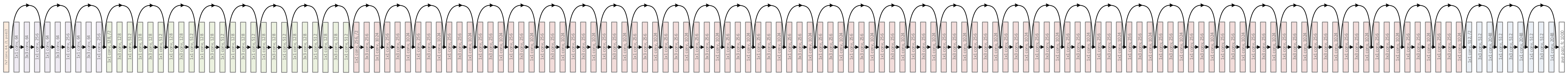


Learning Deep Features for Visual Recognition

CVPR 2017 Tutorial

Kaiming He

Facebook AI Research (FAIR)

covering joint work with:

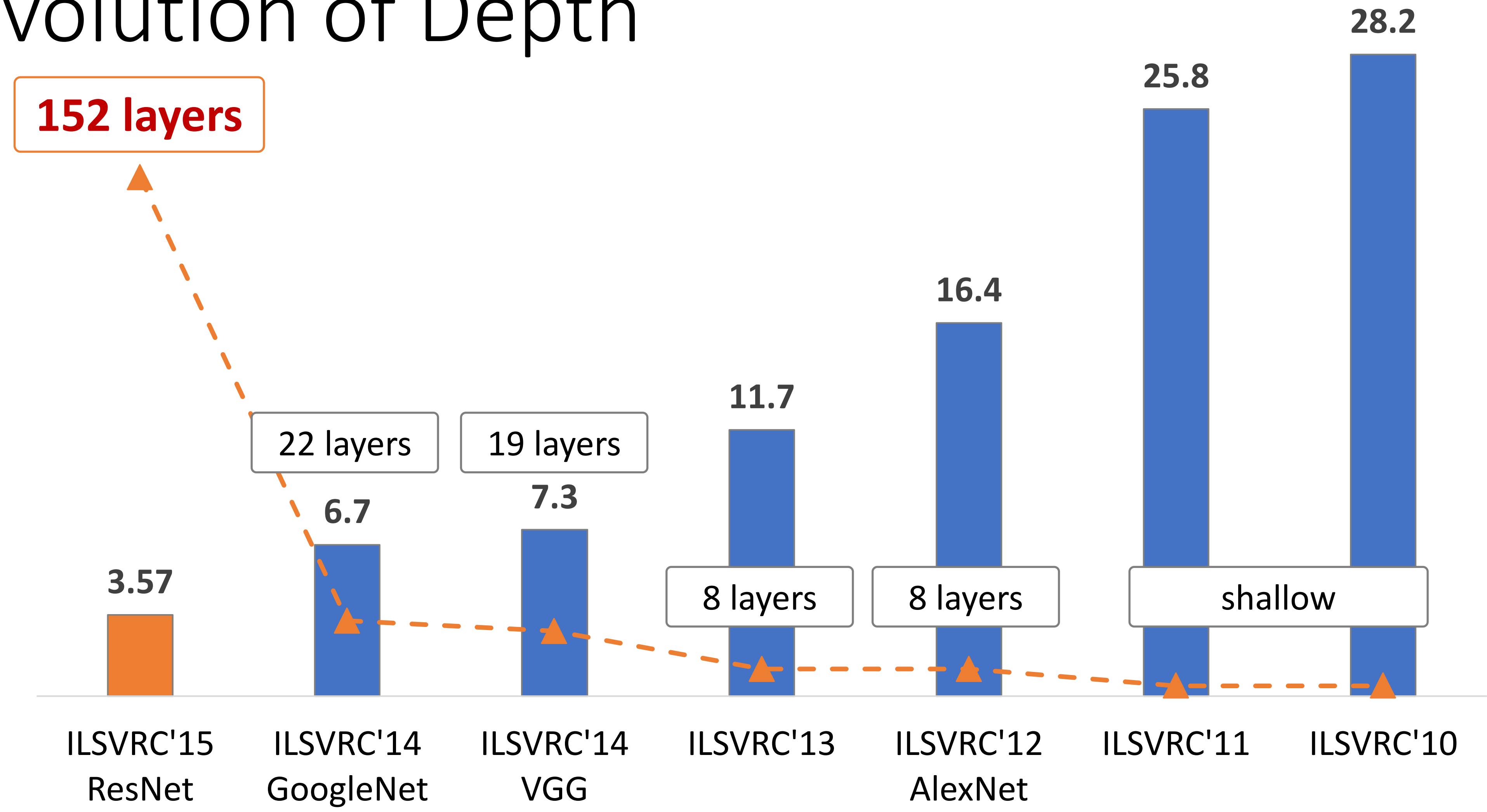
Xiangyu Zhang, Shaoqing Ren, Jian Sun, Saining Xie, Zhuowen Tu, Ross Girshick, Piotr Dollar

Outline

- Introduction
- Convolutional Neural Networks: Recap
  - LeNet, AlexNet, VGG, GoogleNet; Batch Norm
- ResNet
- ResNeXt

slides will be available online

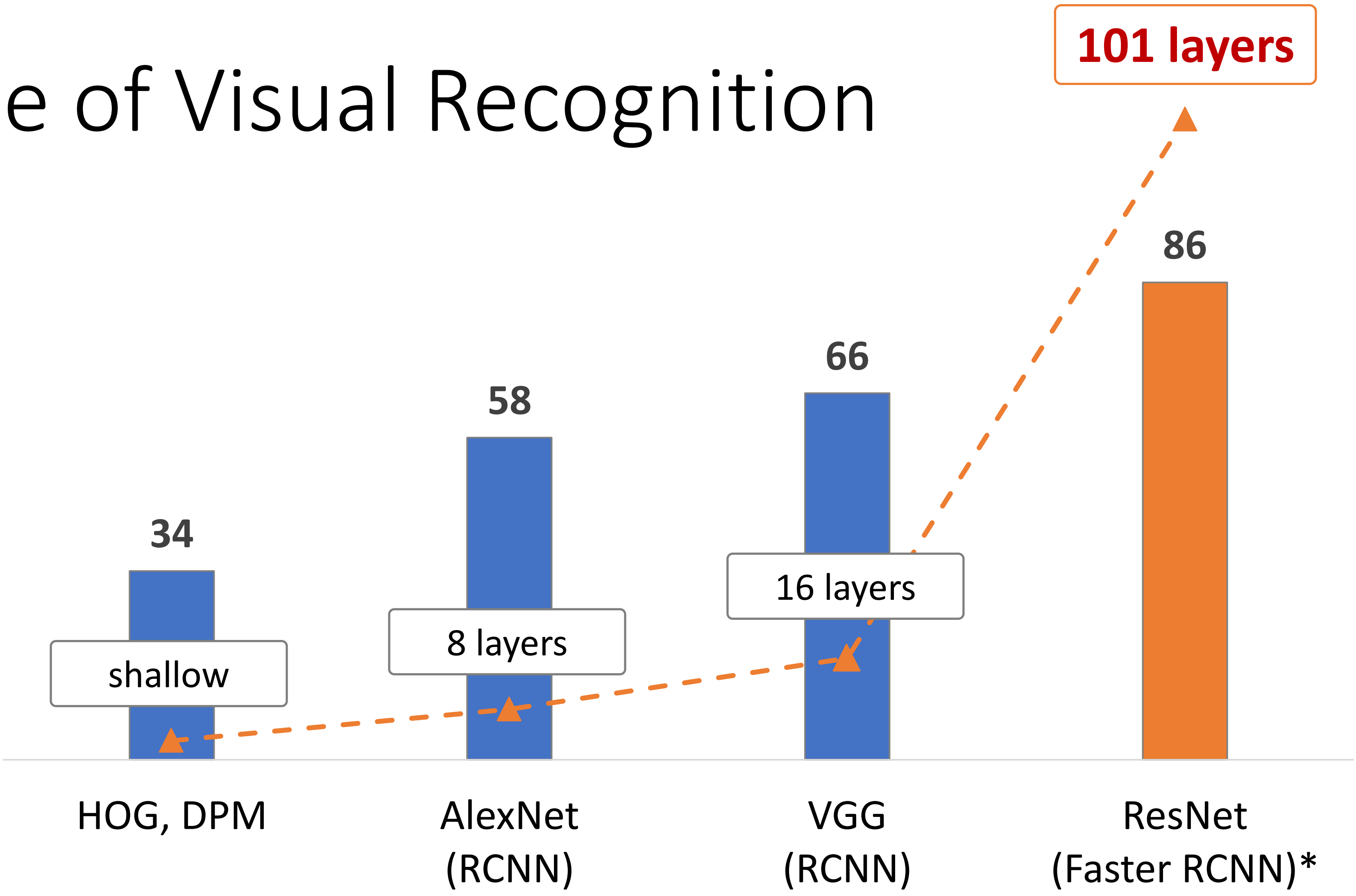
Revolution of Depth



ImageNet Classification top-5 error (%)

