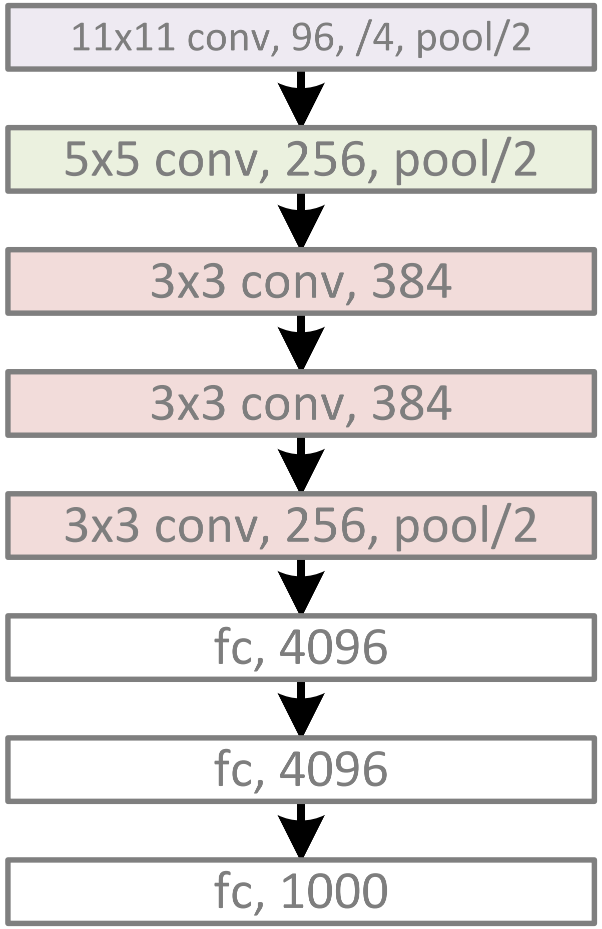


ImageNet Classification top-5 error (%)

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

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

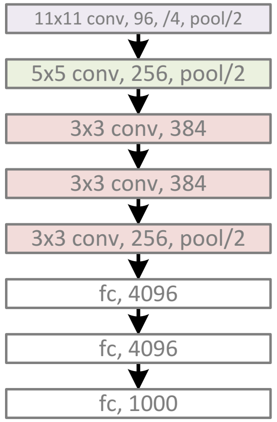


- **ReLU**
- **End-to-end (no pre-training)**
- **Data augmentation**

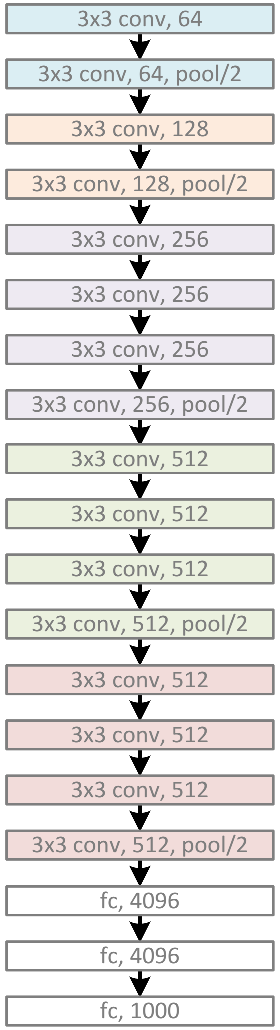
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

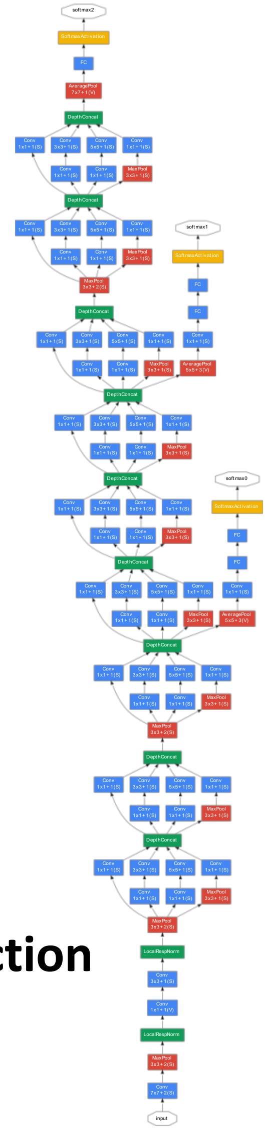


VGG, 19 layers
(ILSVRC 2014)



- **Very deep**
- **Simply deep**

GoogleNet, 22 layers
(ILSVRC 2014)

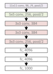


- **Branching**
- **Bottleneck**
- **Skip connection**

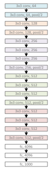
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

Revolution of Depth

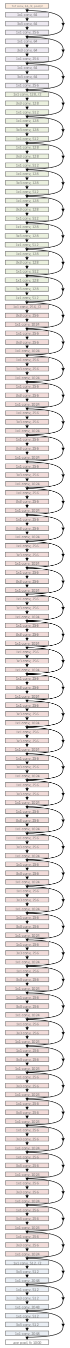
AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



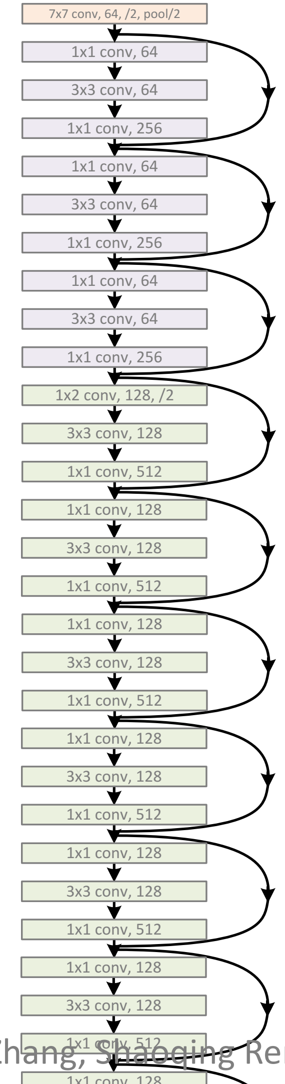
ResNet, 152 layers
(ILSVRC 2015)



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

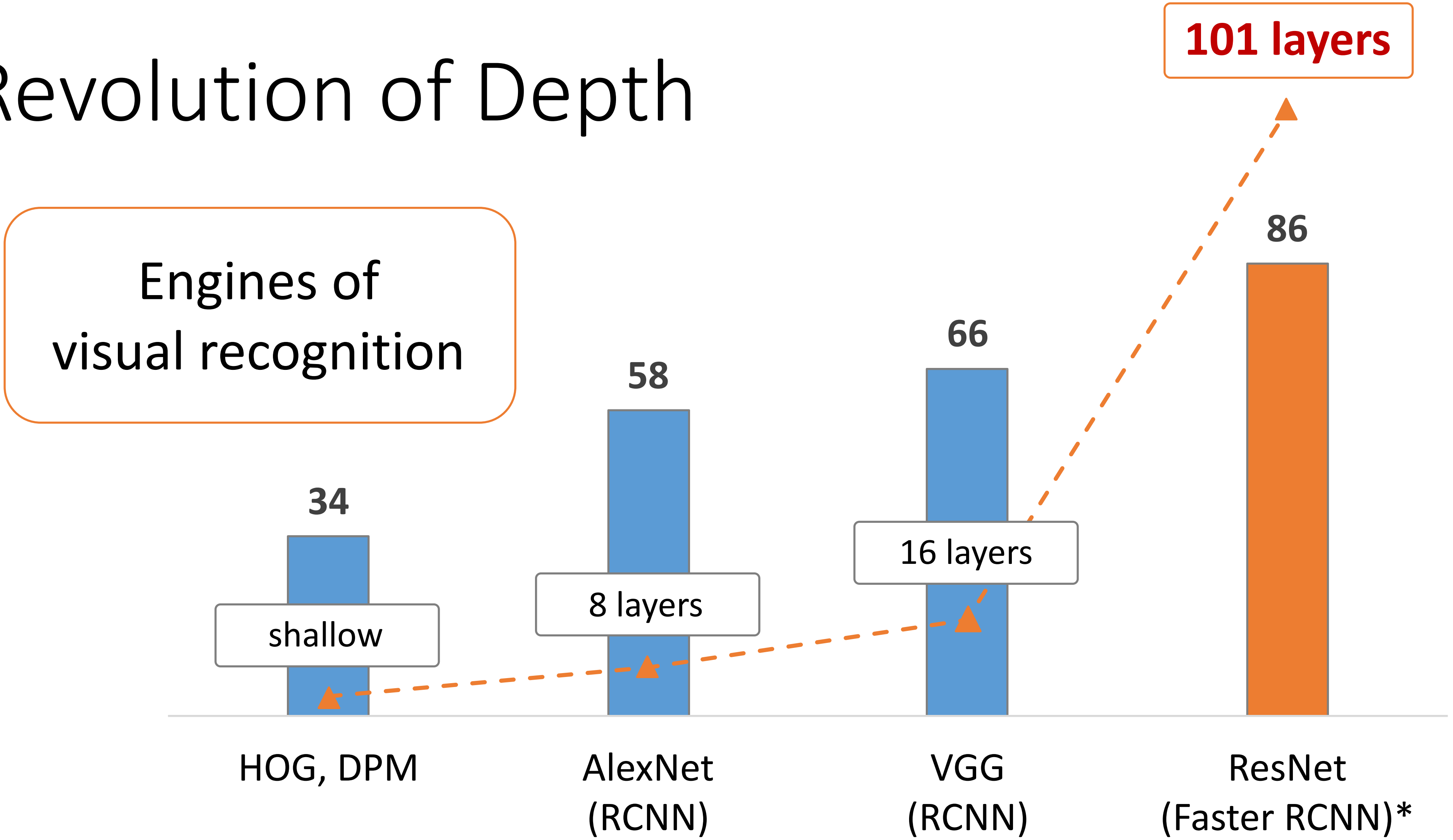
Revolution of Depth

ResNet, 152 layers



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

Revolution of Depth



PASCAL VOC 2007 **Object Detection** mAP (%)

*w/ other improvements & more data

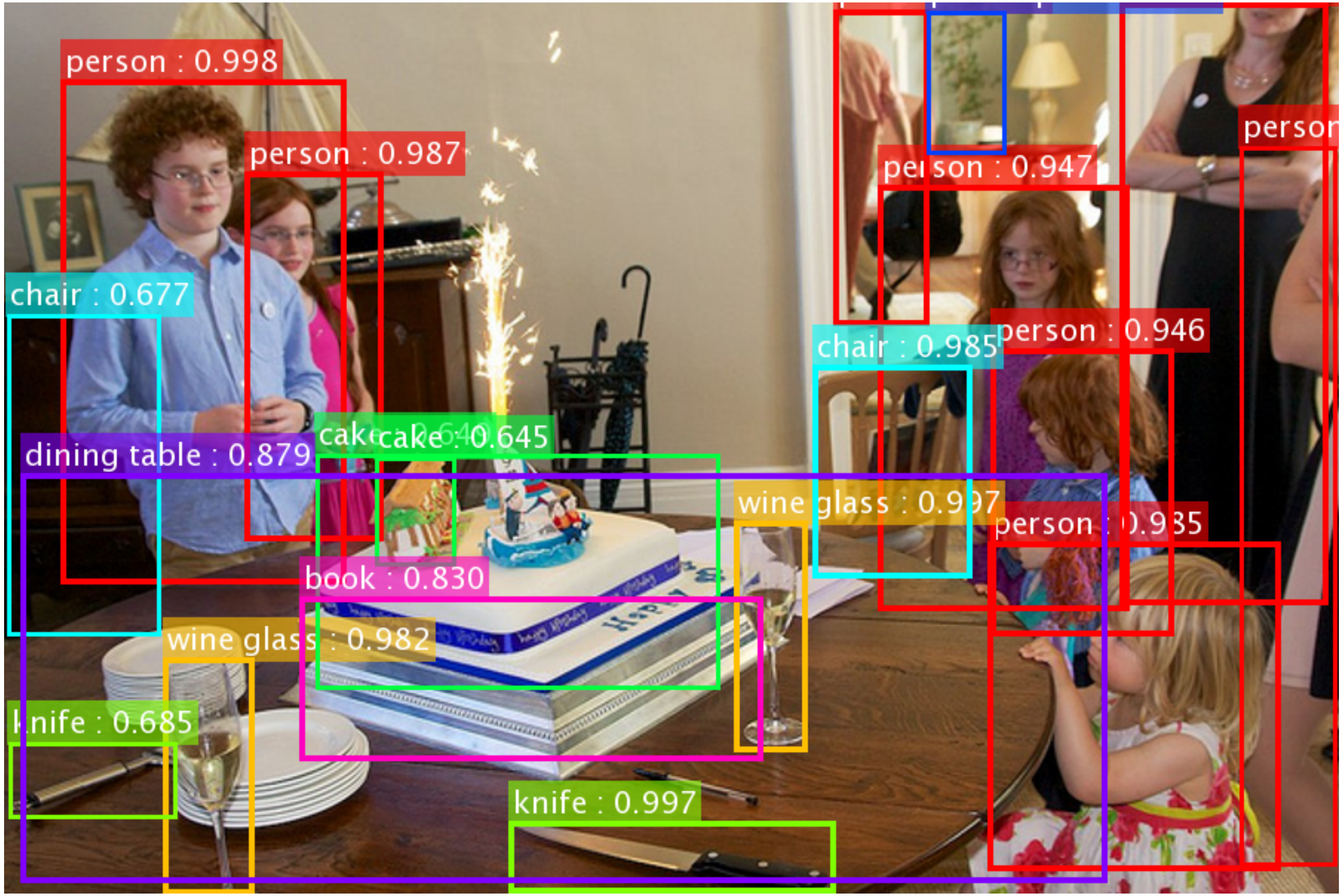
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

ResNets @ 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

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



ResNet’s object detection result on COCO

*the original image is from the COCO dataset

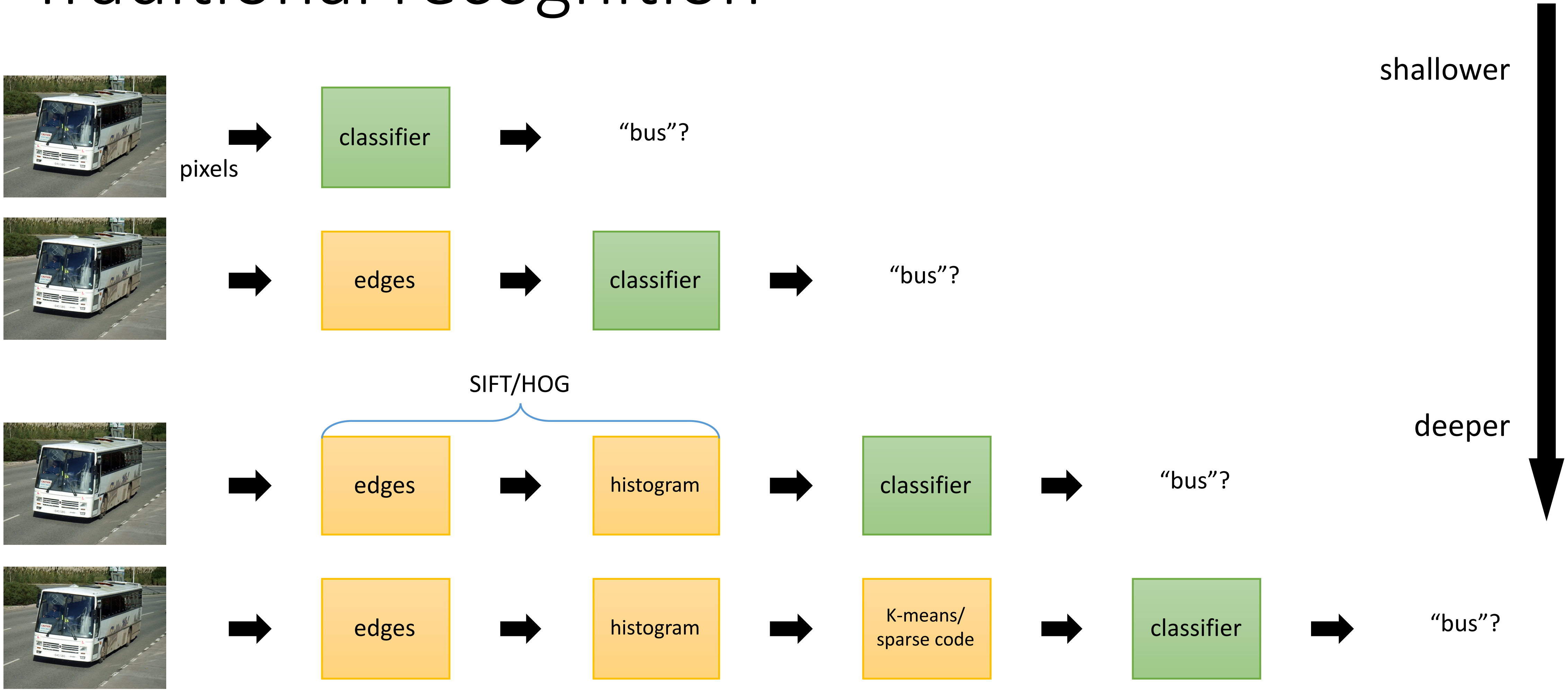
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

Background

From shallow to deep

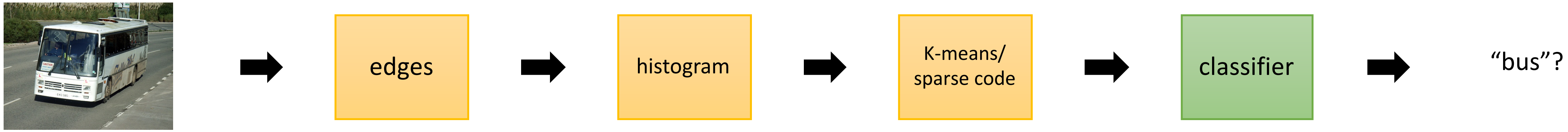
Traditional recognition

But what's next?