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

Engine of Visual Recognition



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

*w/ other improvements & more data

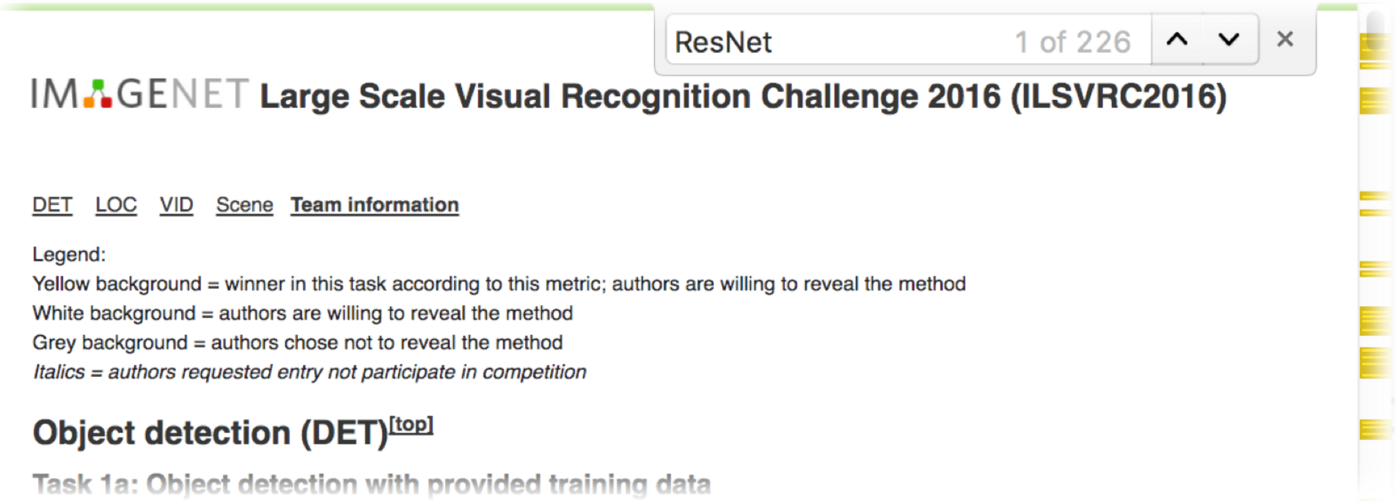
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Engine of Visual Recognition

ResNets/extensions are leading models on popular benchmarks

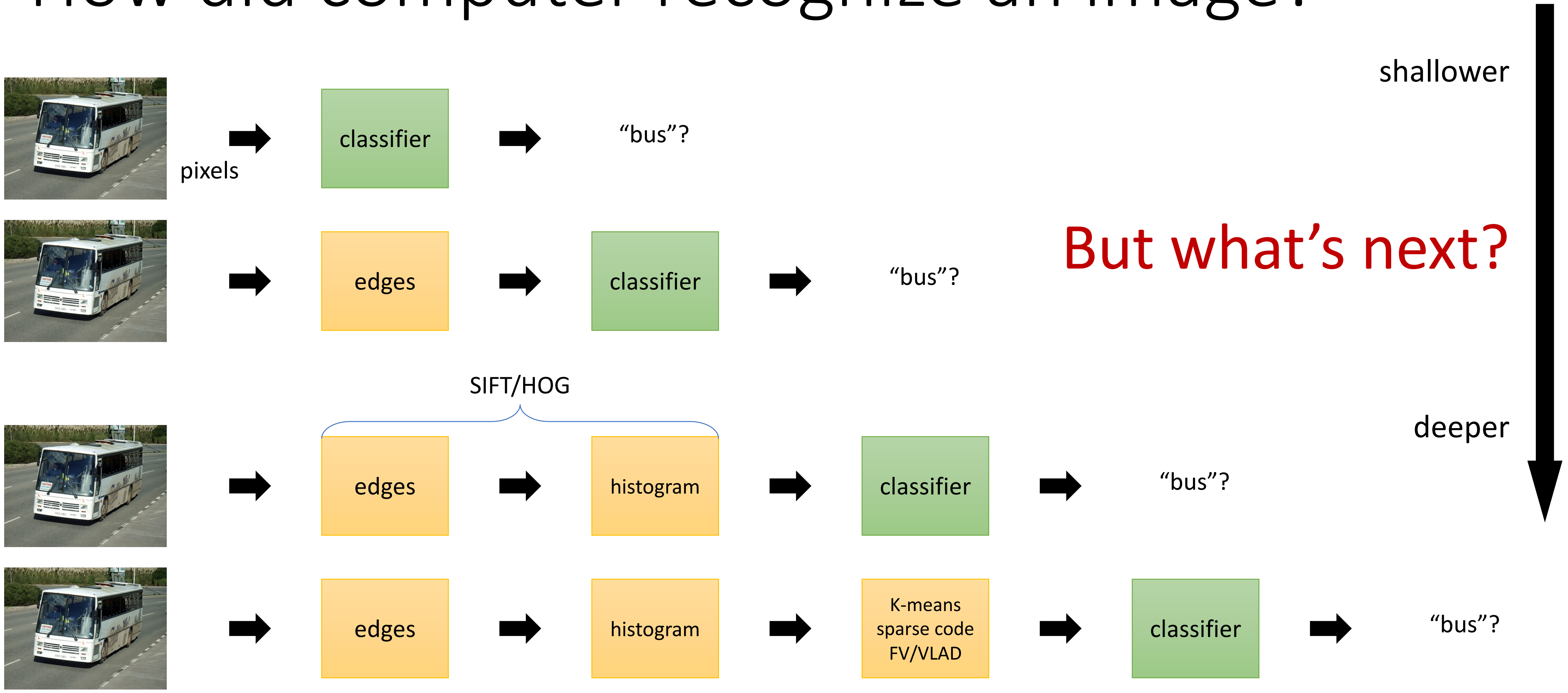
- Detection: COCO/VOC
- Segmentation: COCO/VOC/ADE/Cityscape
- Visual Reasoning: VQA/CLEVR
- Video: UCF101/HMDB
- ...

Search "ResNet" on ILSVRC2016 result page returns 226 entries



Source: Ross Girshick

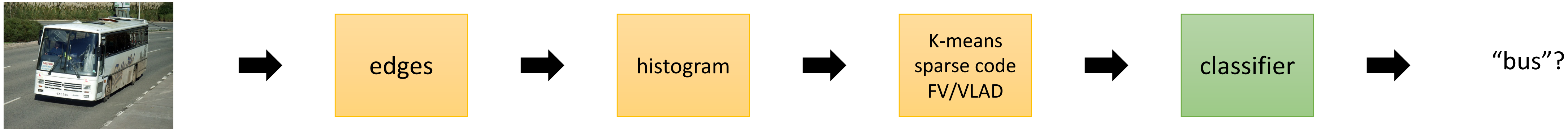
How did computer recognize an image?



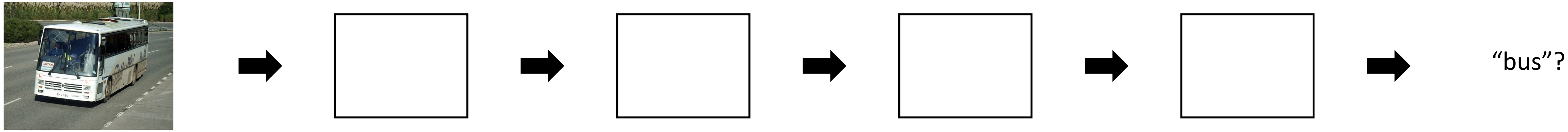
[Lowe 1999, 2004], [Sivic & Zisserman 2003], [Dalal & Triggs 2005], [Grauman & Darrell 2005]
[Lazebnik et al 2006], [Perronnin & Dance 2007], [Yang et al 2009], [Jégou et al 2010], ……

Learning Deep Features

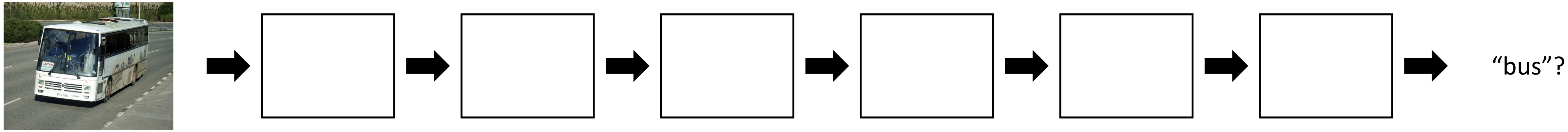
Specialized components, domain knowledge required



Generic components/“layers”, less domain knowledge



Repeat **elementary** layers: going deeper

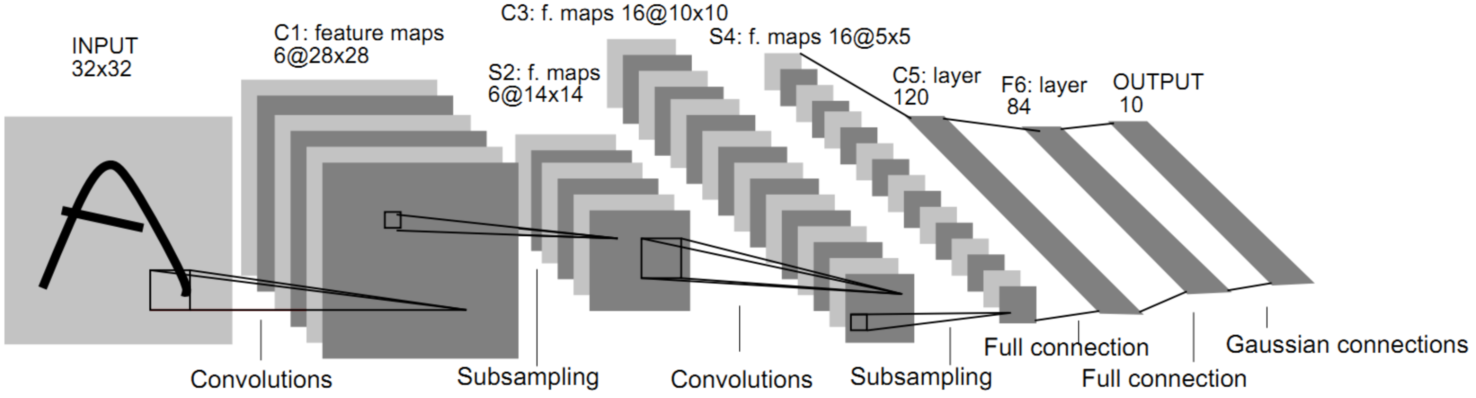


- Richer solution space
- End-to-end learning by BackProp

Convolutional Neural Networks: Recap

LeNet, AlexNet, VGG, GoogleNet; Batch Norm,…

LeNet



- Convolution:
  - locally-connected
  - spatially weight-sharing
    - weight-sharing is a key in DL (e.g., RNN shares weights temporally)
- Subsampling
- Fully-connected outputs
- Train by BackProp

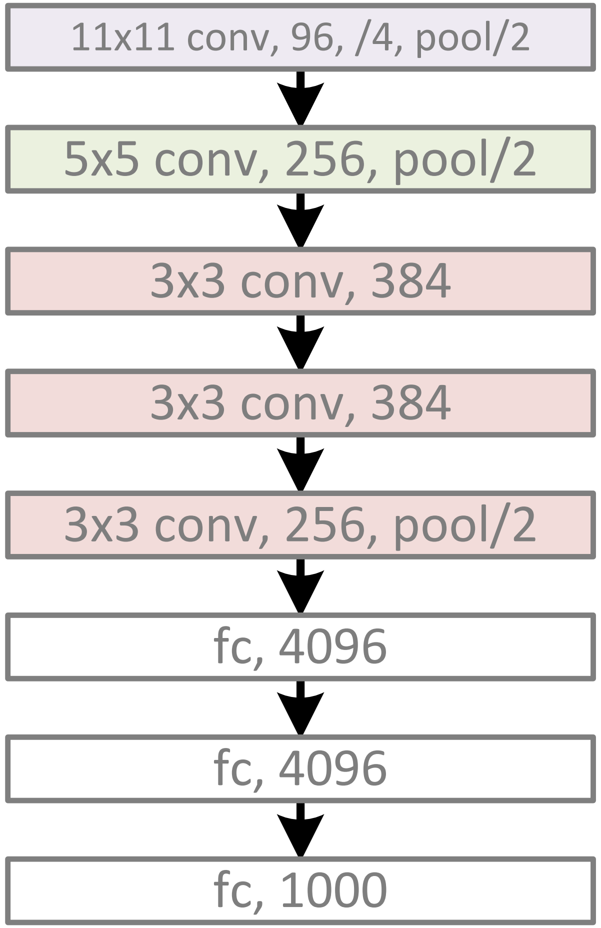
- All are still the basic components of modern ConvNets!

"Gradient-based learning applied to document recognition", LeCun et al. 1998
"Backpropagation applied to handwritten zip code recognition", LeCun et al. 1989

AlexNet

LeNet-style backbone, plus:

- ReLU [Nair & Hinton 2010]
  - "RevoLUtion of deep learning"*
  - Accelerate training; better grad prop (vs. tanh)
- Dropout [Hinton et al 2012]
  - In-network ensembling
  - Reduce overfitting (might be instead done by BN)
- Data augmentation
  - Label-preserving transformation
  - Reduce overfitting



*Quote Christian Szegedy

"ImageNet Classification with Deep Convolutional Neural Networks", Krizhevsky, Sutskever, Hinton. NIPS 2012

VGG-16/19

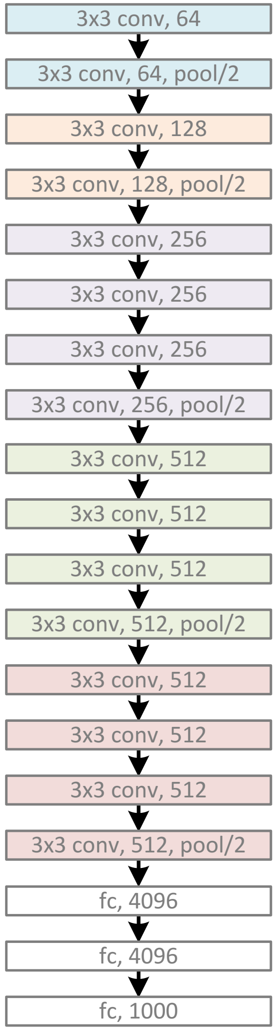
“16 layers are beyond my imagination!”

-- after ILSVRC 2014 result was announced.

Simply “Very Deep”!