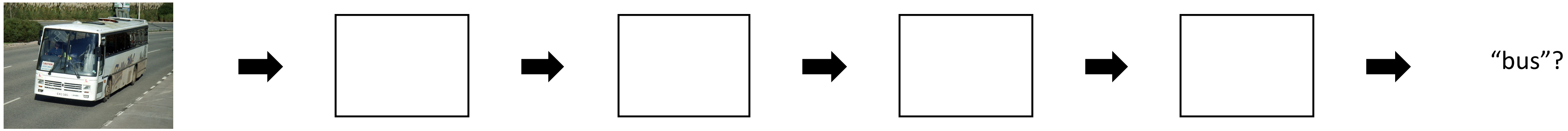


Deep Learning

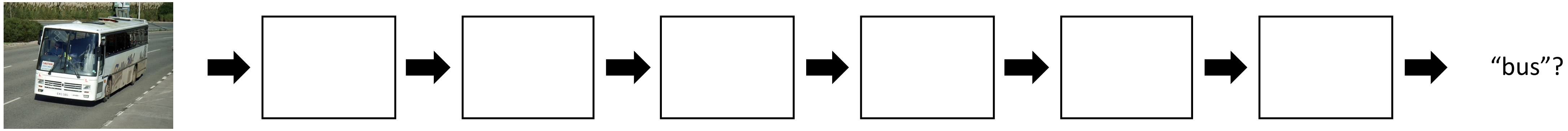
Specialized components



Generic components (“layers”)



Repeat elementary layers => Going deeper



- End-to-end learning
- Richer solution space
- **Minimal domain knowledge**

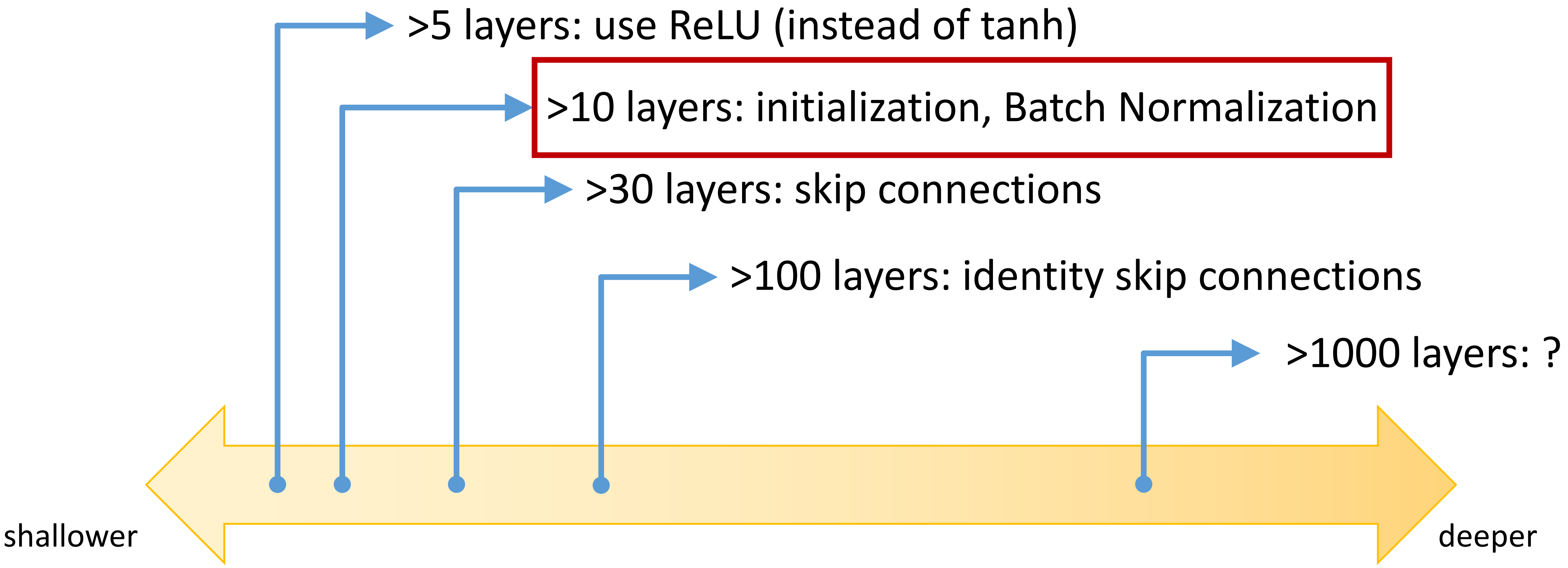
Deep Learning is “Easy”

- Minimal domain knowledge
- Data driven
- Features are generalizable

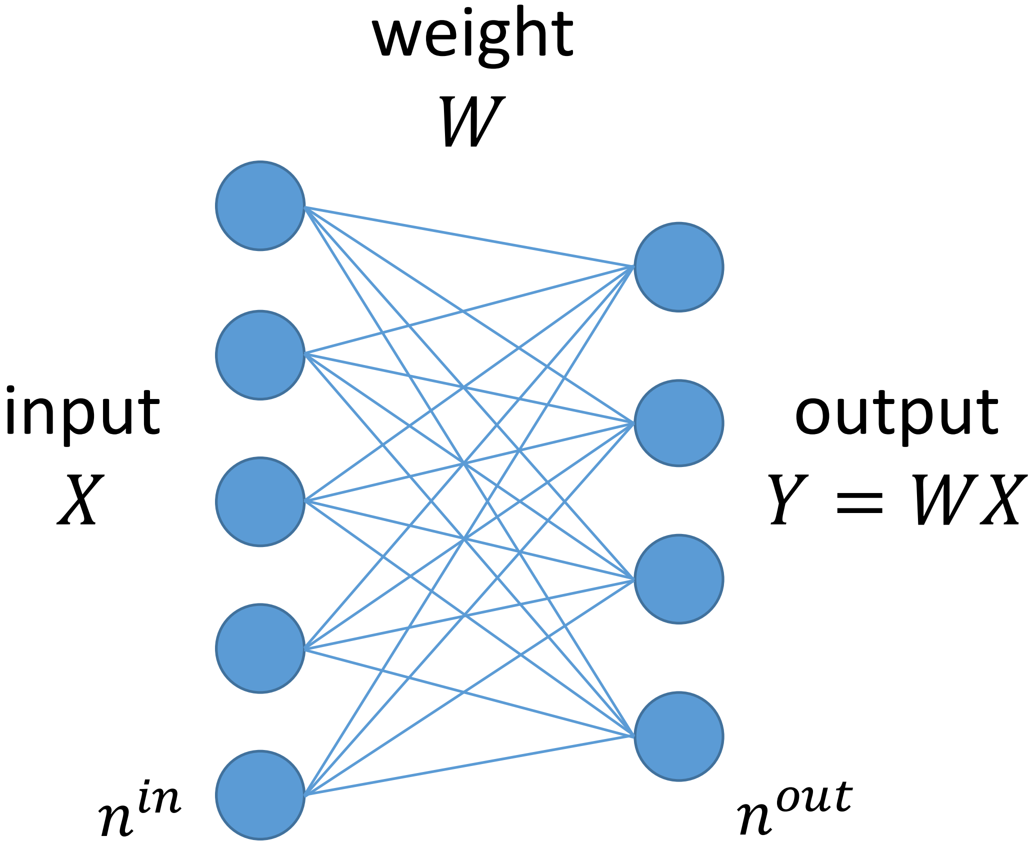
Deep Learning is “Hard”

- Black boxes?
- Unstable (vanishing/exploding)?
- Hard to tune hyper-parameters?

Cheat Sheet of Going Deeper



Initialization



If:

- Linear activation
- x, y, w: independent

Then:

1-layer:

$$Var[y] = (n^{in}Var[w])Var[x]$$

Multi-layer:

$$Var[y] = (\prod_d n_d^{in}Var[w_d])Var[x]$$

LeCun et al 1998 “Efficient Backprop”

Glorot & Bengio 2010 “Understanding the difficulty of training deep feedforward neural networks”

Initialization

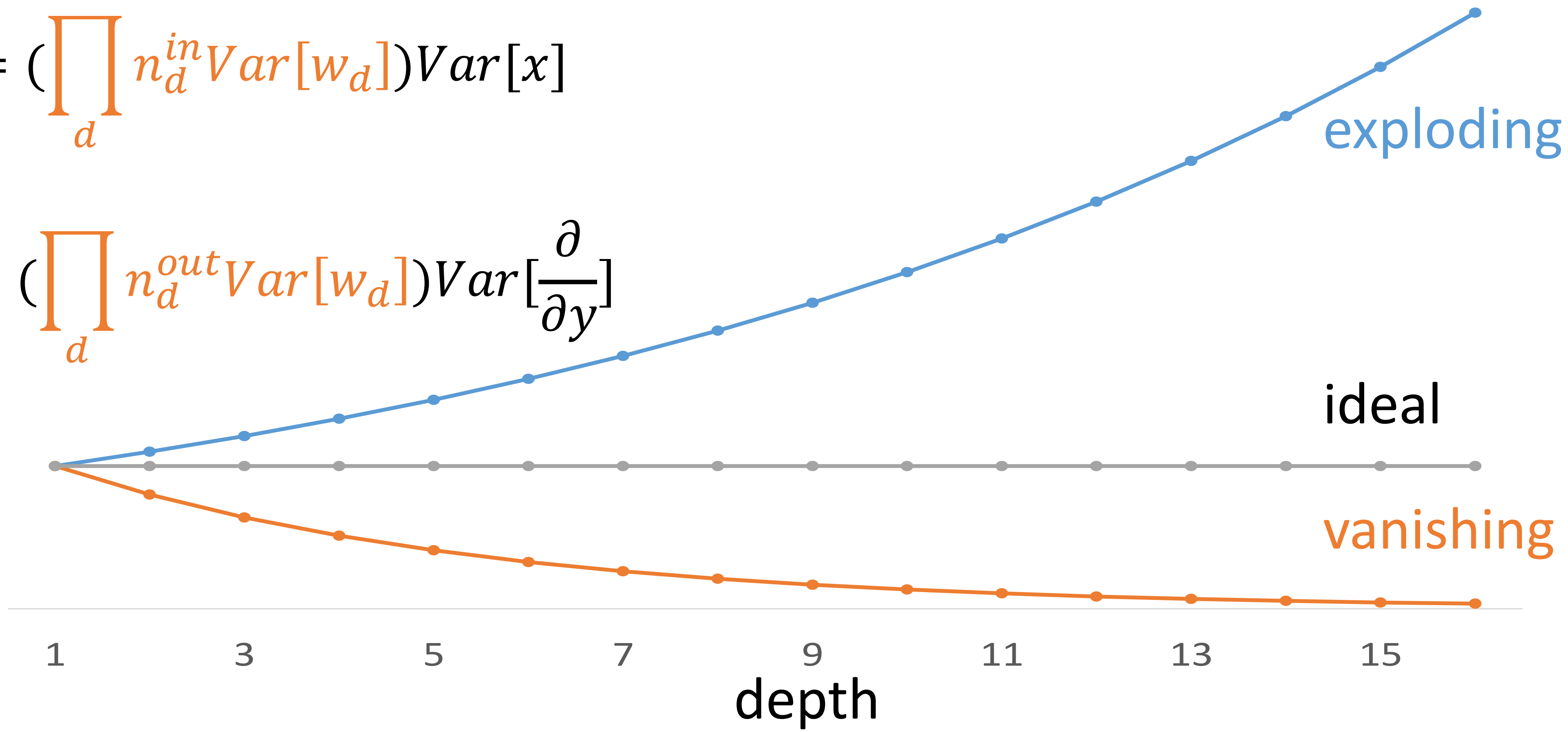
Both forward (response) and backward (gradient) signal can vanish/explode

Forward:

$$Var[y] = (\prod_d n_d^{in} Var[w_d]) Var[x]$$

Backward:

$$Var\left[\frac{\partial}{\partial x}\right] = (\prod_d n_d^{out} Var[w_d]) Var[\frac{\partial}{\partial y}]$$



LeCun et al 1998 “Efficient Backprop”

Glorot & Bengio 2010 “Understanding the difficulty of training deep feedforward neural networks”

Initialization

- Initialization under **linear** assumption

$$\prod_d n_d^{in} Var[w_d] = const_{\mathrm{fw}} \text{ (healthy forward)}$$

and

$$\prod_d n_d^{out} Var[w_d] = const_{\mathrm{bw}} \text{(healthy backward)}$$

➡ $$\boxed{\begin{array}{c} n_d^{in} Var[w_d] = 1 \\ \text{or*} \\ n_d^{out} Var[w_d] = 1 \end{array}}$$

"*Xavier*" init in Caffe

*: $n_d^{out} = n_{d+1}^{in}$, so $\frac{const_{\mathrm{bw}}}{const_{\mathrm{fw}}} = \frac{n_{\mathrm{last}}^{out}}{n_{\mathrm{first}}^{in}} < \infty$.
It is sufficient to use either form.

LeCun et al 1998 "Efficient Backprop"
Glorot & Bengio 2010 "Understanding the difficulty of training deep feedforward neural networks"

Initialization

- Initialization under **ReLU**

$$\prod_d \frac{1}{2} n_d^{in} Var[w_d] = const_{\mathrm{fw}} \text{ (healthy forward)}$$

and

$$\prod_d \frac{1}{2} n_d^{out} Var[w_d] = const_{\mathrm{bw}} \text{(healthy backward)}$$

$$\Rightarrow \boxed{\begin{array}{c} \frac{1}{2} n_d^{in} Var[w_d] = 1 \\ \text{or} \\ \frac{1}{2} n_d^{out} Var[w_d] = 1 \end{array}}$$

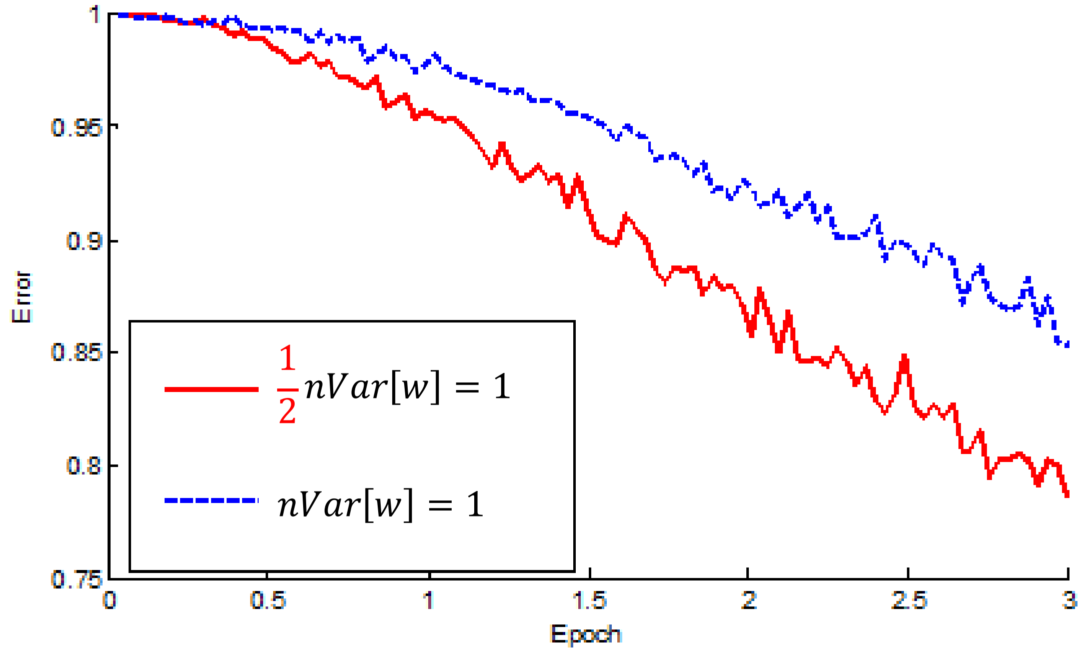
"*MSRA*" init in Caffe

With D layers, a factor of 2 per layer has exponential impact of 2^D

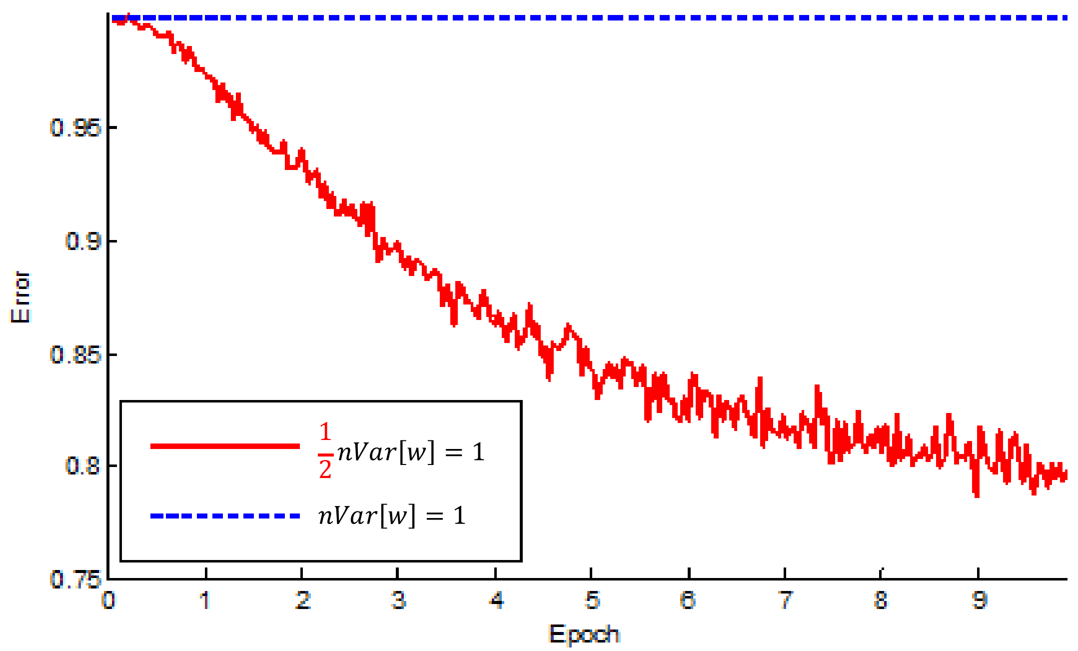
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". ICCV 2015.

Initialization

22-layer ReLU net:
good init converges faster



30-layer ReLU net:
good init is able to converge



*Figures show the beginning of training

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". ICCV 2015.

Batch Normalization (BN)

- Normalizing input (LeCun et al 1998 "Efficient Backprop")
- BN: normalizing each layer, for each mini-batch
- Greatly accelerate training
- Less sensitive to initialization
- Improve regularization

S. Ioffe & C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML 2015

Batch Normalization (BN)

$$\rightarrow \boxed{\text{layer}} \rightarrow x \rightarrow \quad \hat{x} = \frac{x-\mu}{\sigma} \quad \rightarrow \quad y = \gamma\hat{x} + \beta$$

- 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

S. Ioffe & C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML 2015

Batch Normalization (BN)

$$\rightarrow \boxed{\text{layer}} \rightarrow x \rightarrow \quad \hat{x} = \frac{x-\mu}{\sigma} \quad \rightarrow \quad y = \gamma\hat{x} + \beta$$

2 modes of BN:

- Train mode:
  - μ, σ are functions of x; backprop gradients
- Test mode:
  - μ, σ are pre-computed* on training set

Caution: make sure your BN usage is correct

*: by running average, or post-processing after training

S. Ioffe & C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML 2015

Batch Normalization (BN)