- Modularized design
  - 3x3 Conv as the module
  - Stack the same module
  - Same computation for each module (1/2 spatial size => 2x filters)

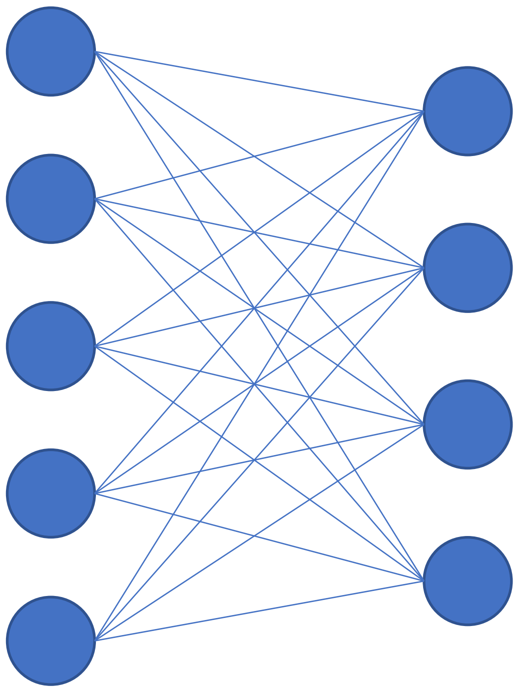
- Stage-wise training
  - VGG-11 => VGG-13 => VGG-16
  - We need a better initialization…



“Very Deep Convolutional Networks for Large-Scale Image Recognition”, Simonyan & Zisserman. arXiv 2014 (ICLR 2015)

Initialization

weight
W



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

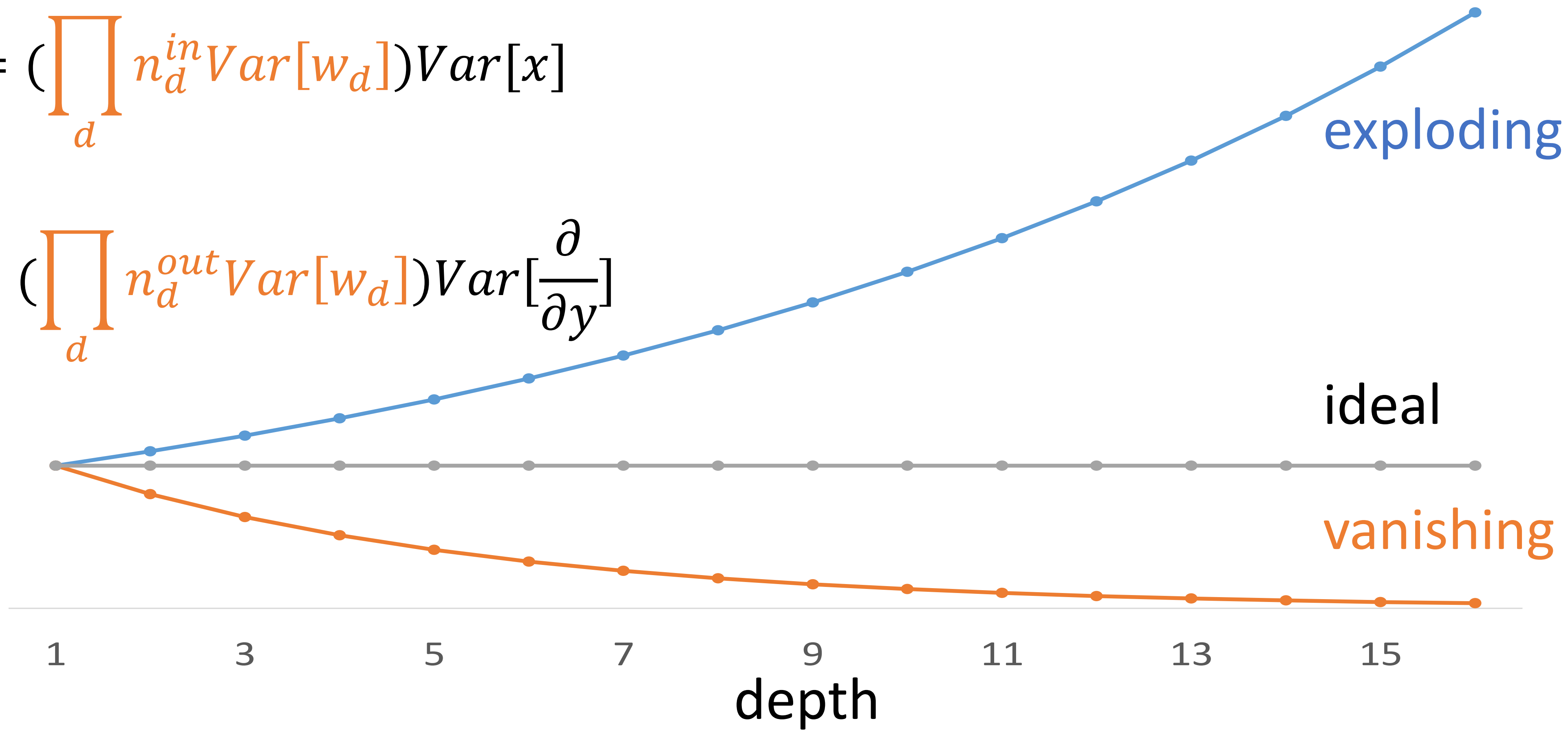
Both forward (response) and backward (gradient) signal can vanish/explode

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: "Xavier"

- Initialization under **linear** assumption

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

and

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

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

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

Initialization: "MSRA"

- Initialization under **ReLU**

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

and

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

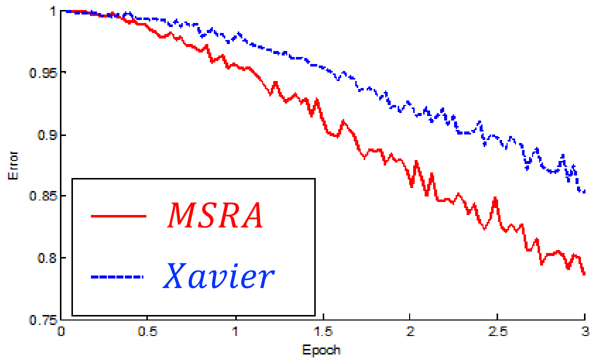
➡ $$\boxed{\begin{array}{c} \frac{1}{2} n_d^{in} Var[w_d] = 1 \\ \text{or} \\ \frac{1}{2} n_d^{out} Var[w_d] = 1 \end{array}}$$

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

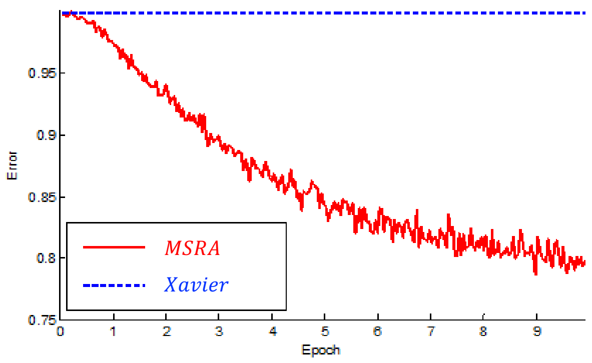
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". ICCV 2015.

Initialization

22-layer VGG-style



30-layer VGG-style



*Figures show the beginning of training

Xavier/MSRA init

- Required for training VGG-16/19 from scratch
- Deeper (>20) VGG-style nets can be trained w/ MSRA init
  - but deeper plain nets are not better (see ResNets)
- Recommended for newly initialized layers in fine-tuning
  - e.g., Fast/er RCNN, FCN, etc.

- $\sqrt{\frac{1}{n}}$or $\sqrt{\frac{2}{n}}$ doesn't directly apply to multi-branch nets (e.g., GoogleNet)
  - but the same derivation methodology is applicable
  - does not matter, if BN is applicable…

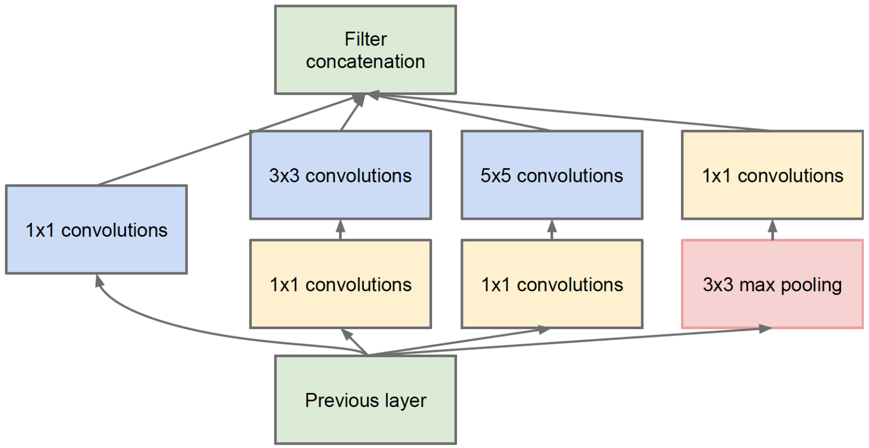
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". ICCV 2015.

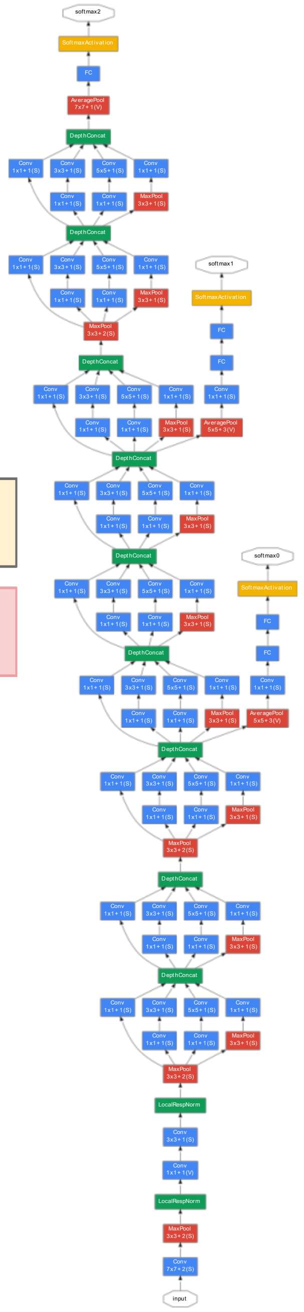
GoogleNet/Inception

Accurate with small footprint.

My take on GoogleNets:

- Multiple branches
  - e.g., 1x1, 3x3, 5x5, pool
- Shortcuts
  - stand-alone 1x1, merged by concat.
- Bottleneck
  - Reduce dim by 1x1 before expensive 3x3/5x5 conv



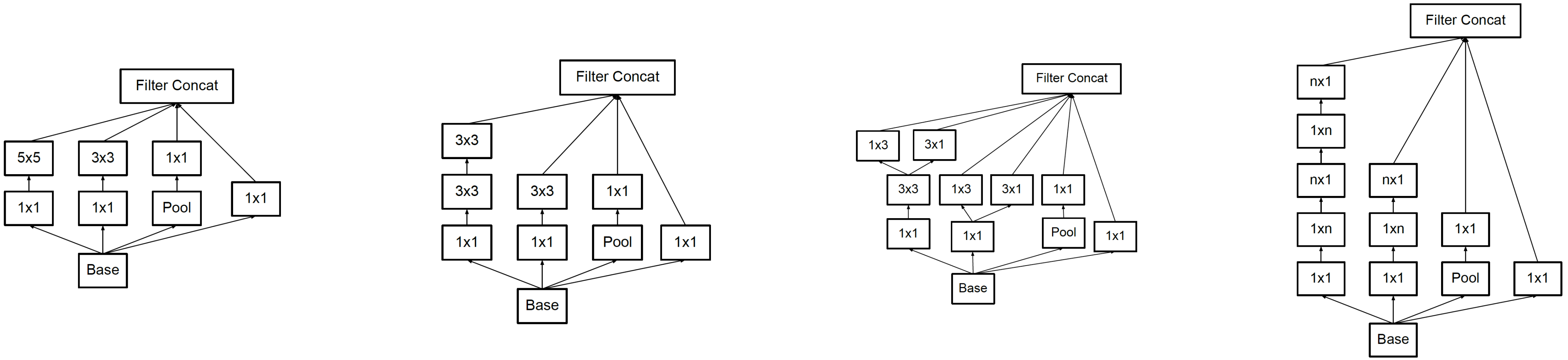


Szegedy et al. "Going deeper with convolutions". arXiv 2014 (CVPR 2015).

GoogleNet/Inception v1-v3

More templates, but the same 3 main properties are kept:

- Multiple branches
- Shortcuts (1x1, concate.)
- Bottleneck

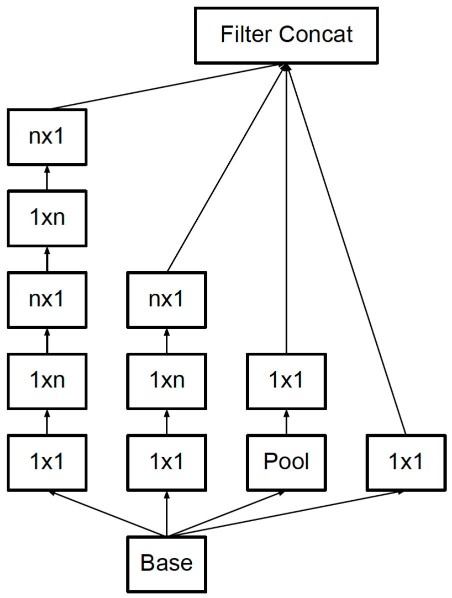


Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". arXiv 2015 (CVPR 2016).

Batch Normalization (BN)

- Recap: Xavier/MSRA init are not directly applicable for multi-branch nets

- Optimizing multi-branch ConvNets largely benefits from BN
  - including all Inceptions and ResNets



Ioffe & Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". ICML 2015.

Batch Normalization (BN)

- Recap: Normalizing image input (LeCun et al 1998 “Efficient Backprop”)
- Xavier/MSRA init: Analytic normalizing each layer
- BN: data-driven normalizing each layer, for each mini-batch
  - Greatly accelerate training
  - Less sensitive to initialization
  - Improve regularization

Ioffe & Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. ICML 2015.

Batch Normalization (BN)

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

- 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

Ioffe & 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 a batch of x
- Test mode:
  - μ, σ are pre-computed* on training set

Caution: make sure your BN usage is correct!
(this causes many of my bugs in my research experience!)

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

Ioffe & Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". ICML 2015.

Batch Normalization (BN)