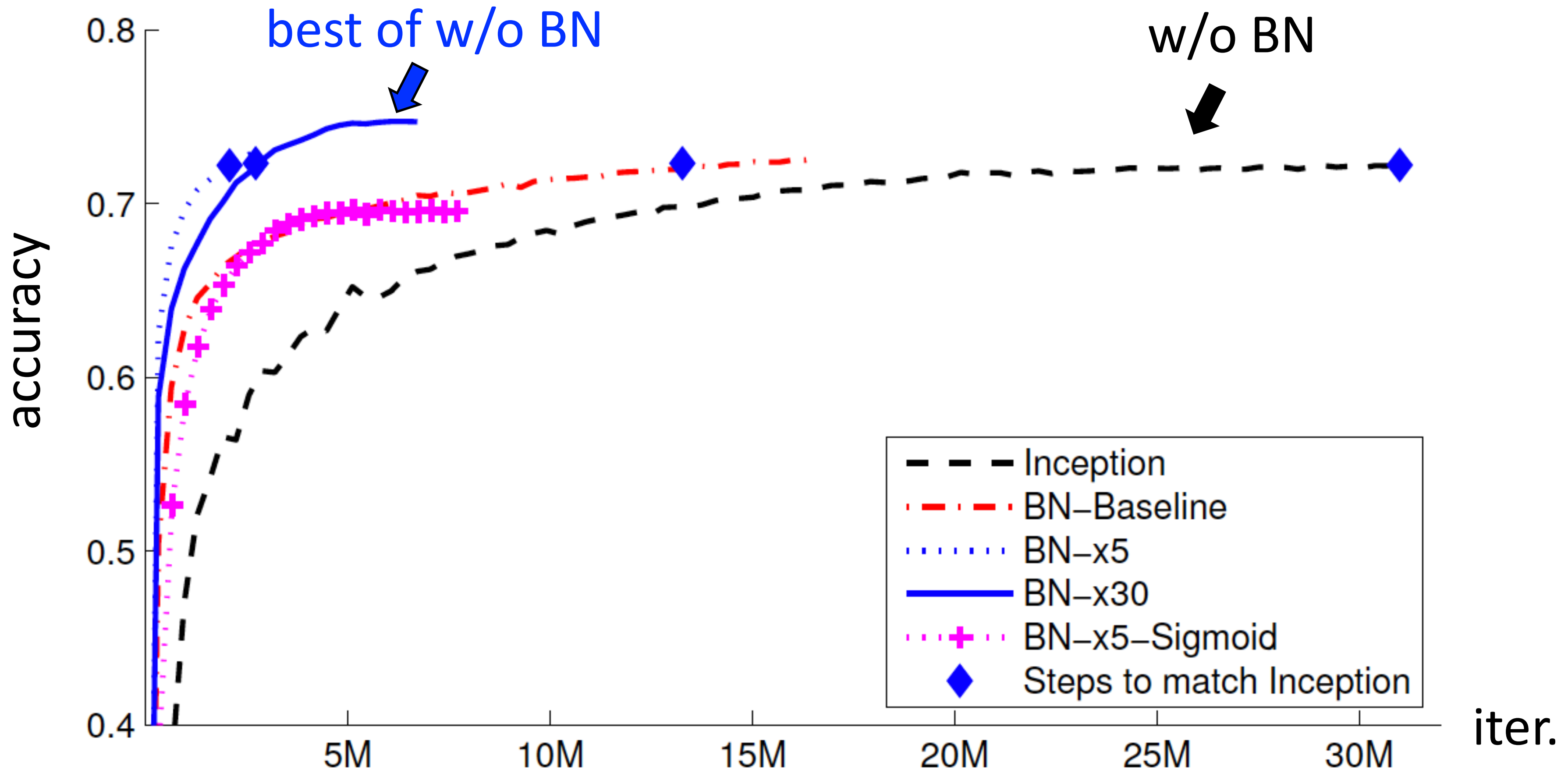


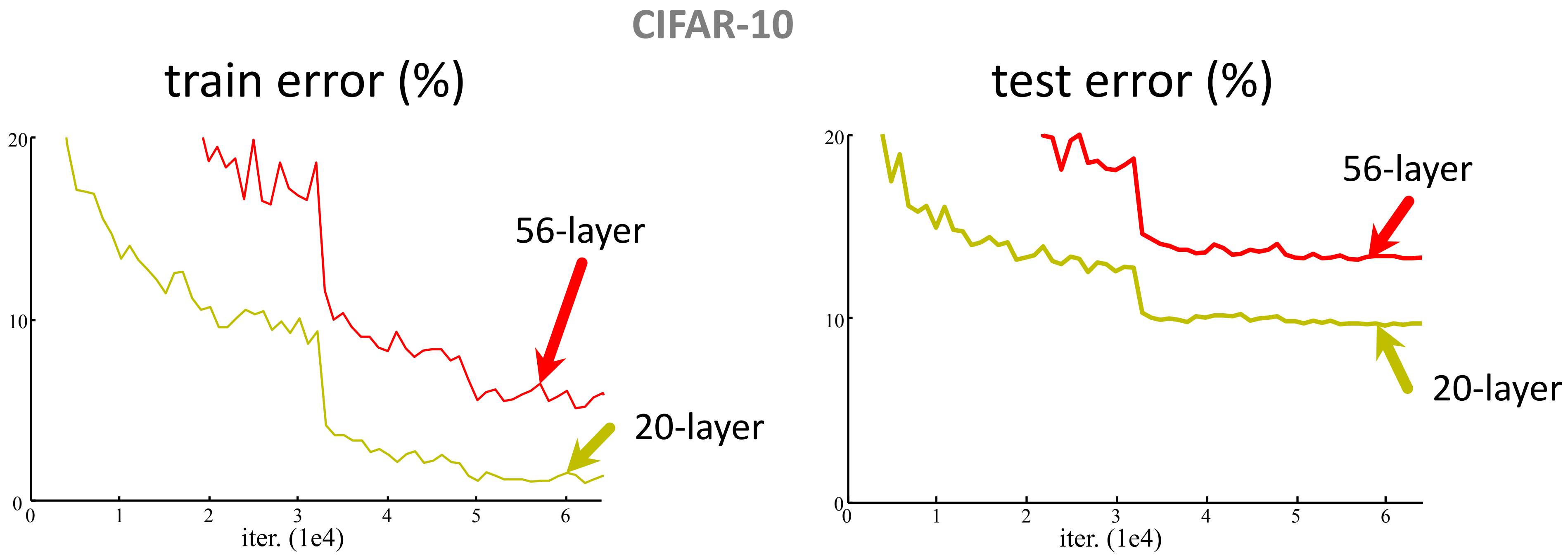
Figure taken from [S. Ioffe & C. Szegedy]

S. Ioffe & C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML 2015

Deep Residual Networks

From 10 layers to 100+ layers

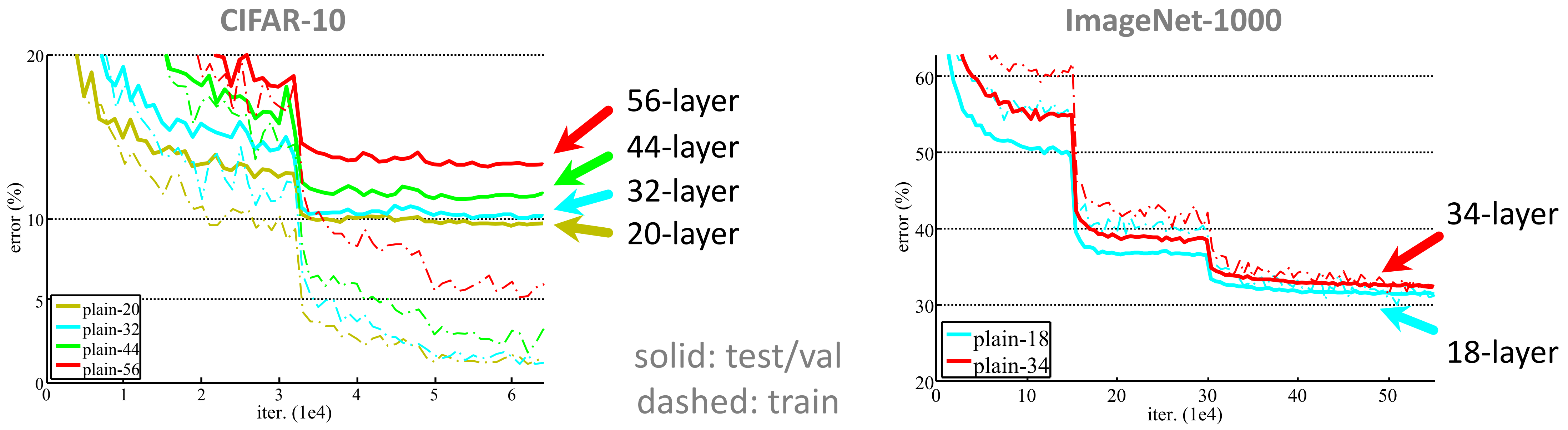
Simply stacking layers?



- *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.

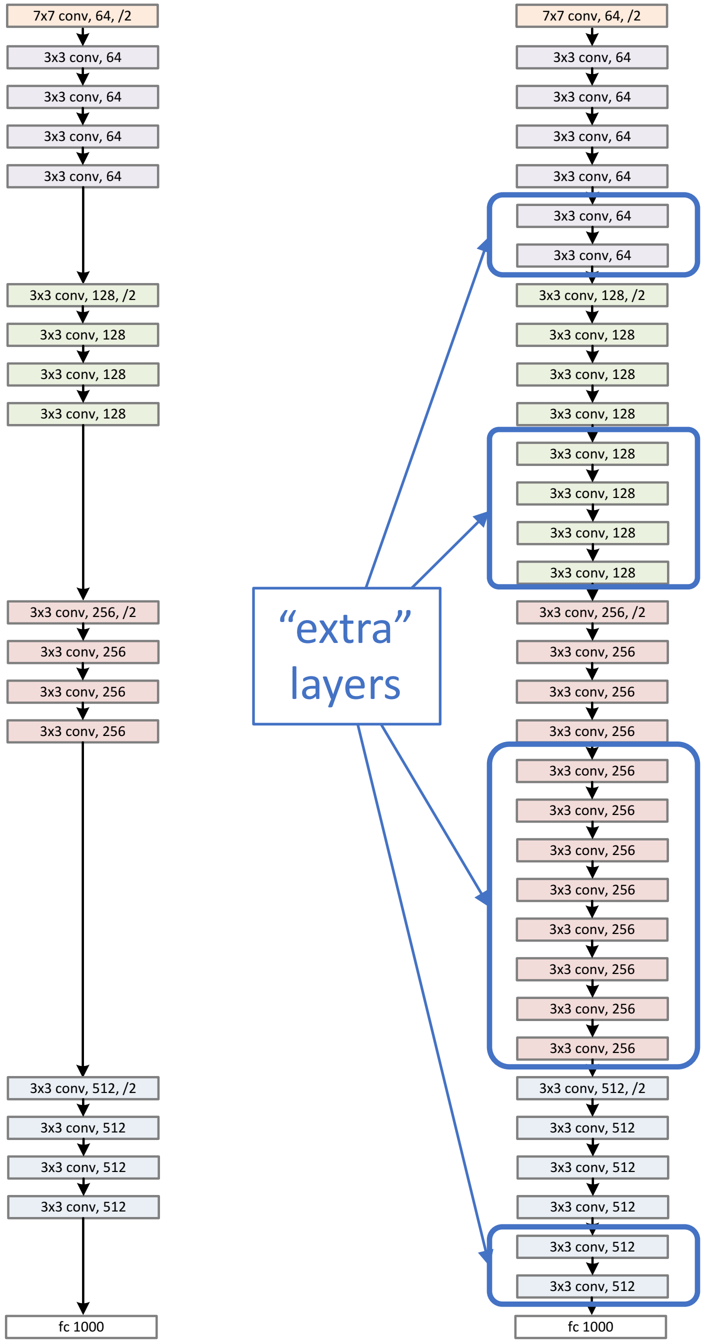
Simply stacking layers?



- “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)



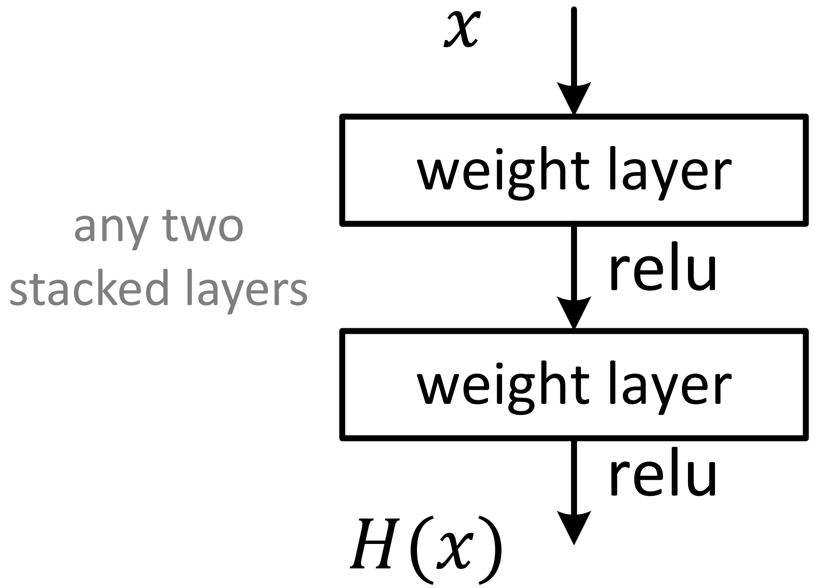
a deeper counterpart (34 layers)

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

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

Deep Residual Learning

- Plaint net



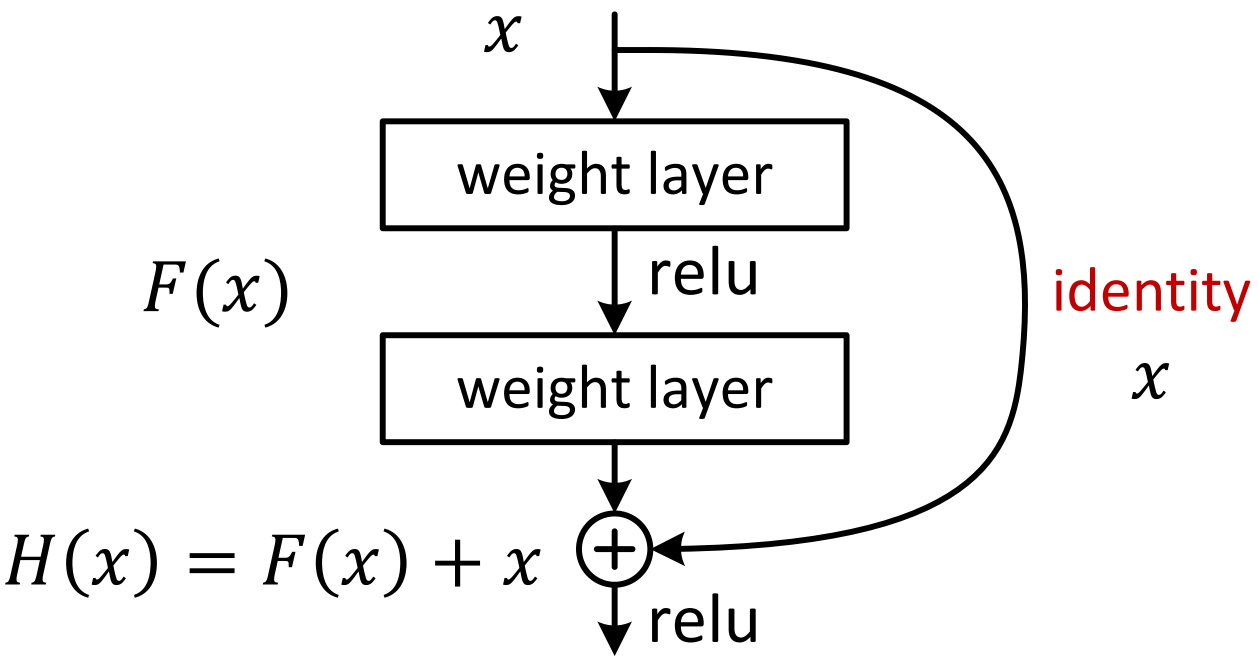
$H(x)$ is any desired mapping,

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

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

Deep Residual Learning

- Residual net



$H(x)$ is any desired mapping,

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

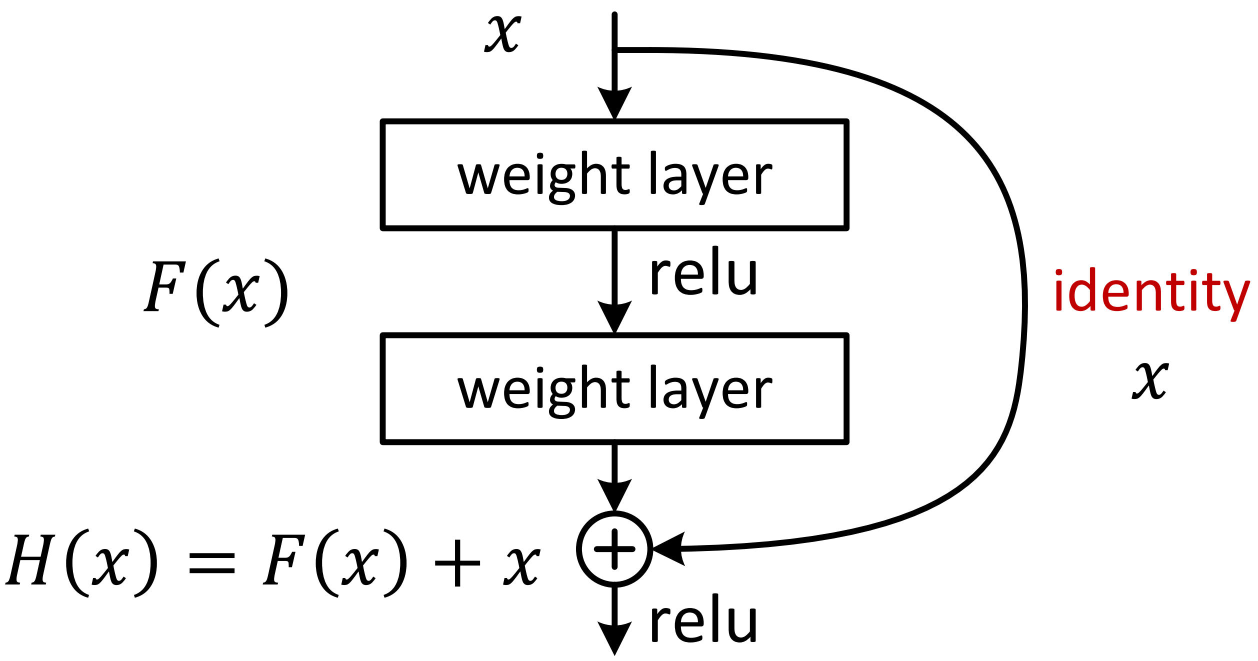
hope the 2 weight layers fit $F(x)$

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

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

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

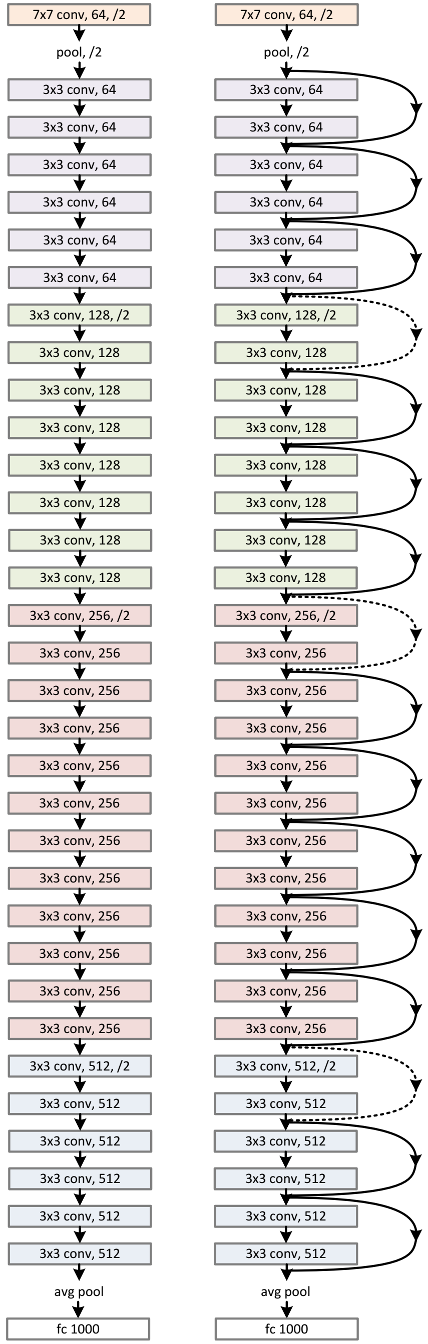
Related Works – Residual Representations

- VLAD & Fisher Vector [Jegou et al 2010], [Perronnin et al 2007]
  - Encoding residual vectors; powerful shallower representations.
- Product Quantization (IVF-ADC) [Jegou et al 2011]
  - Quantizing residual vectors; efficient nearest-neighbor search.
- MultiGrid & Hierarchical Precondition [Briggs, et al 2000], [Szeliski 1990, 2006]
  - Solving residual sub-problems; efficient PDE solvers.