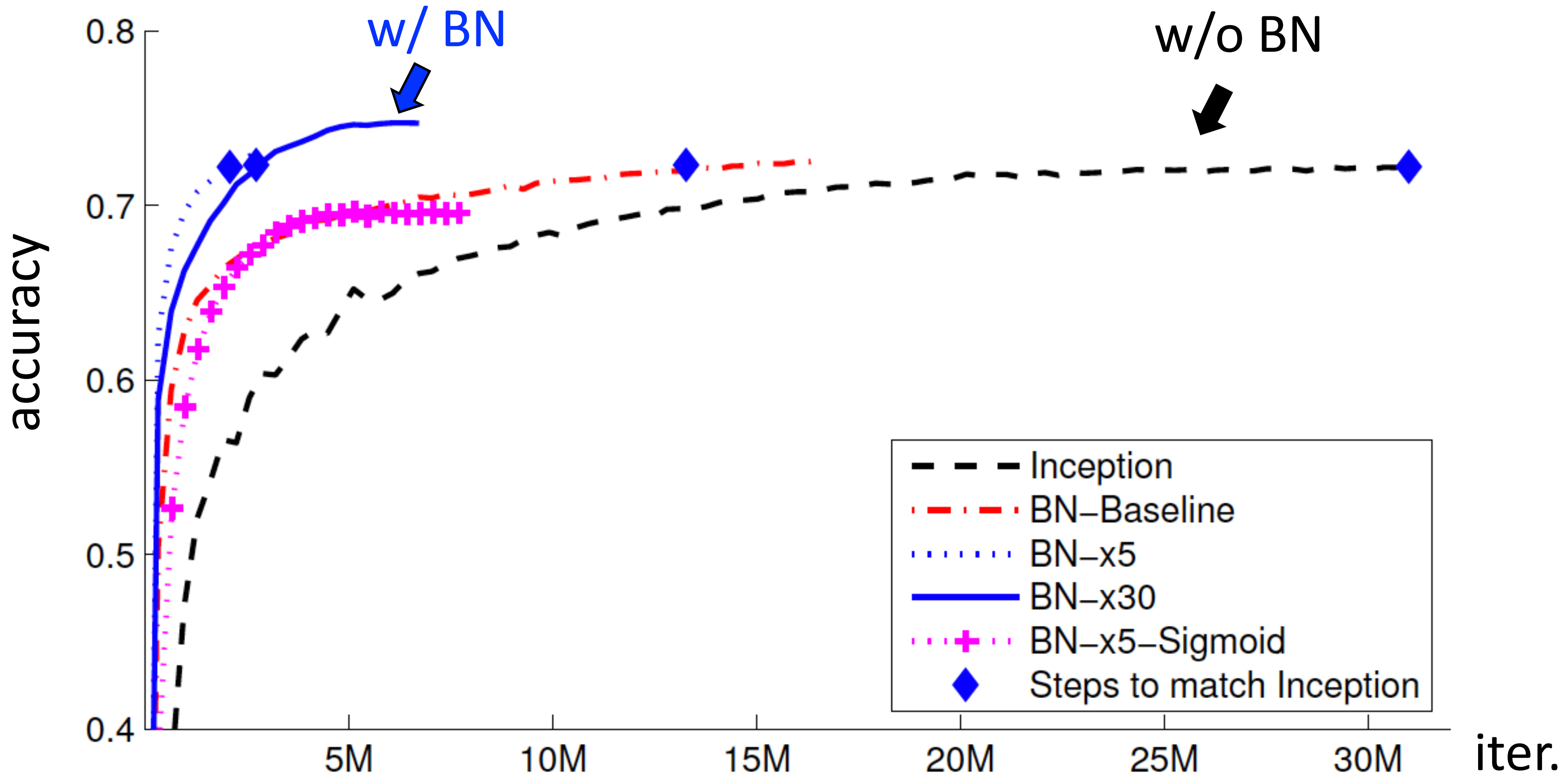


Figure credit: Ioffe & Szegedy

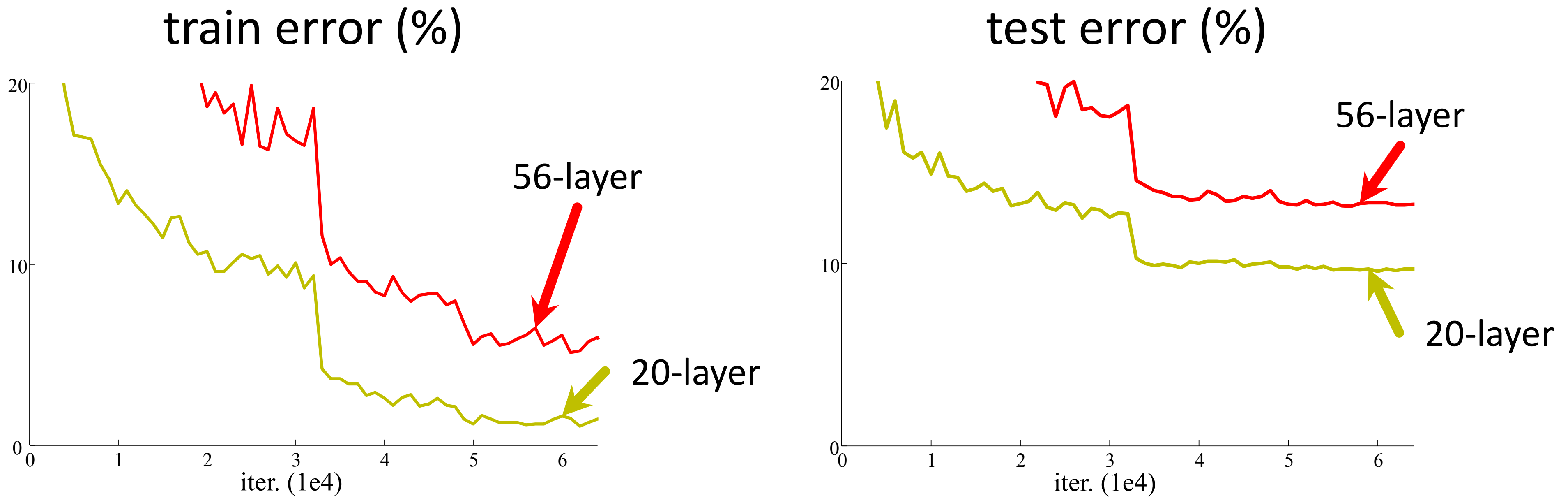
Ioffe & Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. ICML 2015.

ResNets

Credit: ???

Simply stacking layers?

CIFAR-10

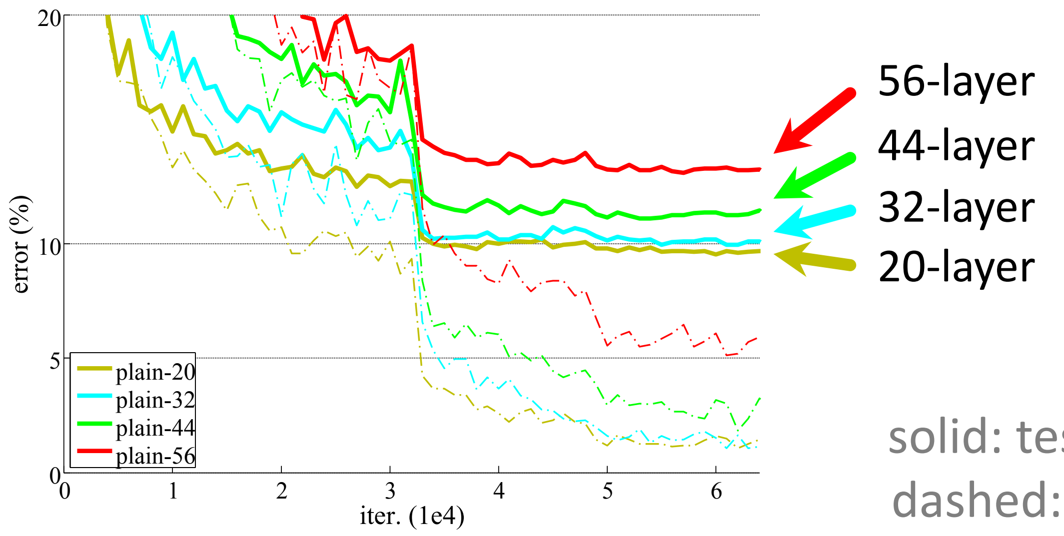


- *Plain* nets: stacking 3x3 conv layers…
- 56-layer net has **higher training error** and test error than 20-layer net