Network “Design”

- Keep it simple
- Our basic design (VGG-style)
  - all 3x3 conv (almost)
  - spatial size /2 => # filters x2 (~same complexity per layer)
  - Simple design; just deep!
- Other remarks:
  - no hidden fc
  - no dropout

plain net



ResNet

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

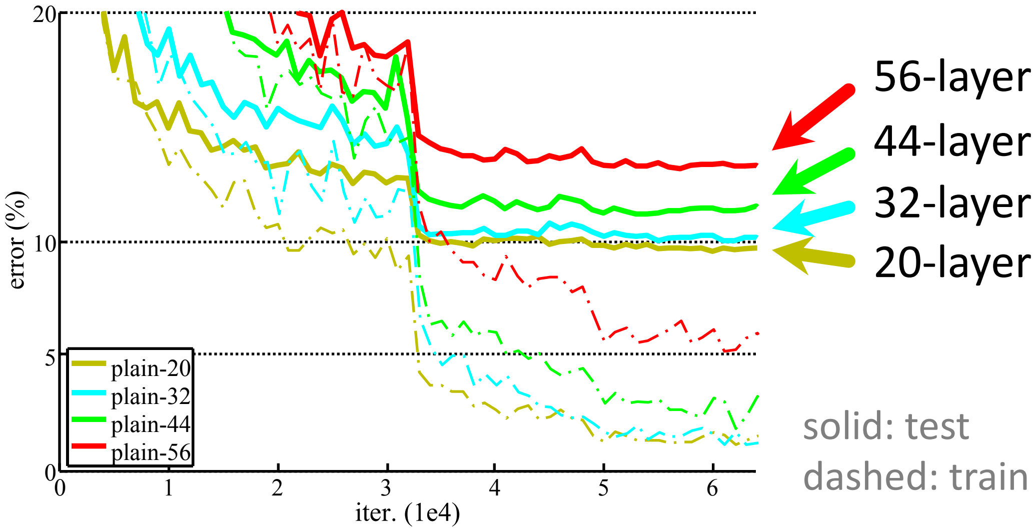
Training

- All plain/residual nets are trained from scratch
- All plain/residual nets use Batch Normalization
- Standard hyper-parameters & augmentation

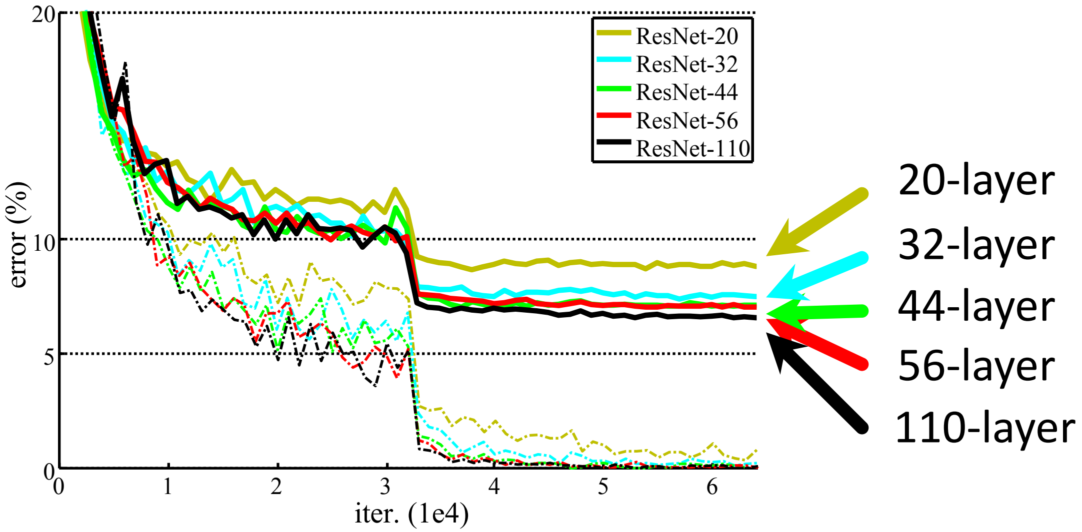
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

CIFAR-10 experiments

CIFAR-10 plain nets



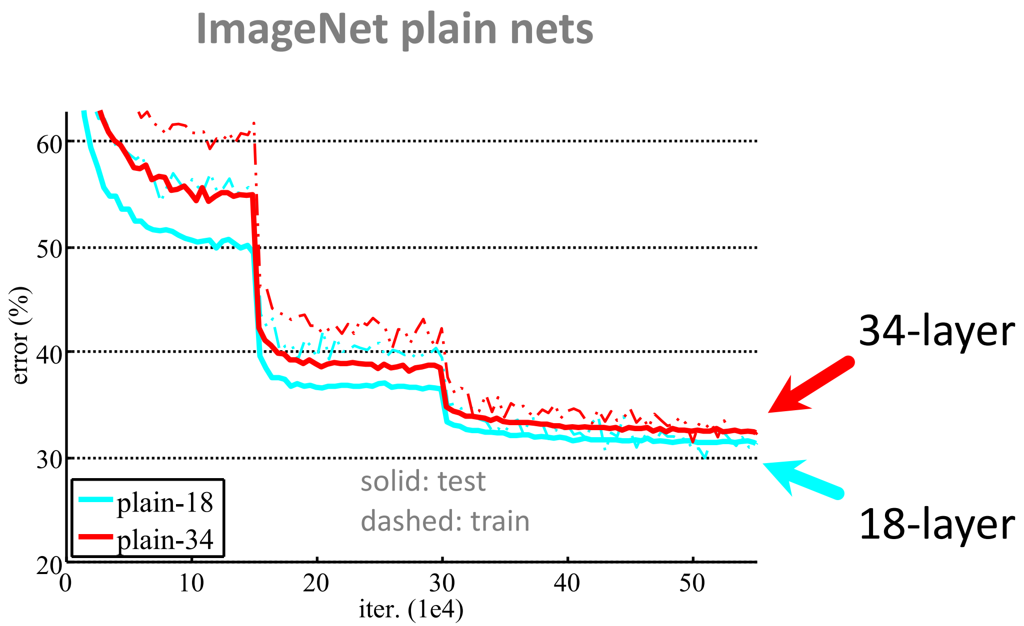
CIFAR-10 ResNets

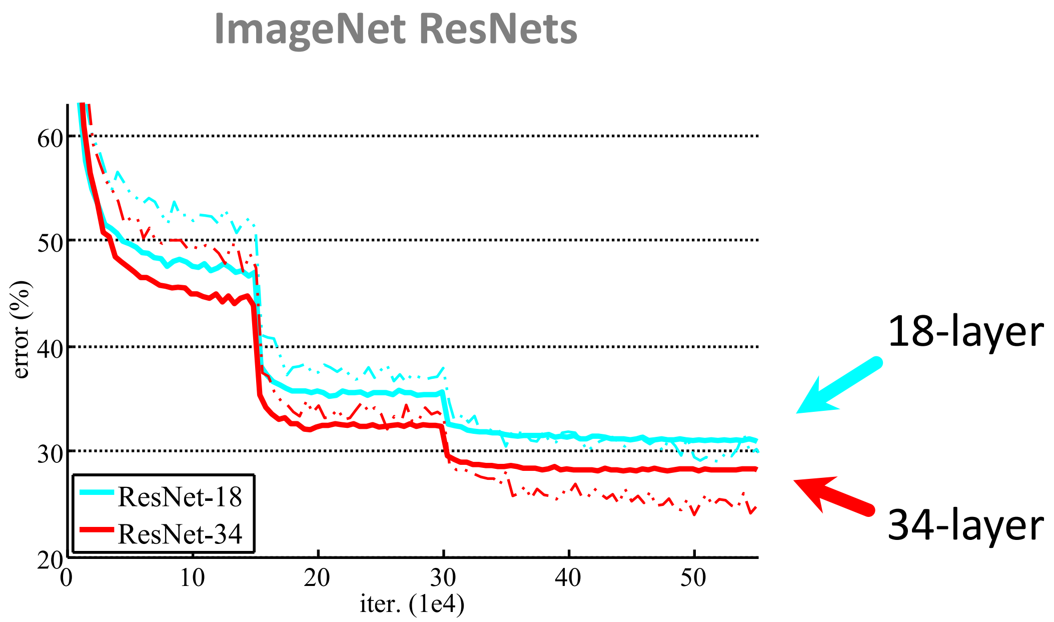


- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

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

ImageNet experiments



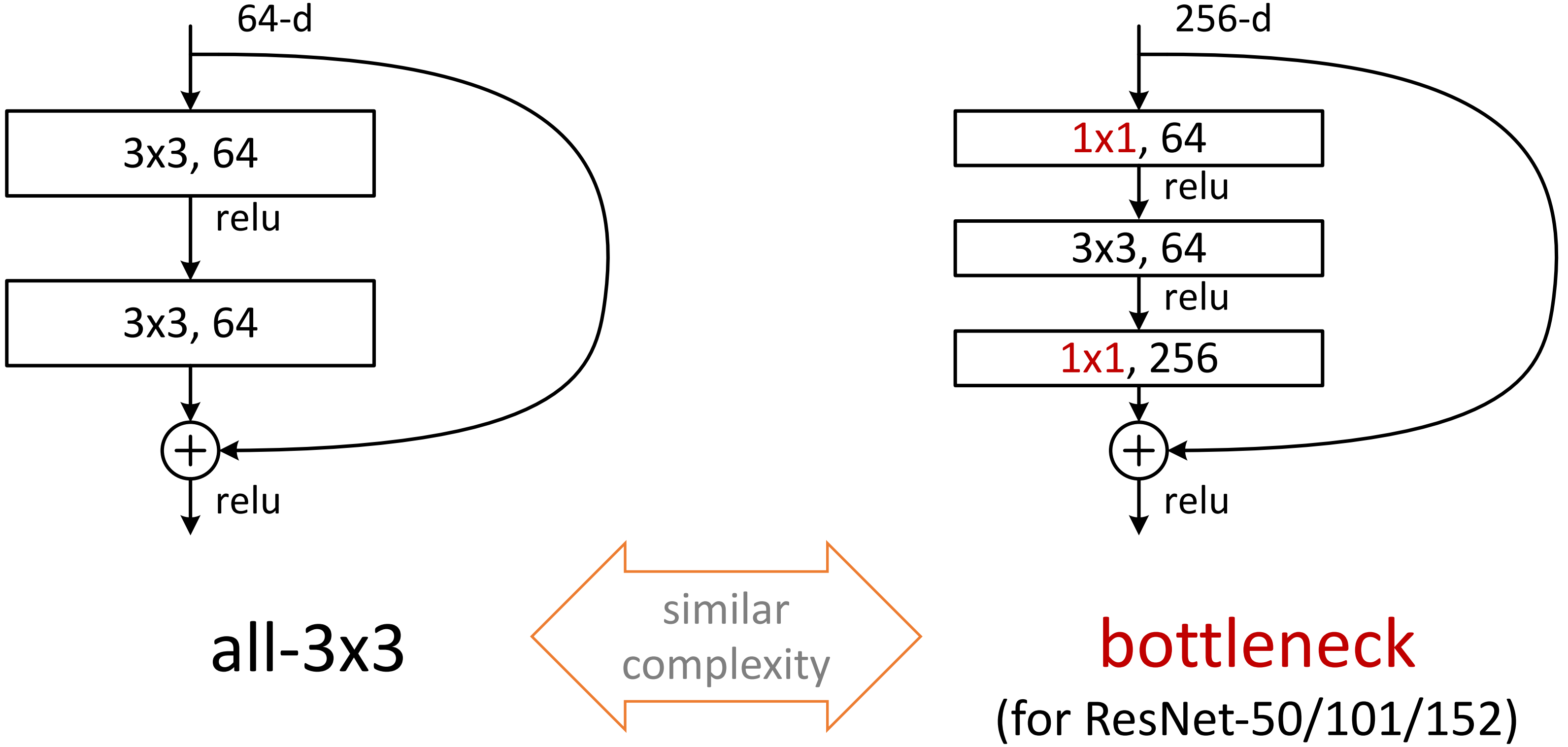


- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

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

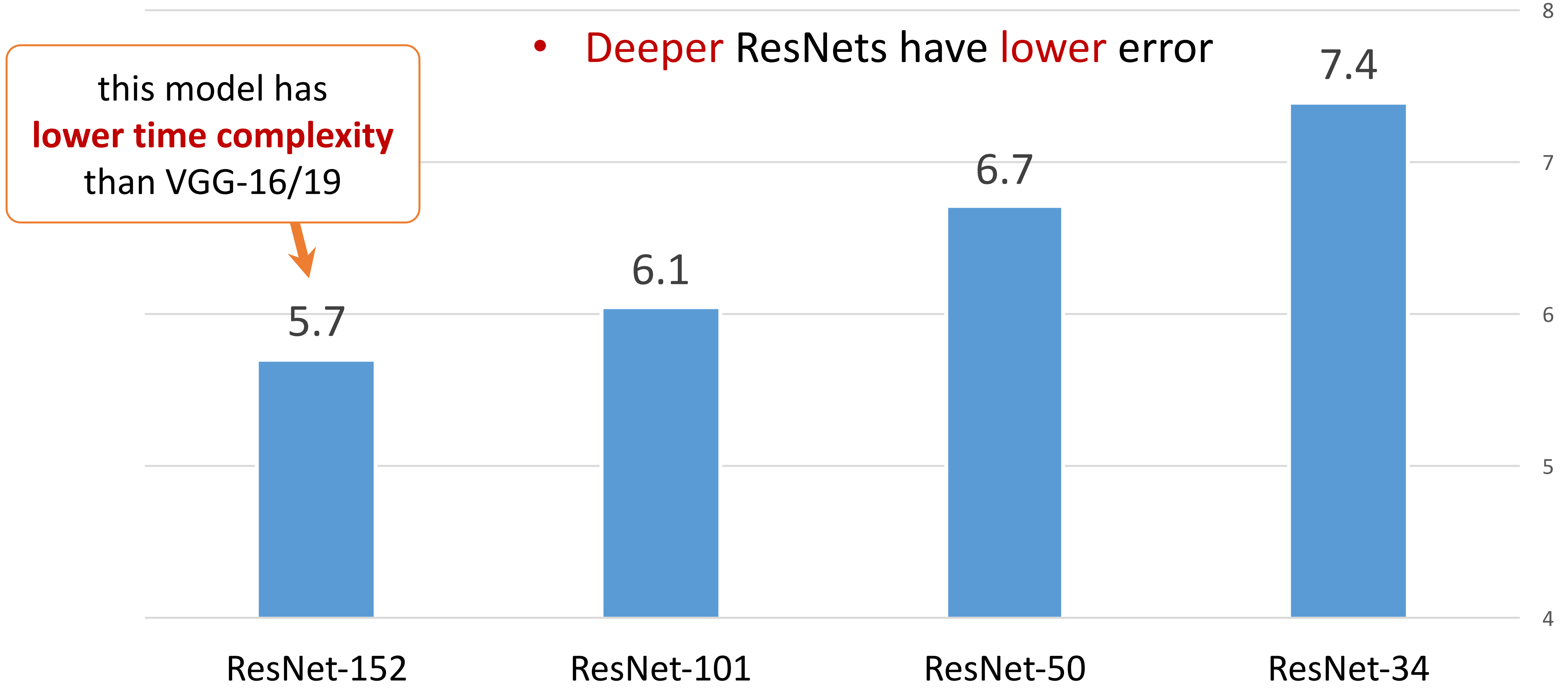
ImageNet experiments

- A practical design of going deeper



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

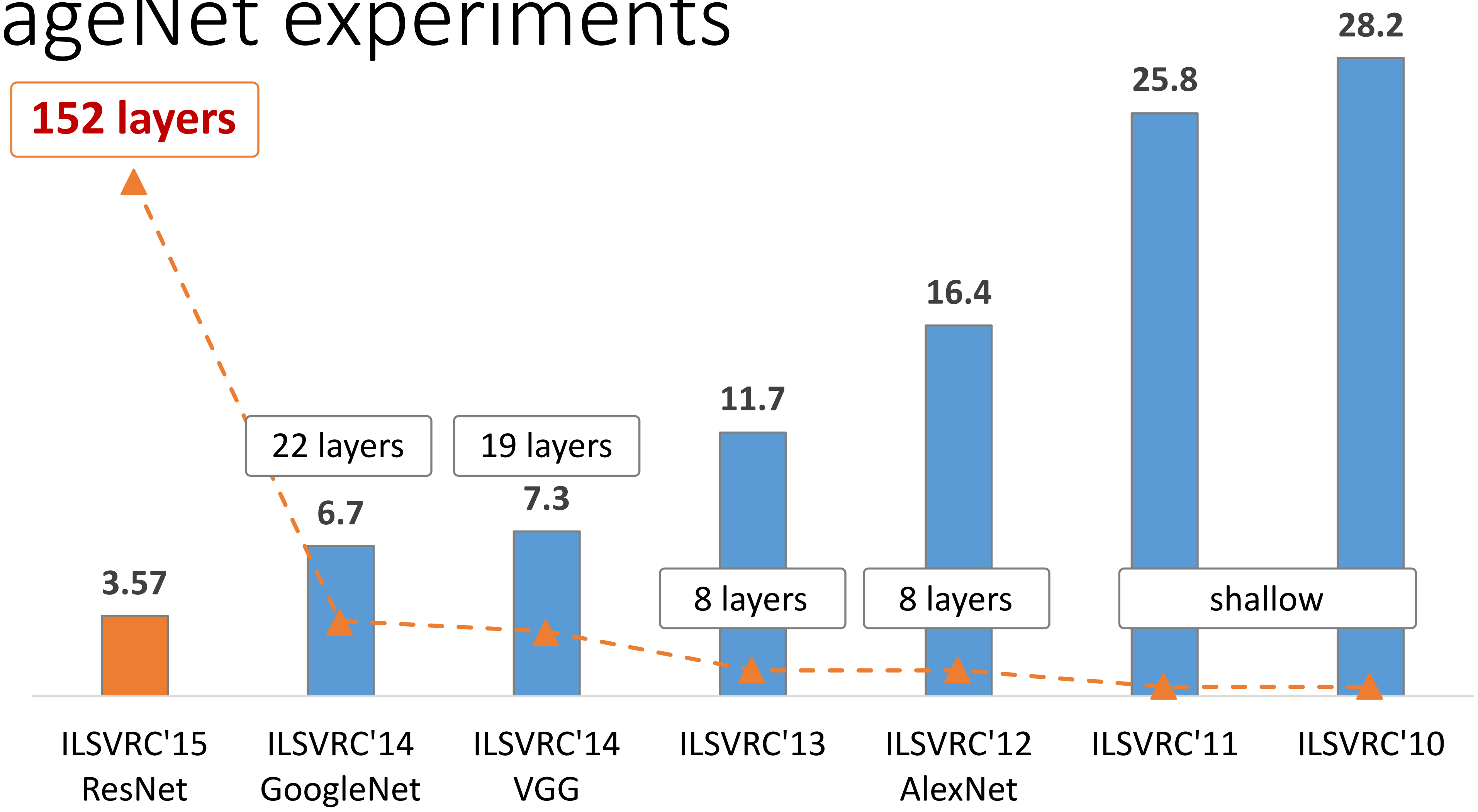
ImageNet experiments



10-crop testing, top-5 val error (%)