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

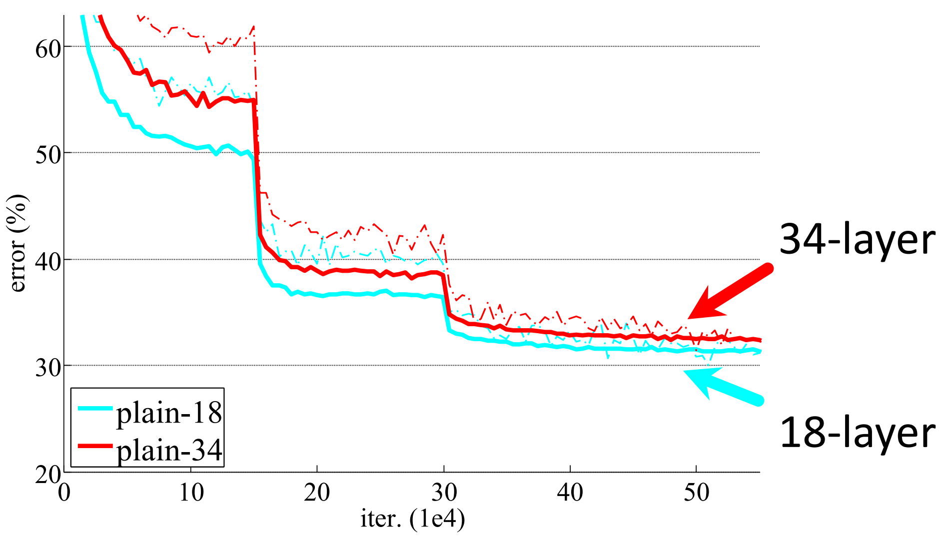
Simply stacking layers?

CIFAR-10



solid: test/val
dashed: train

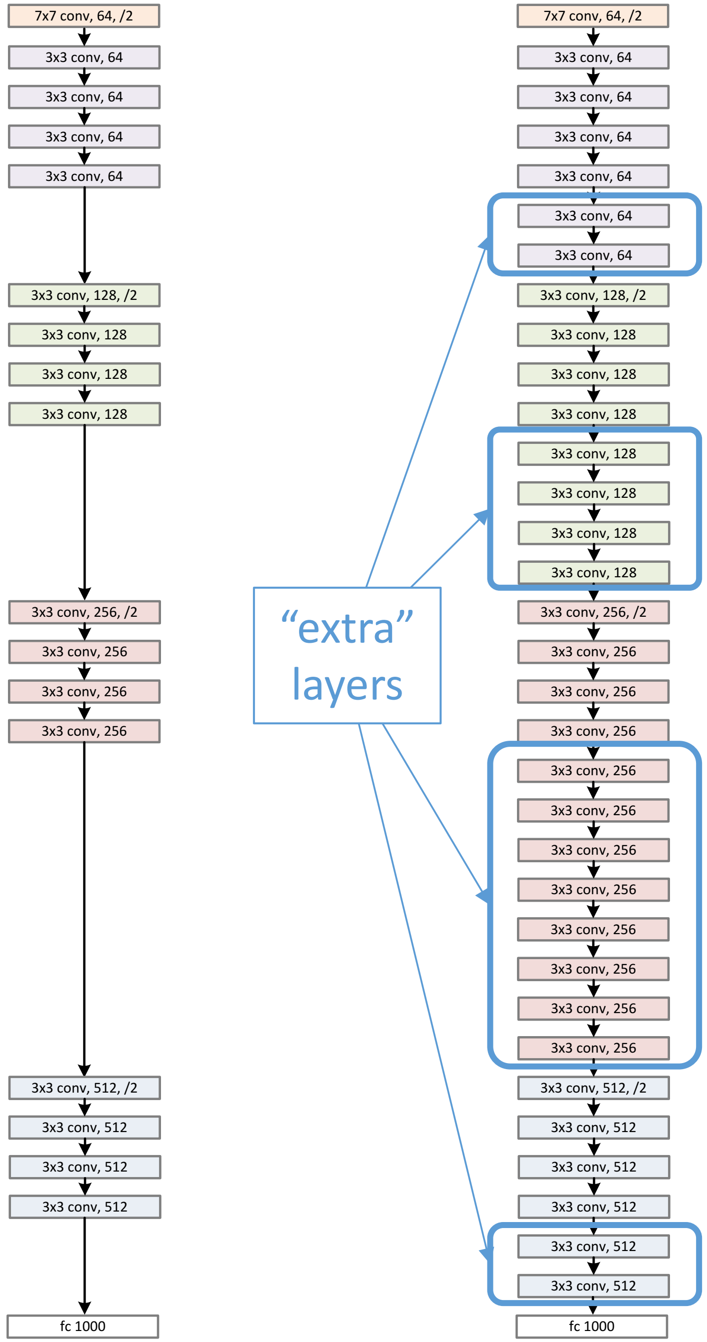
ImageNet-1000



- "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)



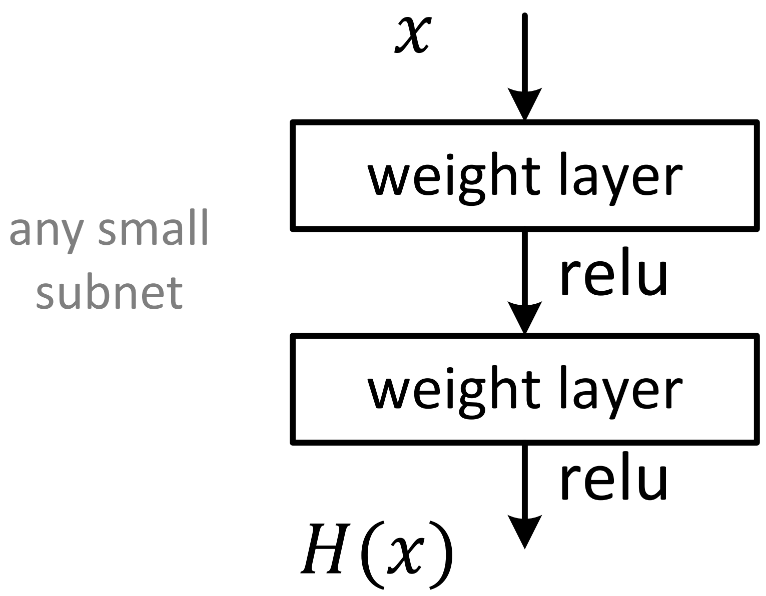
a deeper counterpart (34 layers)

- 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

- Plain net



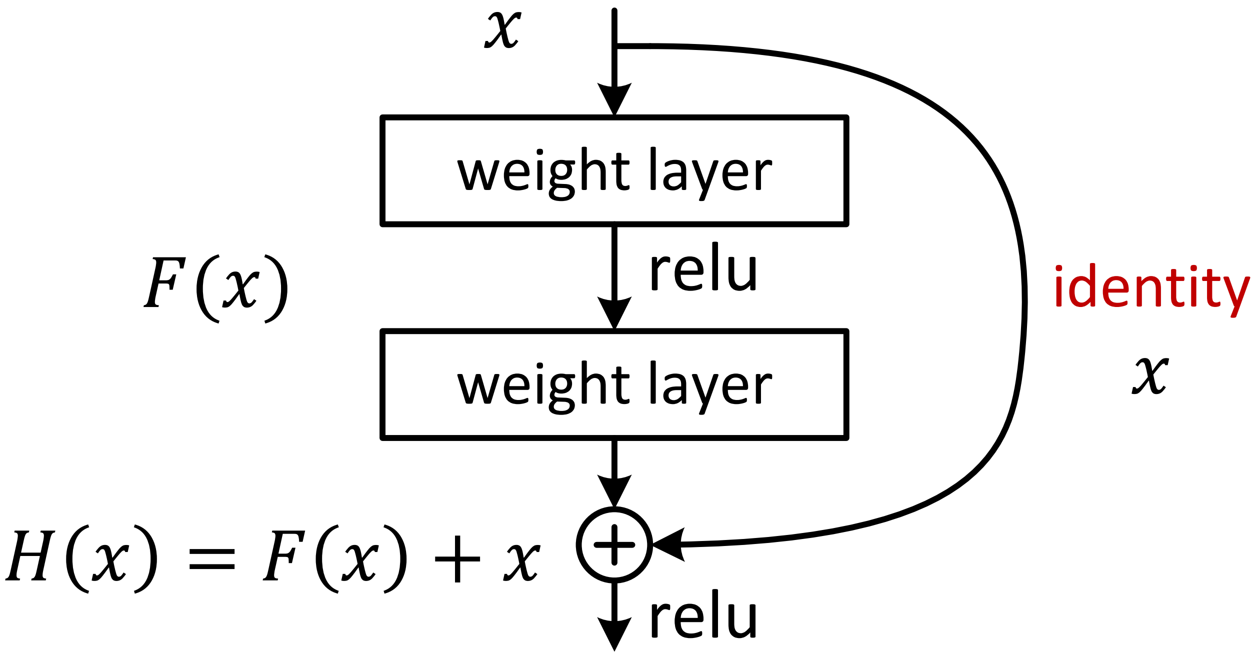
$H(x)$ is any desired mapping,

hope the small subnet 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 small subnet fit $H(x)$~~