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

ImageNet experiments



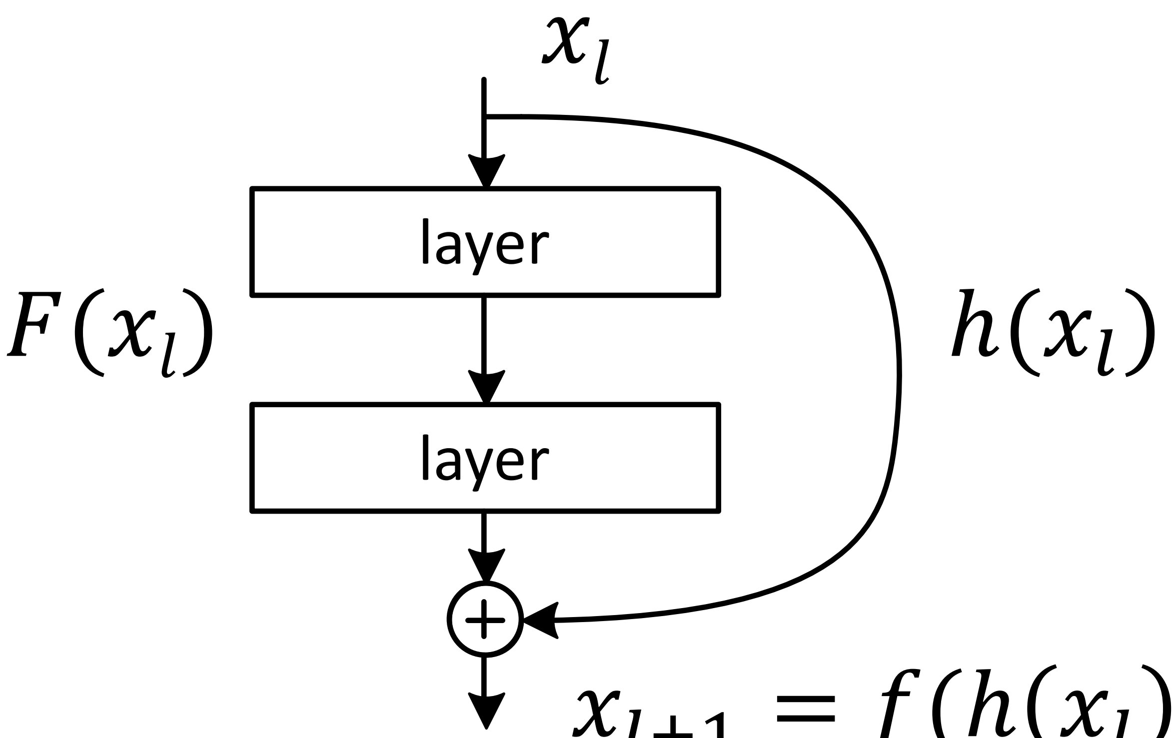
ImageNet Classification top-5 error (%)

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

On the Importance of Identity Mapping

A Deeper Look at ResNets

On identity mappings for **optimization**

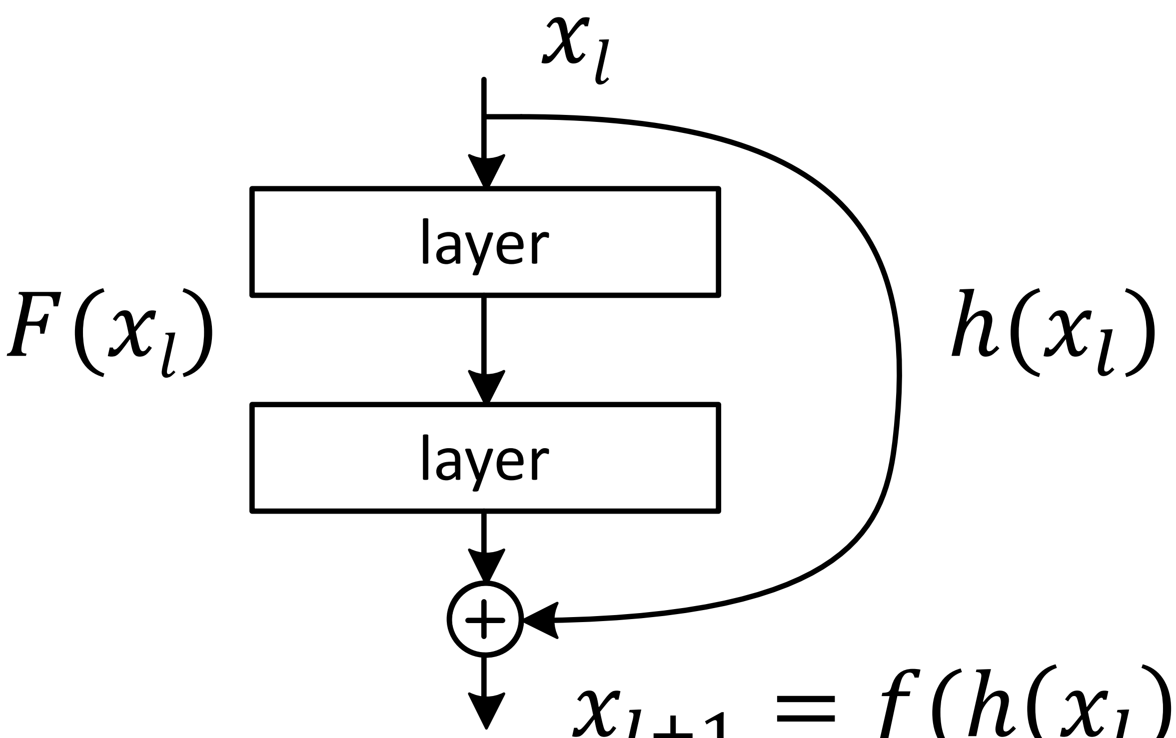


$$x_{l+1} = f(h(x_l) + F(x_l))$$

- shortcut mapping: h = identity
- after-add mapping: f = ReLU

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". ECCV 2016.

On identity mappings for **optimization**

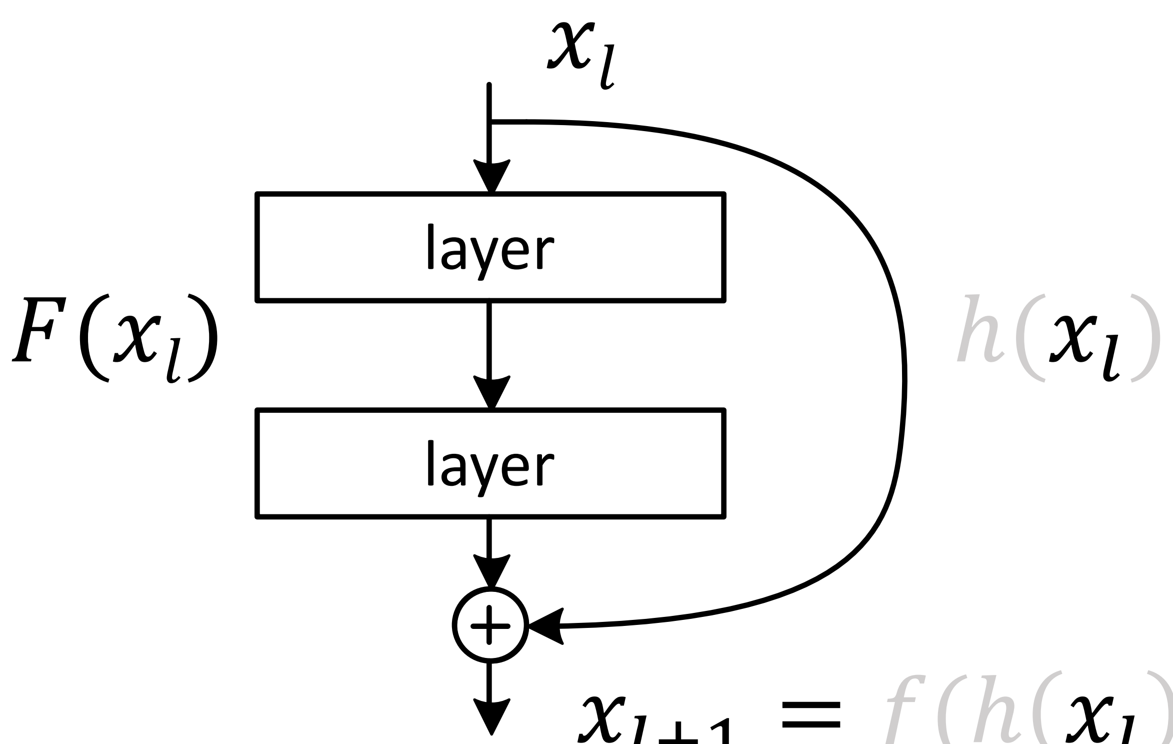


$$x_{l+1} = f(h(x_l) + F(x_l))$$

- shortcut mapping: h = identity
- ~~after-add mapping: f = ReLU~~
- What if f = identity?

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". ECCV 2016.

On identity mappings for **optimization**



$$x_{l+1} = f(h(x_l) + F(x_l))$$

- shortcut mapping: h = identity
- ~~after-add mapping: f = ReLU~~
- What if f = identity?

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". ECCV 2016.

Very smooth forward propagation

$$x_{l+1} = x_l + F(x_l)$$

$$x_{l+2} = x_{l+1} + F(x_{l+1})$$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. ECCV 2016.

Very smooth forward propagation

$$x_{l+1} = x_l + F(x_l)$$

$$x_{l+2} = x_{l+1} + F(x_{l+1})$$

$$x_{l+2} = x_l + F(x_l) + F(x_{l+1})$$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. ECCV 2016.

Very smooth forward propagation

$$x_{l+1} = x_l + F(x_l)$$

$$x_{l+2} = x_{l+1} + F(x_{l+1})$$

$$x_{l+2} = x_l + F(x_l) + F(x_{l+1})$$

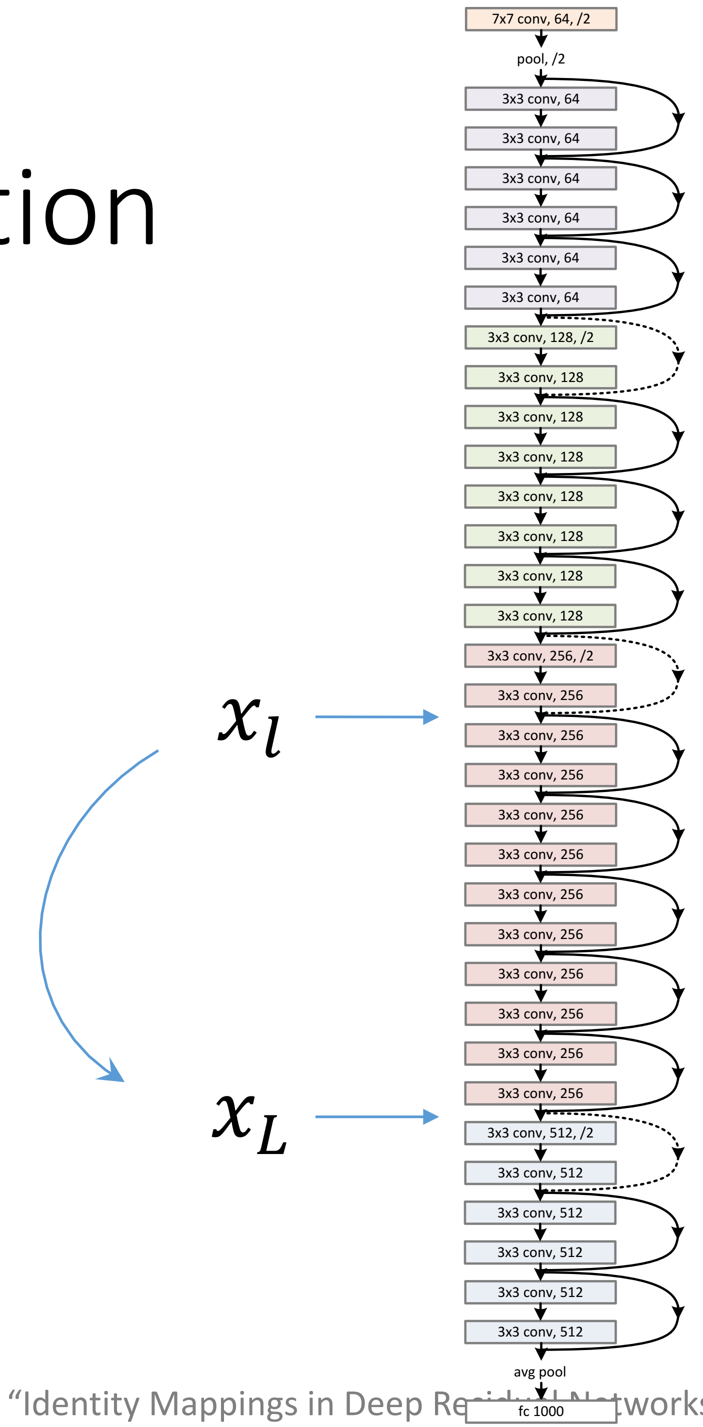
$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. ECCV 2016.

Very smooth forward propagation

$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$$

- Any x_l is directly forward-prop to any x_L, plus residual.
- Any x_L is an additive outcome.
  - in contrast to multiplicative: $x_L = \prod_{i=l}^{L-1} W_i x_l$



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. ECCV 2016.

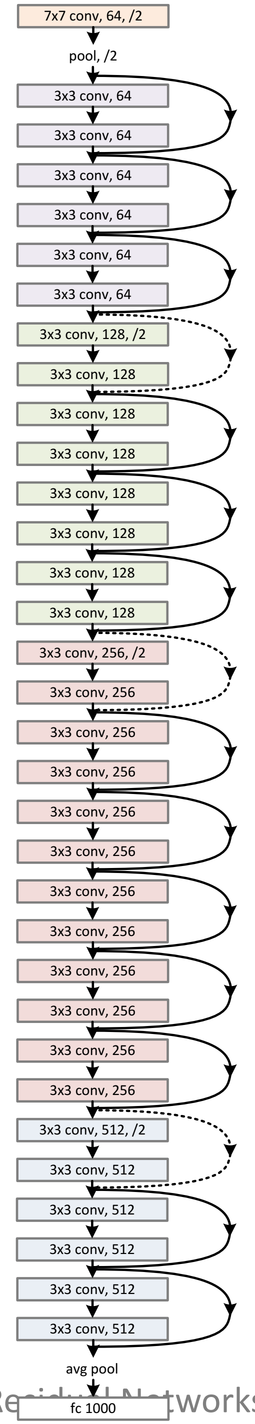
Very smooth backward propagation

$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$$

$$\frac{\partial E}{\partial x_l} = \frac{\partial E}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial E}{\partial x_L}\left(1 + \frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i)\right)$$

$$\frac{\partial E}{\partial x_l}$$

$$\frac{\partial E}{\partial x_L}$$

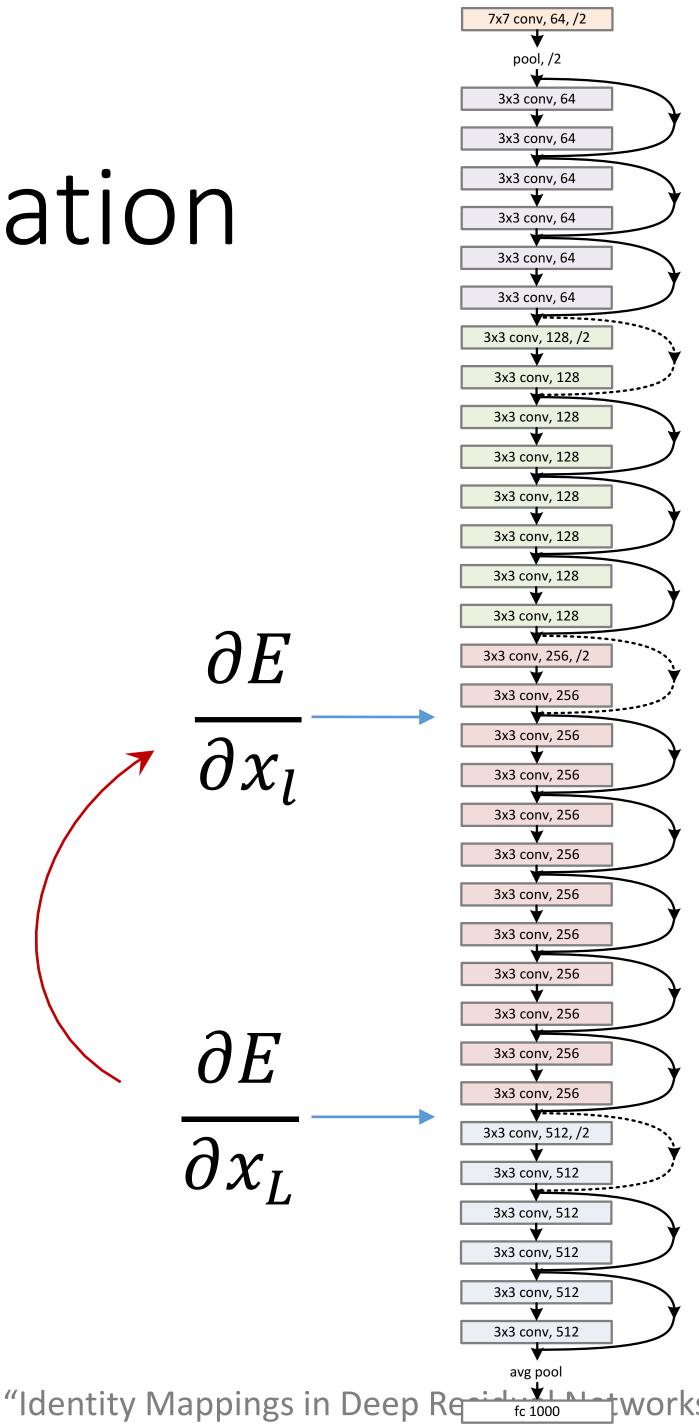


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. ECCV 2016.

Very smooth backward propagation

$$\frac{\partial E}{\partial x_l} = \frac{\partial E}{\partial x_L}\left(1 + \frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i)\right)$$

- Any $\frac{\partial E}{\partial x_L}$ is directly back-prop to any $\frac{\partial E}{\partial x_l}$, plus residual.
- Any $\frac{\partial E}{\partial x_l}$ is additive; unlikely to vanish
  - in contrast to multiplicative: $\frac{\partial E}{\partial x_l} = \prod_{i=l}^{L-1} W_i \frac{\partial E}{\partial x_L}$



Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. ECCV 2016.

Residual for every layer

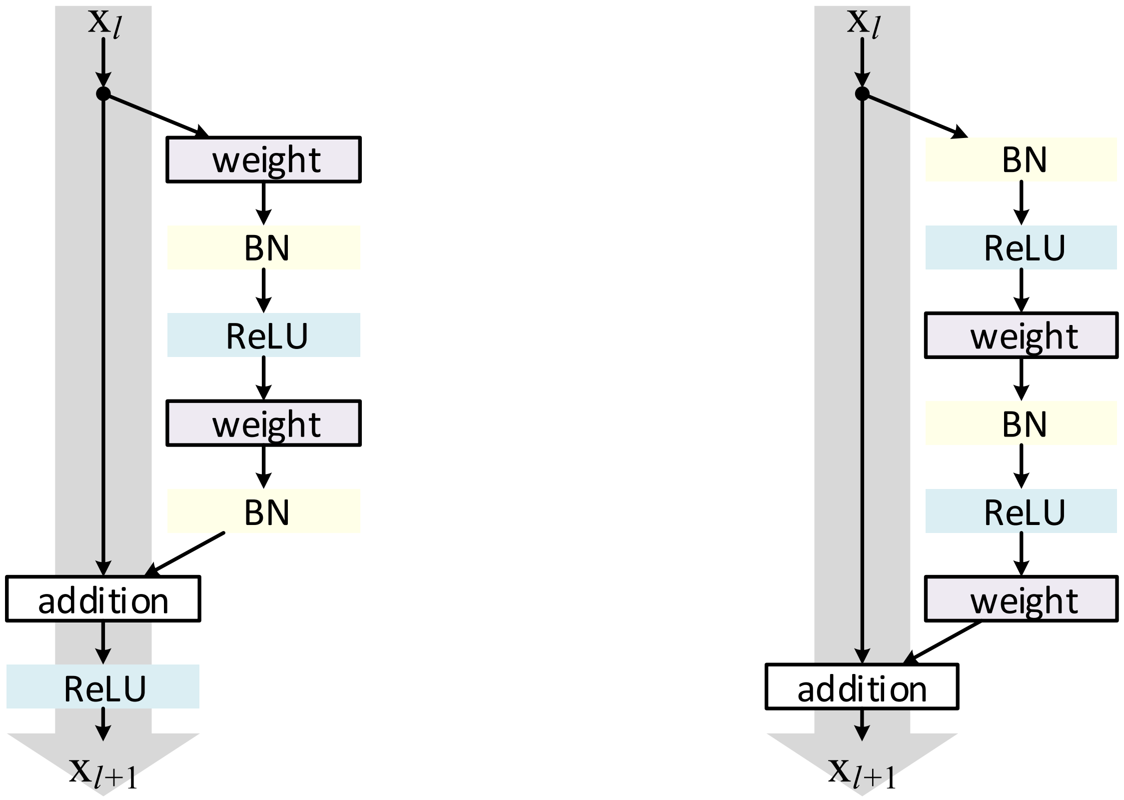
forward: $x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$

Enabled by:

- shortcut mapping: h = identity
- after-add mapping: f = identity

backward: $\frac{\partial E}{\partial x_l} = \frac{\partial E}{\partial x_L}\left(1 + \frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i)\right)$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. ECCV 2016.

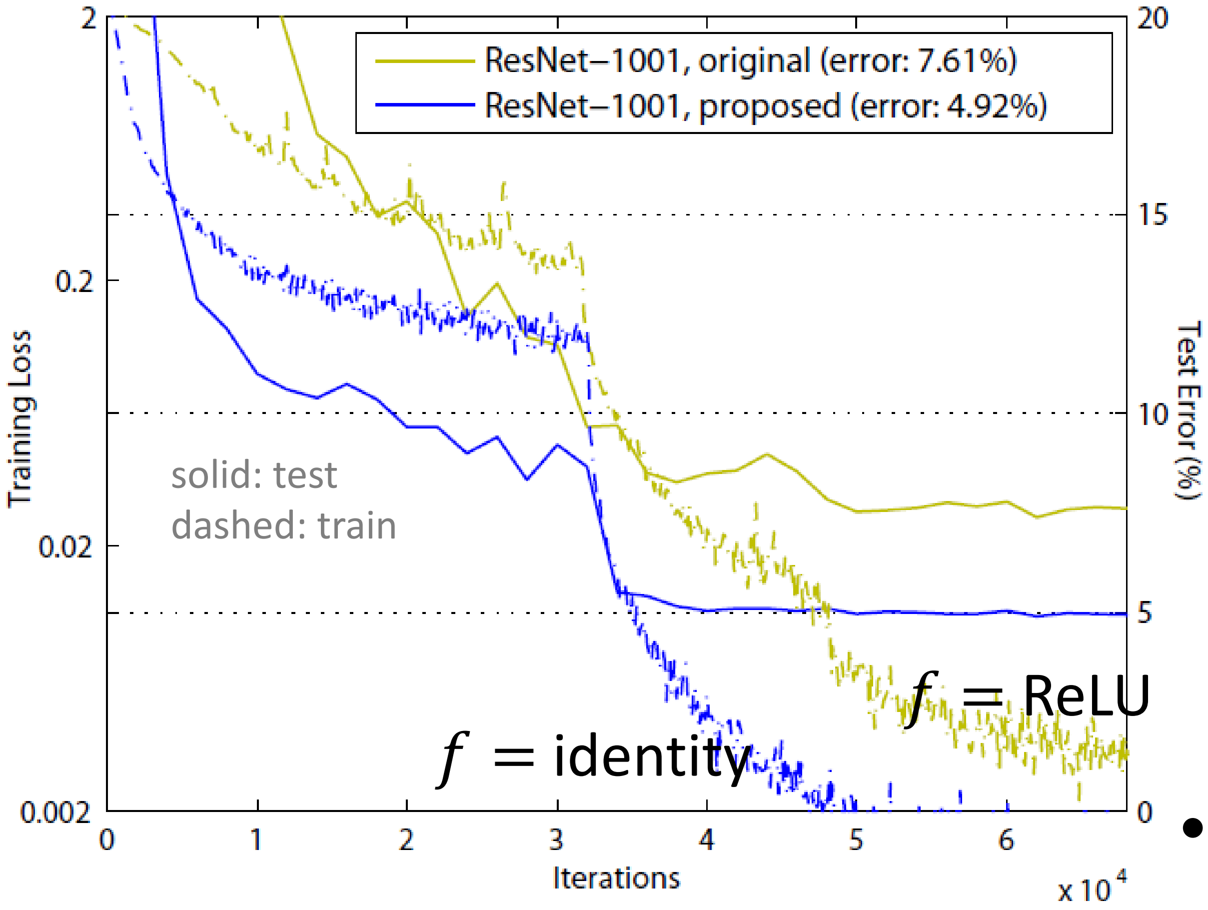


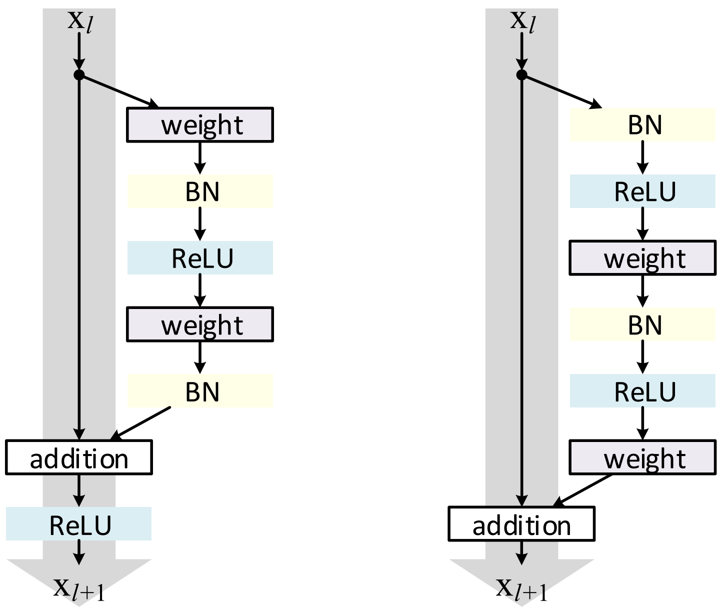
original ResNet

pre-activation
ResNet

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". ECCV 2016.

1001-layer ResNets on CIFAR-10





f = ReLU f = identity

- ReLU could also block prop when there are 1000 layers
- pre-activation design eases optimization (and improves generalization; see paper)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". ECCV 2016.

Comparisons on CIFAR-10/100

CIFAR-10

method	error (%)
NIN	8.81
DSN	8.22
FitNet	8.39
Highway	7.72
ResNet-110 (1.7M)	6.61
ResNet-1202 (19.4M)	7.93
ResNet-164, pre-activation (1.7M)	5.46
ResNet-1001, pre-activation (10.2M)	**4.92** (4.89±0.14)

CIFAR-100

method	error (%)
NIN	35.68
DSN	34.57
FitNet	35.04
Highway	32.39
ResNet-164 (1.7M)	25.16
ResNet-1001 (10.2M)	27.82
ResNet-164, pre-activation (1.7M)	24.33
ResNet-1001, pre-activation (10.2M)	**22.71** (22.68±0.22)

*all based on moderate augmentation

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. ECCV 2016.

ImageNet Experiments

ImageNet single-crop (320x320) val error

method	data augmentation	top-1 error (%)	top-5 error (%)
ResNet-152, original	scale	21.3	5.5
ResNet-152, pre-activation	scale	21.1	5.5
ResNet-200, original	scale	21.8	6.0
ResNet-200, pre-activation	scale	**20.7**	**5.3**
ResNet-200, pre-activation	scale + aspect ratio	**20.1***	**4.8***

*https://github.com/facebook/fb.resnet.torch/tree/master/pretrained#notes

training code and models available.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Identity Mappings in Deep Residual Networks". ECCV 2016.

From Classification to Detection

“Features matter”

“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%**

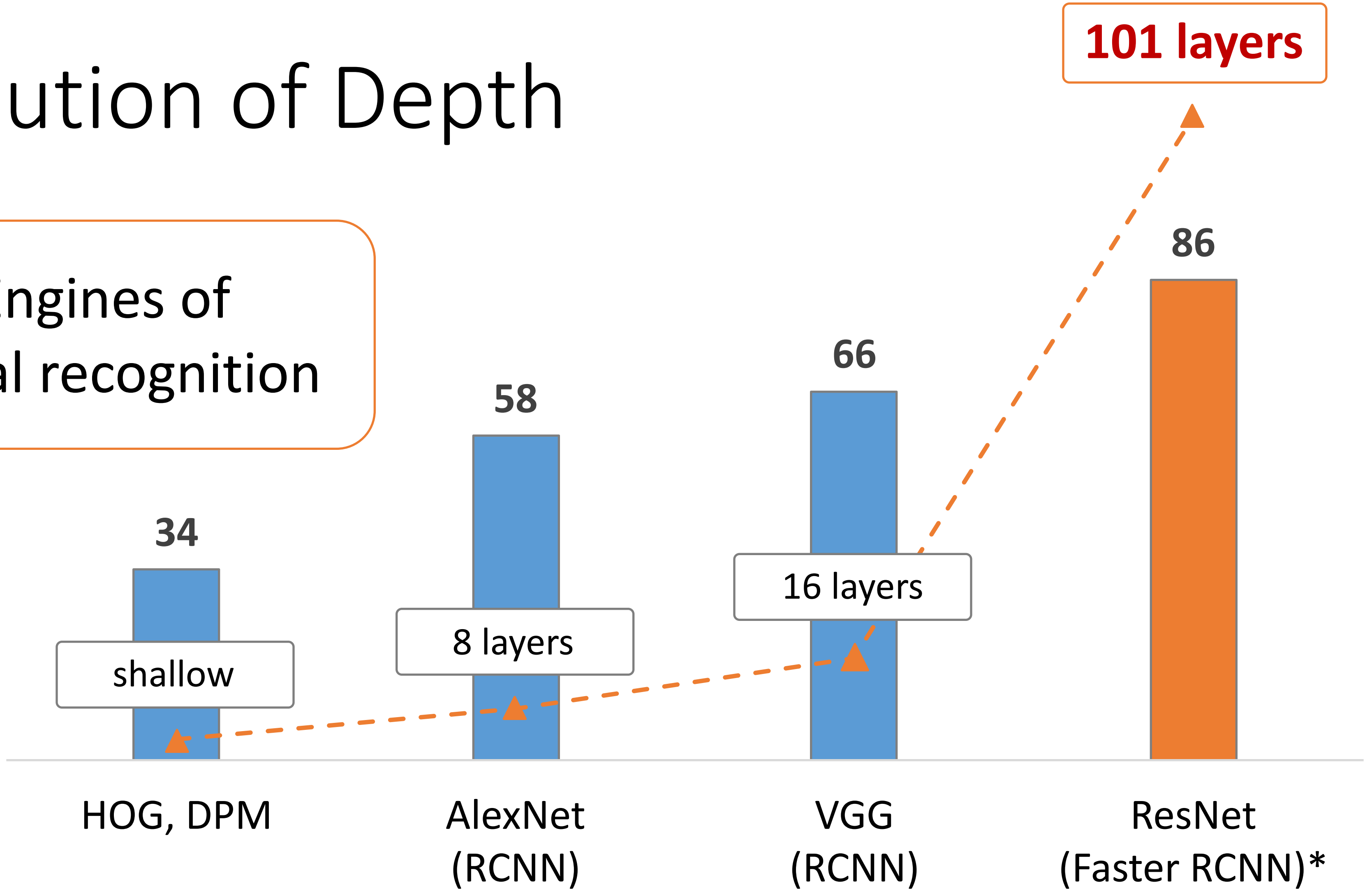
absolute 8.5% better!

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

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

Revolution of Depth

Engines of visual recognition

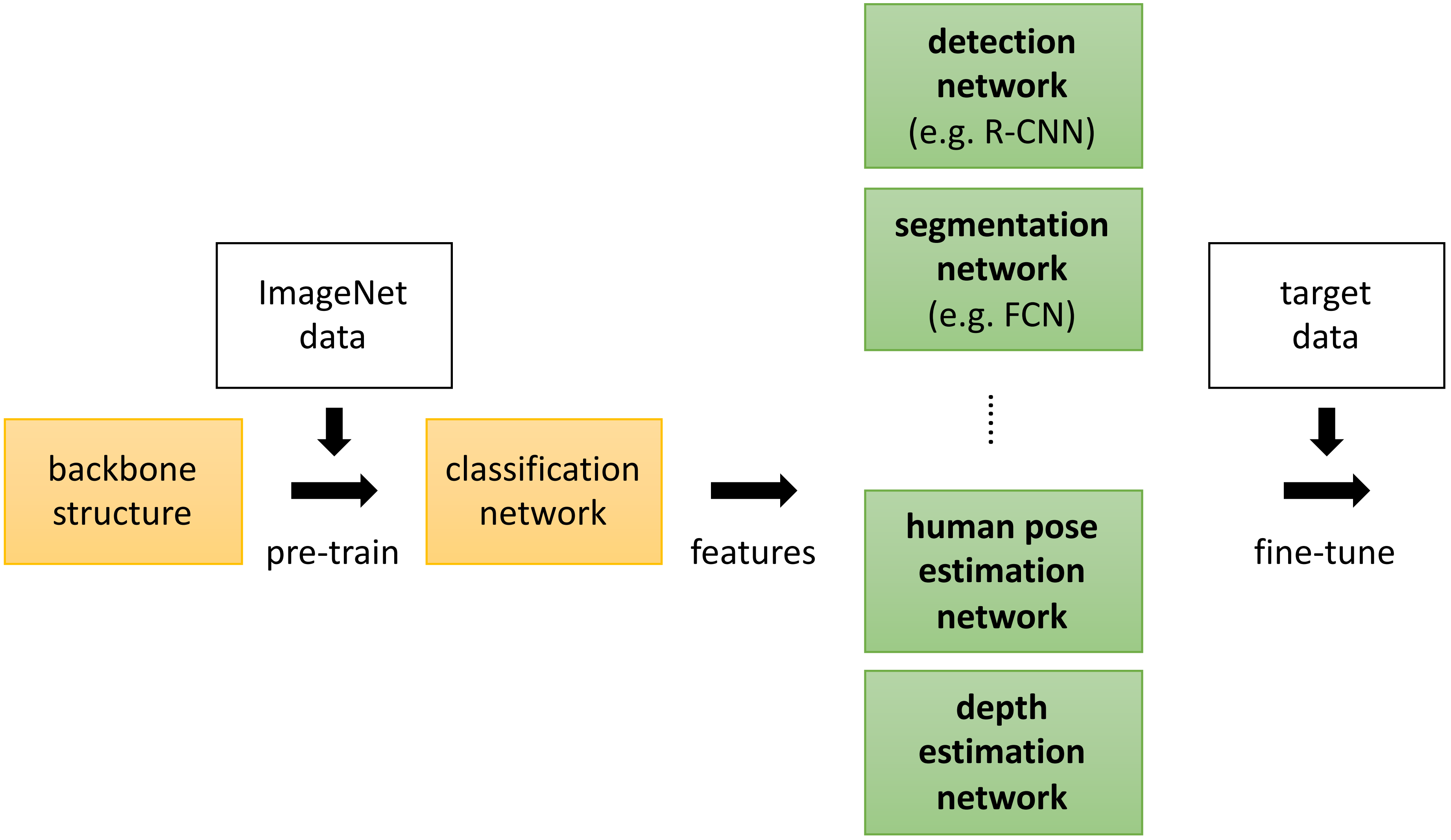


PASCAL VOC 2007 **Object Detection** mAP (%)

*w/ other improvements & more data

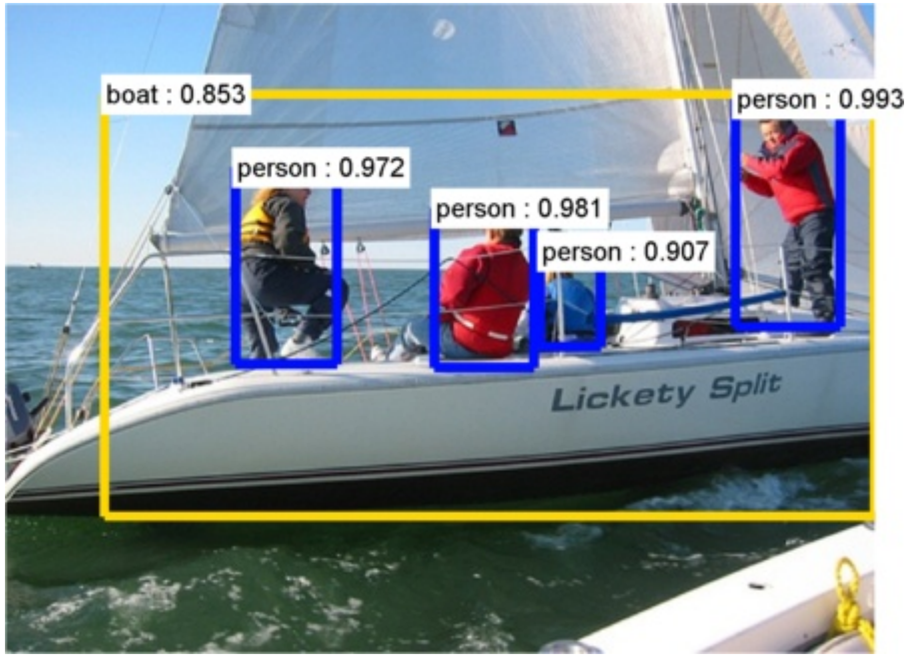
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Deep Learning for Computer Vision



Example: Object Detection

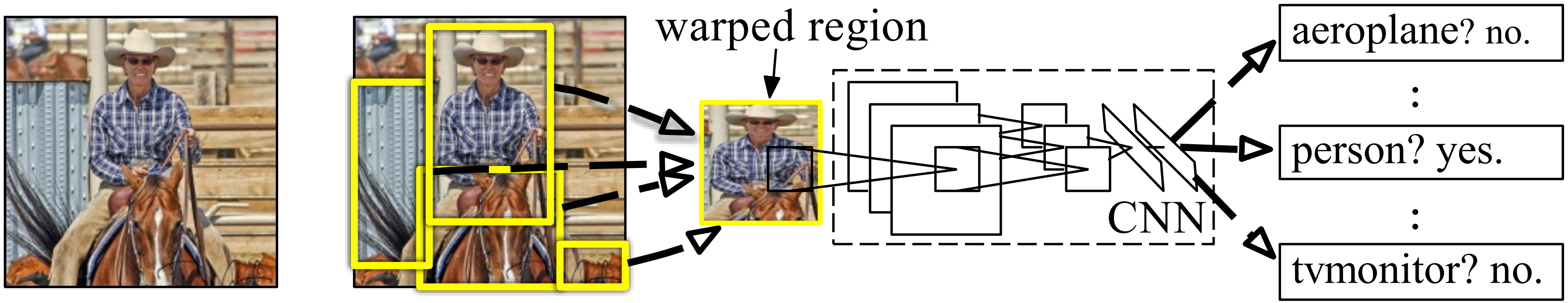
Image Classification
(what?)



Object Detection
(what + where?)

Object Detection: R-CNN

figure credit: R. Girshick et al.



input image

region proposals
~2,000

1 CNN for each region

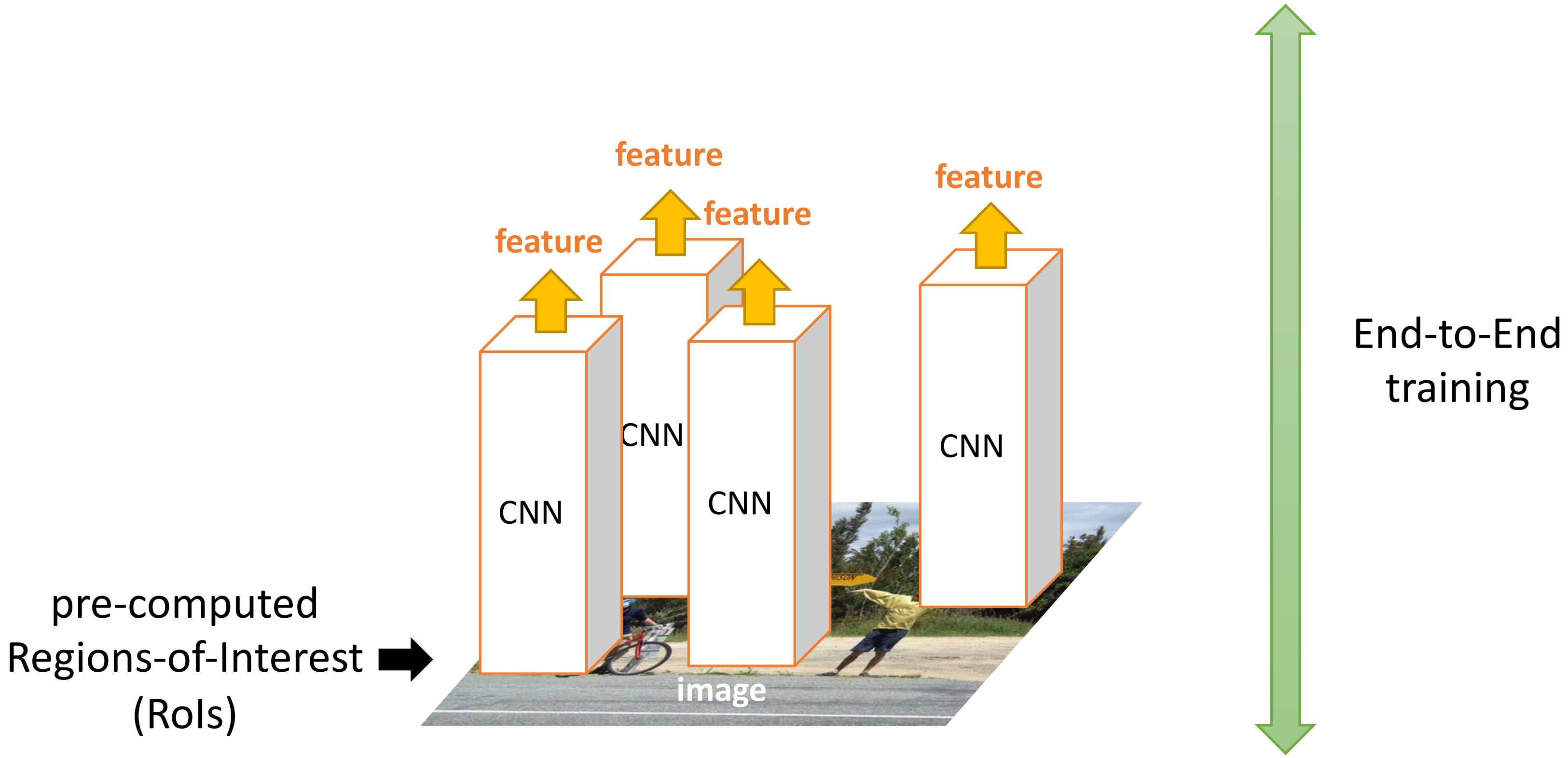
classify regions

Region-based **CNN** pipeline

Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014

Object Detection: R-CNN

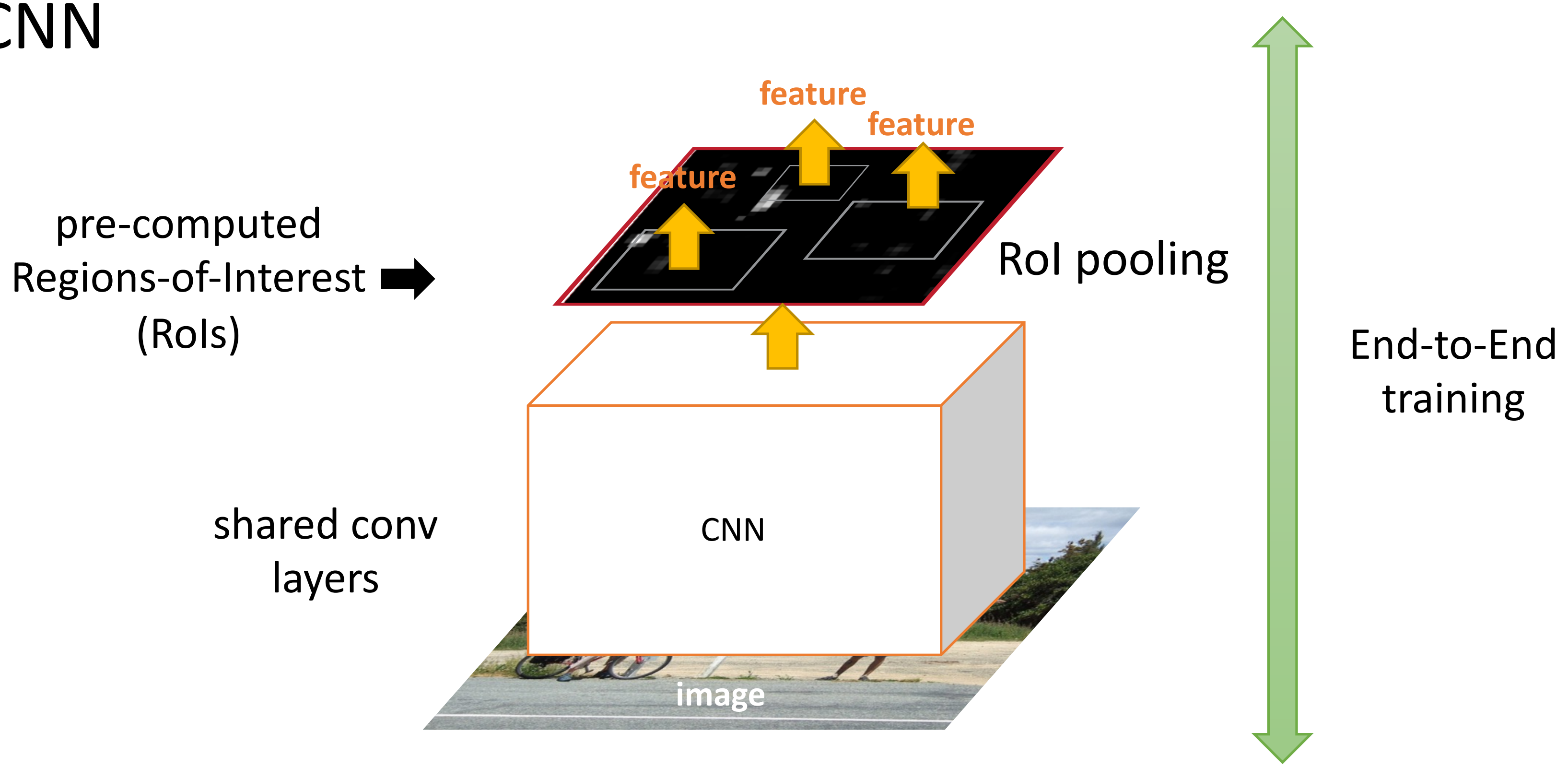
- R-CNN



Girshick, Donahue, Darrell, Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. CVPR 2014

Object Detection: Fast R-CNN

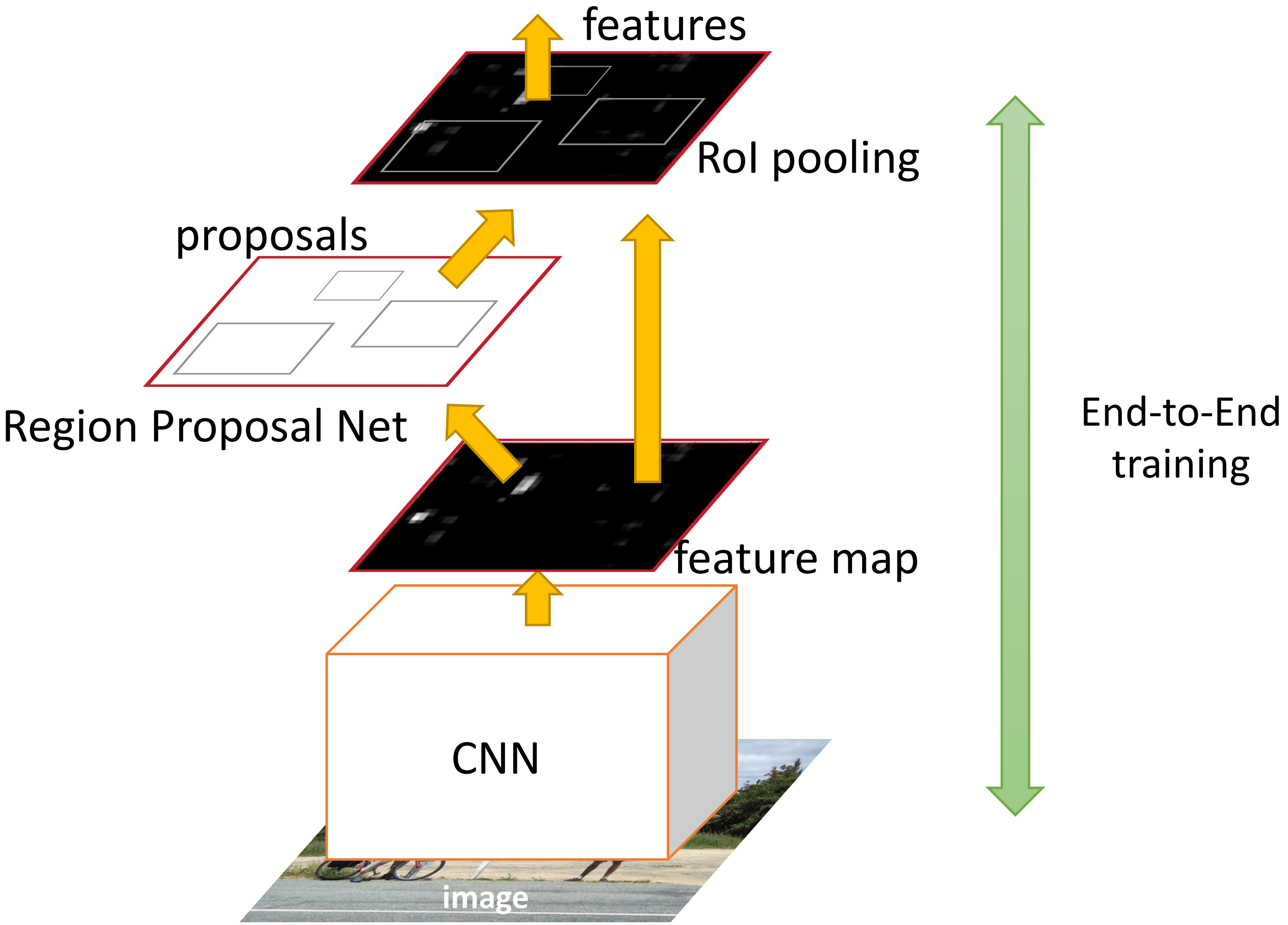
- Fast R-CNN



Girshick. Fast R-CNN. ICCV 2015

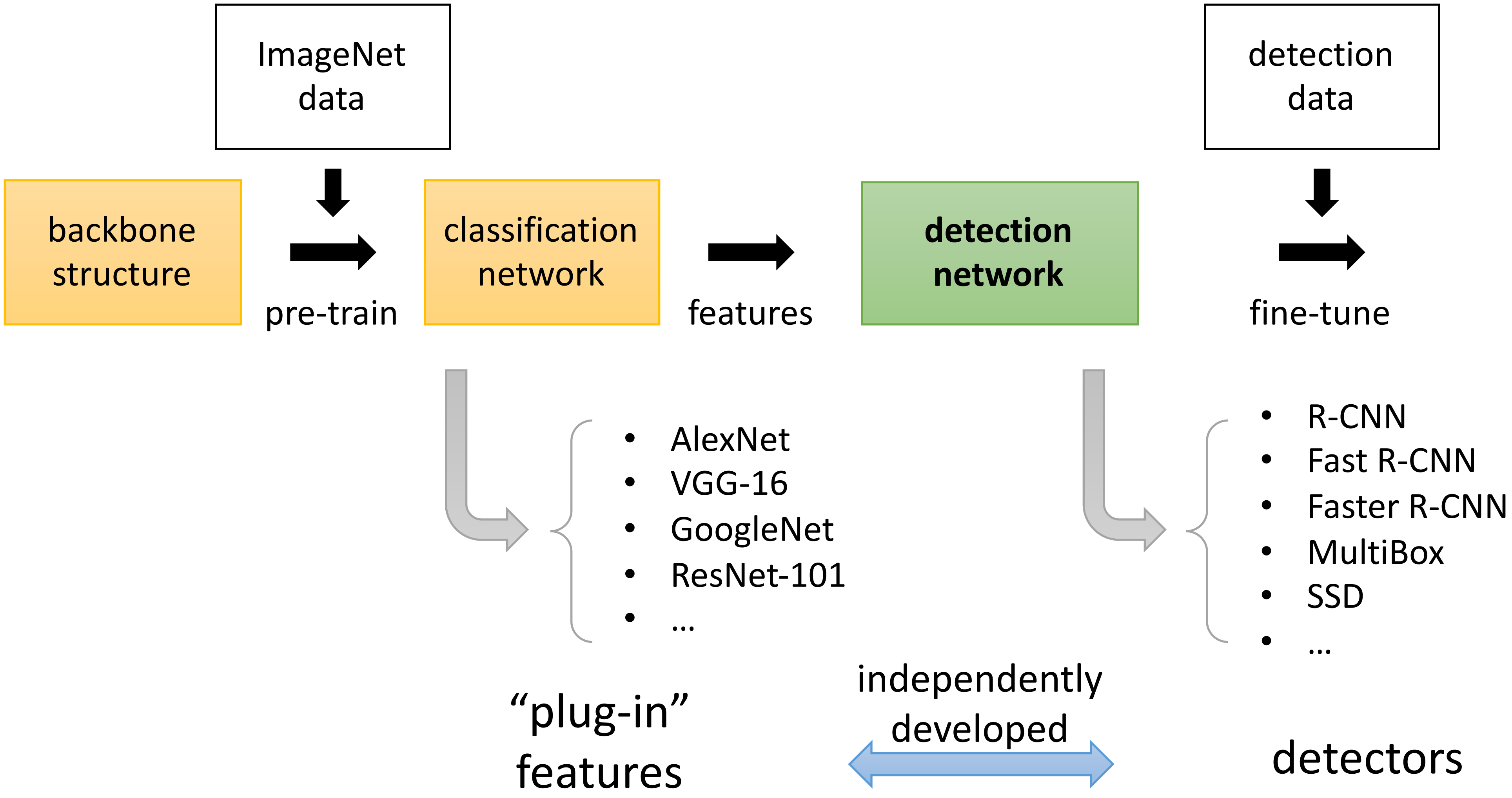
Object Detection: Faster R-CNN

- Faster R-CNN
  - Solely based on CNN
  - No external modules
  - Each step is end-to-end



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

Object Detection



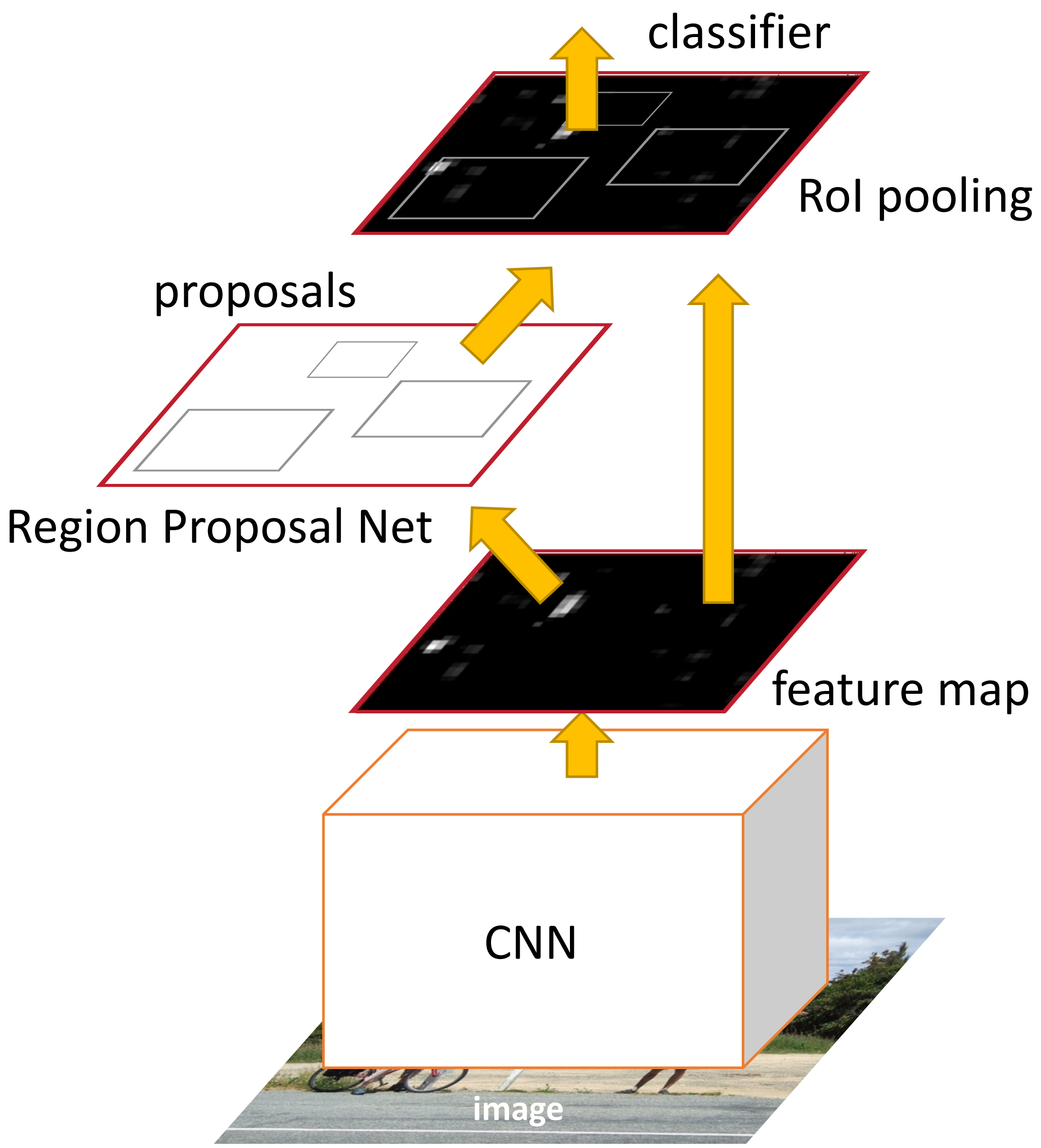
Object Detection

- 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-101 has 28% relative gain vs VGG-16

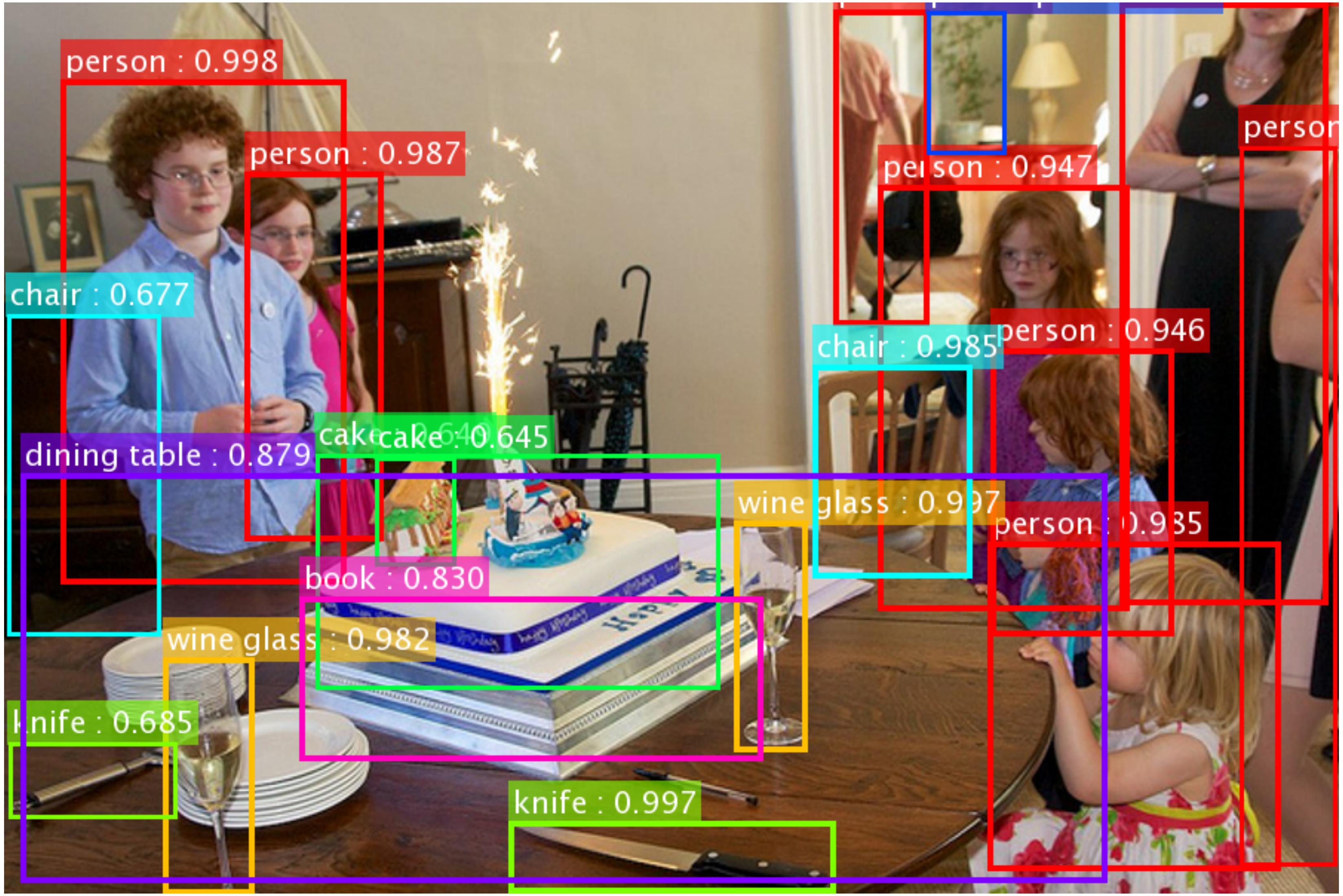


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. NIPS 2015.

Object Detection

- RPN learns proposals by extremely deep nets
  - We use only 300 proposals (no hand-designed proposals)
- Add components:
  - Iterative localization
  - Context modeling
  - Multi-scale testing
- All components are based on CNN features; all steps are end-to-end
- All benefit more from deeper features – cumulative gains!

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.

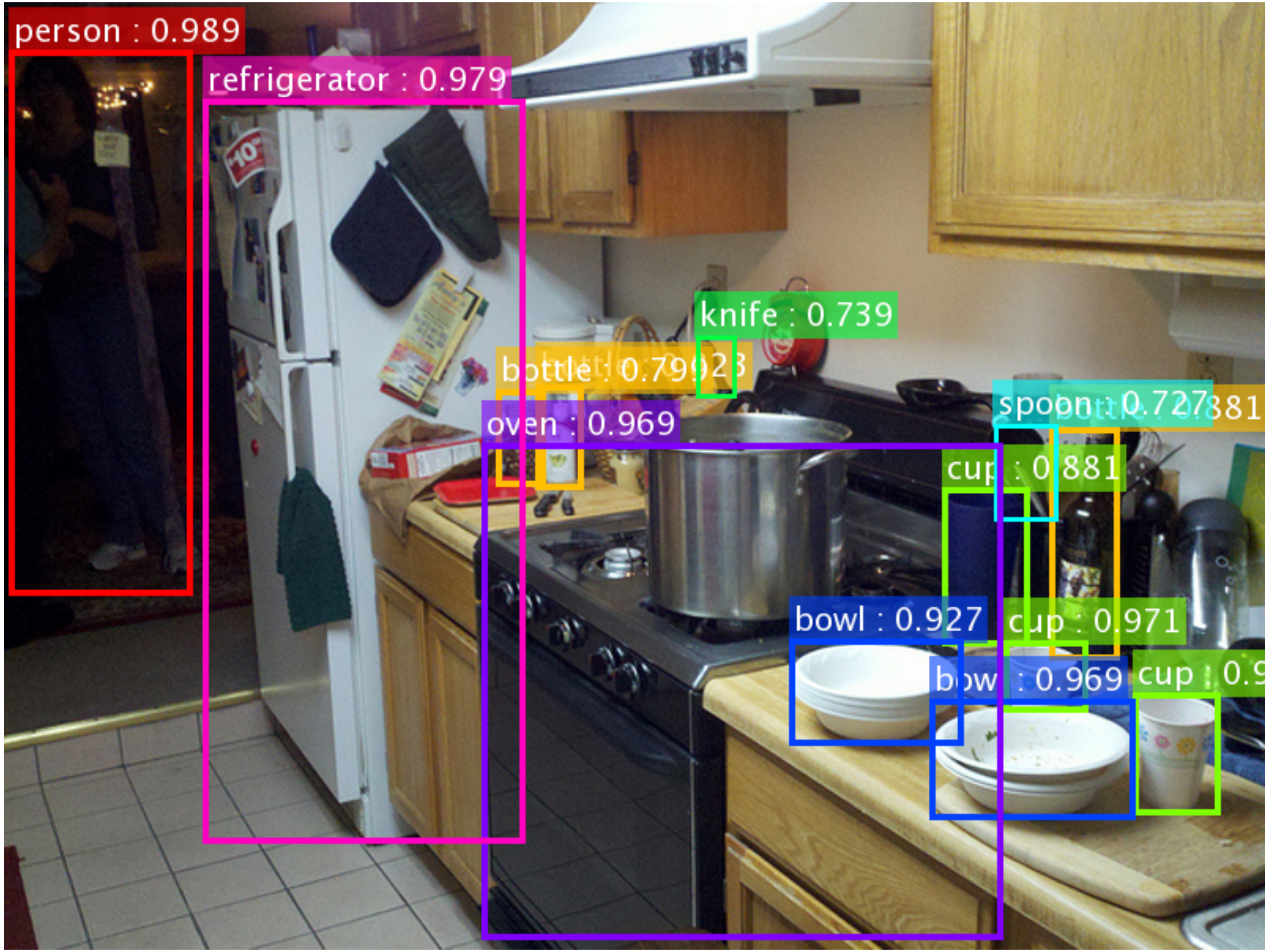


ResNet's object detection result on COCO

*the original image is from the COCO dataset

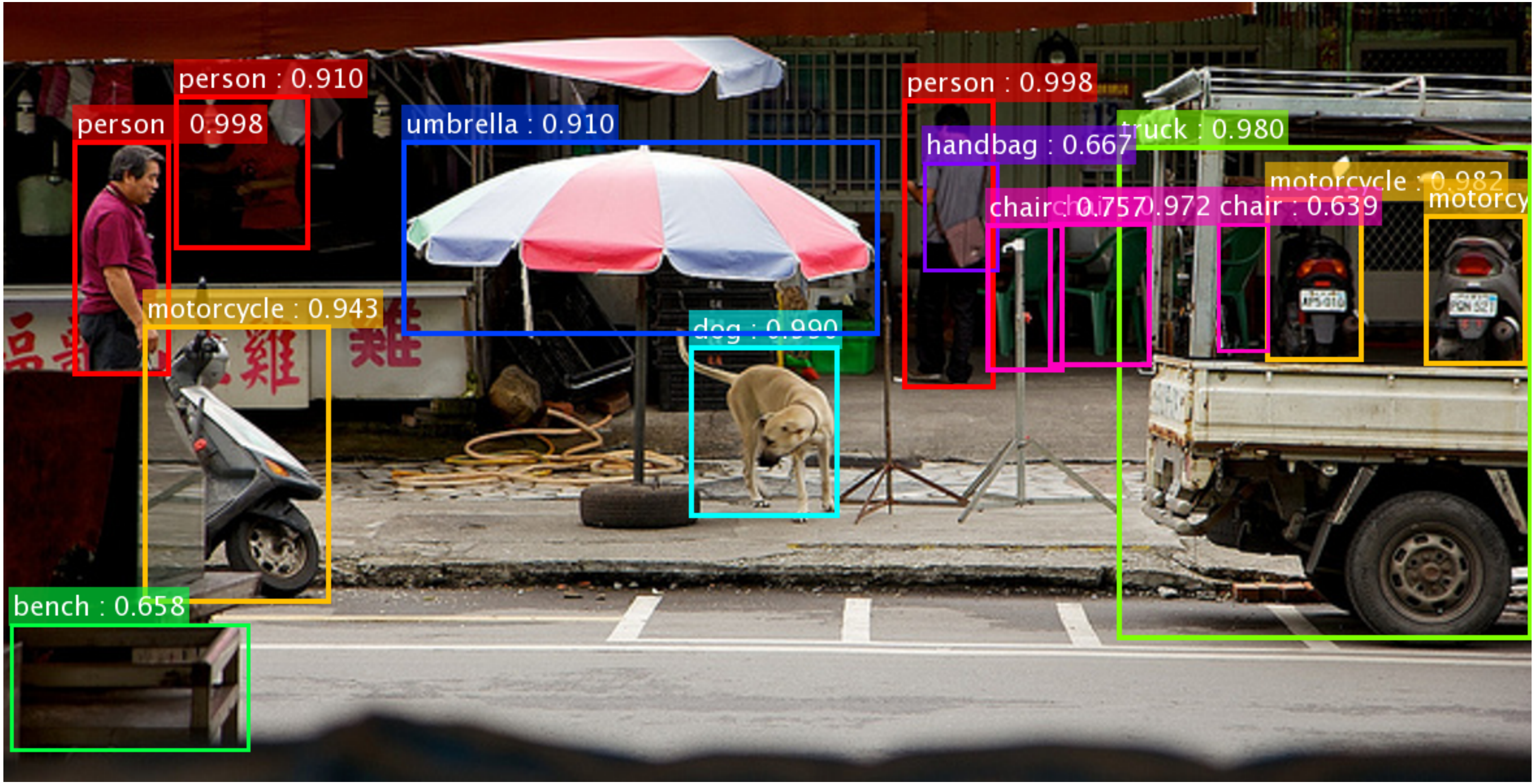
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". arXiv 2015.

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



*the original image is from the COCO dataset

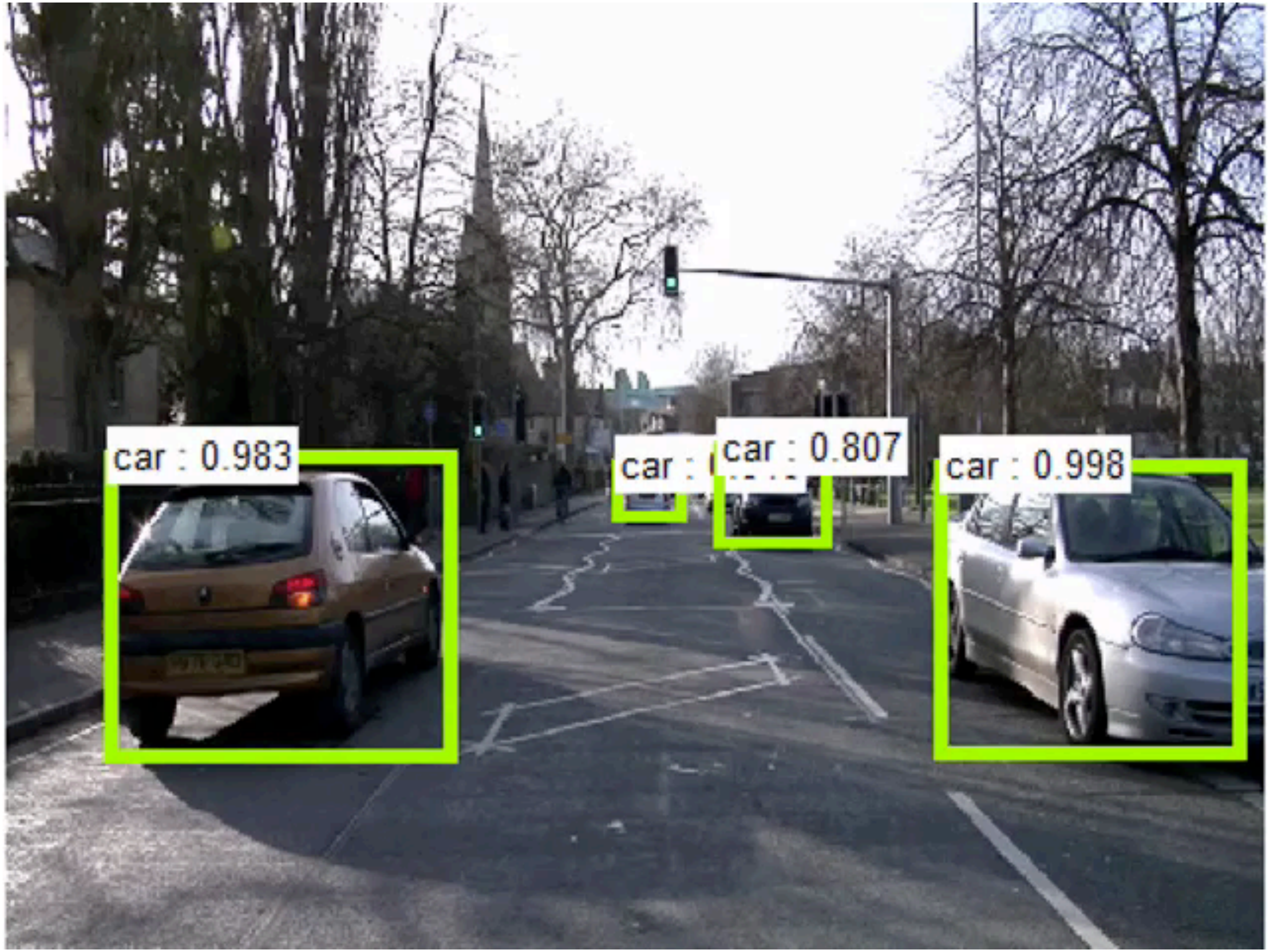
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015.
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. NIPS 2015.



*the original image is from the COCO dataset

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

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



Results on real video. Models trained on MS COCO (80 categories).
(frame-by-frame; no temporal processing)

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

More Visual Recognition Tasks

ResNet-based methods lead on these benchmarks (incomplete list):

- ImageNet classification, detection, localization
- MS COCO detection, segmentation
- PASCAL VOC detection, segmentation
- Visual Question Answering Challenge 2016

- Human pose estimation [Newell et al 2016]
- Depth estimation [Laina et al 2016]
- Segment proposal [Pinheiro et al 2016]
- …

	mean	aero plane	bicycle	bird	boat	bottle	bus	car
DeepLabv2-CRF [?]	79.7	92.6	60.4	91.6	63.4	76.3	95.0	88.4
CASIA_SegResNet_CRF_COCO [?]	79.3	93.8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Adelaide_VeryDeep_FCN_VOC [?]	79.1	91.9	48.1	93.4	69.3	75.5	94.2	87.5
LRR_4x_COCO [?]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
CASIA_IVA_OASeg [?]	78.3	93.8	41.9	89.4	67.5	71.5	94.6	85.3
Oxford_TVG_HO_CRF [?]	77.9	92.5	59.1	90.3	70.6	74.4	92.4	84.1
Adelaide_Context_CNN_CRF_COCO [?]	77.8	92.9	39.6	84.0	67.9	75.3	92.7	83.8

ResNet-101

PASCAL **segmentation** leaderboard

	mean	aero plane	bicycle	bird	boat	bottle	bus	car	cat
Faster RCNN, ResNet (VOC+COCO) [?]	83.8	92.1	88.4	[illegible]	75.9	[illegible]	86.3	87.8	94.2
R-FCN, ResNet (VOC+COCO) [?]	82.0	89.5	88.3	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
OHEM+FRCN, VGG16, VOC+COCO [?]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SSD500 VGG16 VOC + COCO [?]	78.7	89.1	85.7	78.9	63.3	57.0	85.3	84.1	92.3
HFM_VGG16 [?]	77.5	88.8	85.1	76.8	64.8	61.4	85.0	84.1	90.0
IFRN_07+12 [?]	76.6	87.8	83.9	79.0	64.5	58.9	82.2	82.0	91.4
ION [?]	76.4	87.5	84.7	76.8	63.8	58.3	82.6	79.0	90.9

ResNet-101

PASCAL **detection** leaderboard

More Applications

ResNets have shown outstanding or promising results on:

- Visual Recognition
- Image Generation (Pixel RNN, Neural Art, etc.)
- Natural Language Processing (Very deep CNN)
- Speech Recognition
- Advertising, user prediction

Resources

- Models and Code
  - https://github.com/KaimingHe/deep-residual-networks
- Many available implementations
  (see https://github.com/KaimingHe/deep-residual-networks)
  - Facebook AI Research's Torch ResNet:
    https://github.com/facebook/fb.resnet.torch
  - Torch, CIFAR-10, with ResNet-20 to ResNet-110, training code, and curves: code
  - Lasagne, CIFAR-10, with ResNet-32 and ResNet-56 and training code: code
  - Neon, CIFAR-10, with pre-trained ResNet-32 to ResNet-110 models, training code, and curves: code
  - Torch, MNIST, 100 layers: blog, code
  - A winning entry in Kaggle's right whale recognition challenge: blog, code
  - Neon, Place2 (mini), 40 layers: blog, code
  - …….

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

References

Classification

- "ImageNet Classification with Deep Convolutional Neural Networks", Krizhevsky et al. NIPS 2012
- "Visualizing and Understanding Convolutional Networks", Zeiler & Fergus. ECCV 2014
- "Very Deep Convolutional Networks for Large-Scale Image Recognition", Simonyan & Zisserman. ICLR 2015
- "Going deeper with convolutions", Szegedy et al. CVPR 2015
- "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", Szegedy et al. ICML 2015
- "Deep Residual Learning for Image Recognition", He et al. CVPR 2016

Detection

- "Rich feature hierarchies for accurate object detection and semantic segmentation", Girshick et al. CVPR 2014
- "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition", He et al. ECCV 2014
- "Fast R-CNN", Girshick. ICCV 2015
- "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", Ren et al. NIPS 2015

Segmentation

- "Fully Convolutional Networks for Semantic Segmentation", Long et al. CVPR 2015
- "Learning to Segment Object Candidates", Pinhero et al. NIPS 2015

Language

- "Long-term Recurrent Convolutional Networks for Visual Recognition and Description", Donahue et al. CVPR 2015
- "Deep visual-semantic alignments for generating image descriptions", Karpathy & Fei-Fei. CVPR 2015

Super-Resolution

- "Learning a Deep Convolutional Network for Image Super-Resolution", Dong et al. ECCV 2014

Neural Art

- "A Neural Algorithm of Artistic Style", Gatys et al. CVPR 2016

Generative models

- "Generative Adversarial Nets", Goodfellow et al. NIPS 2015