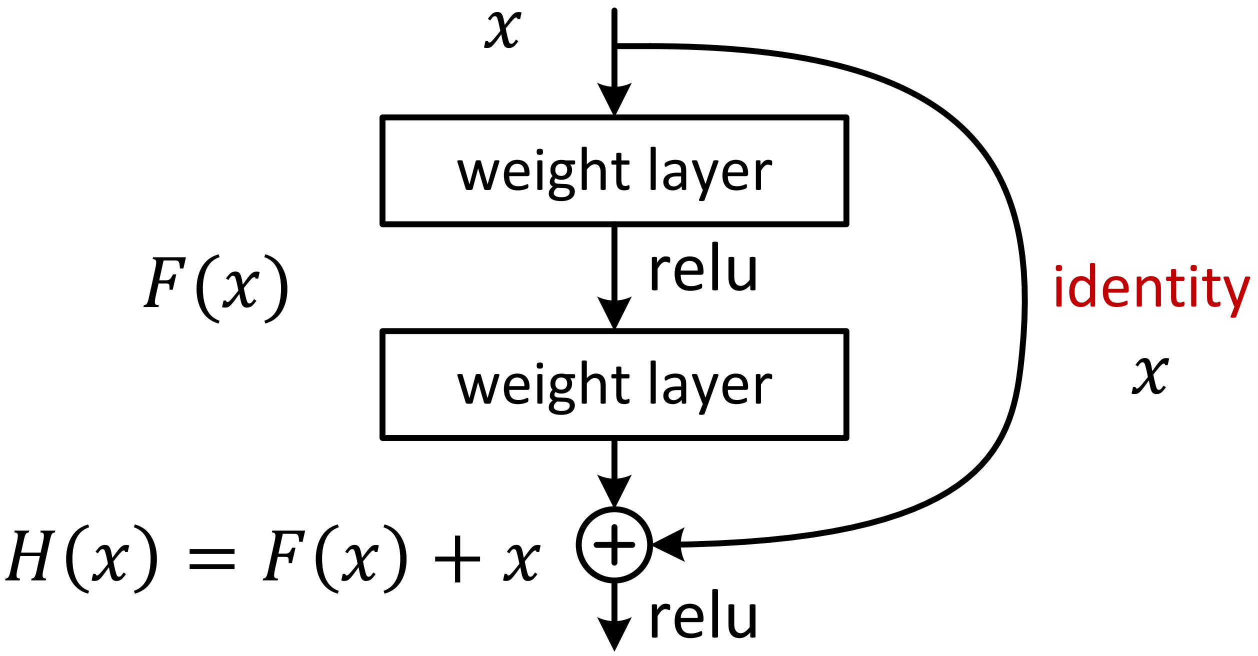
hope the small subnet fit $F(x)$

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

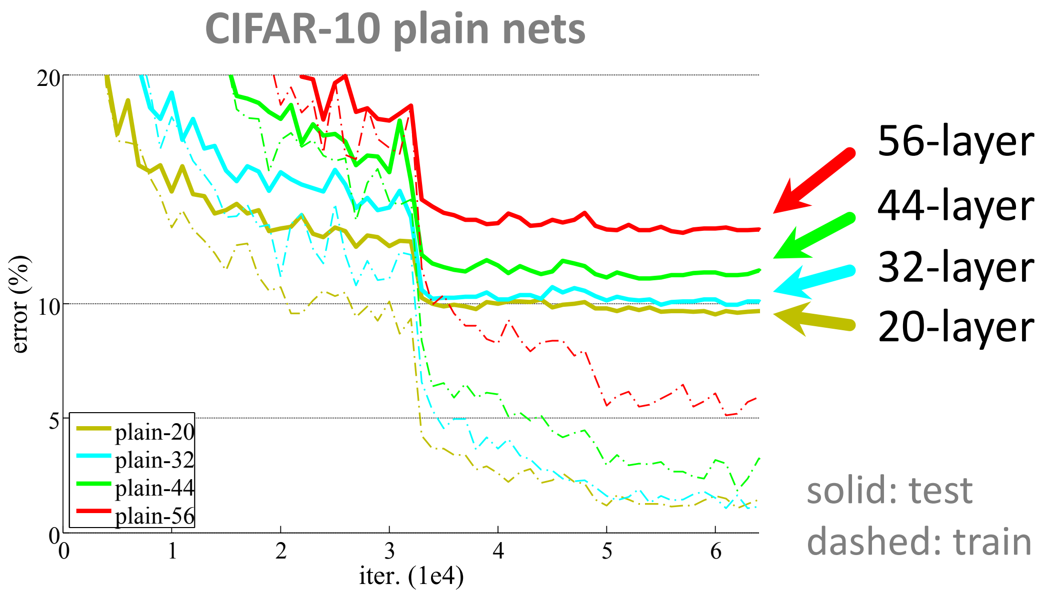
- $F(x)$ is a residual mapping w.r.t. identity

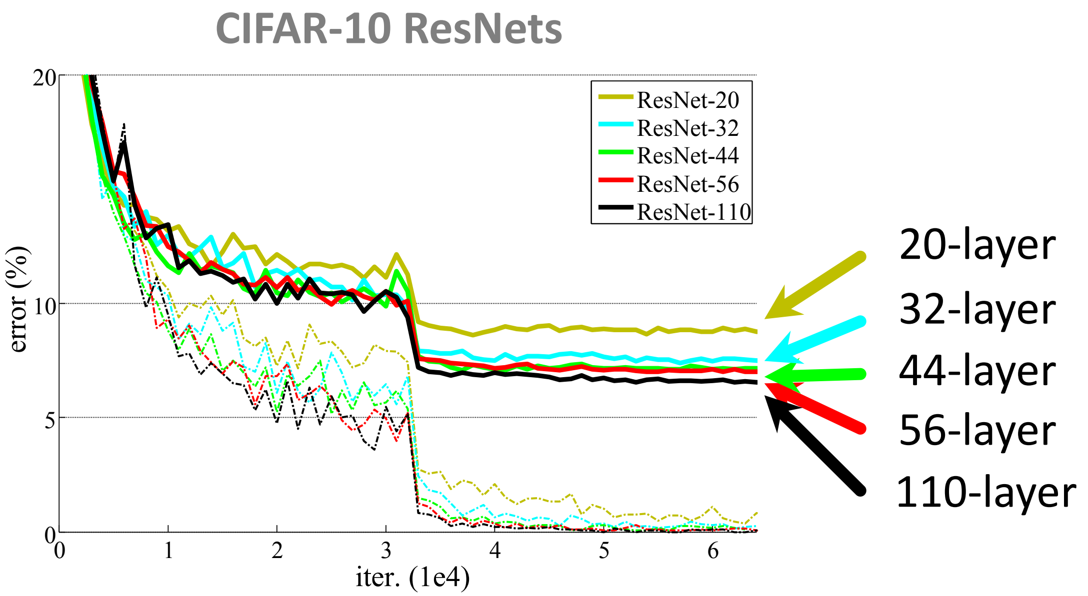


- If identity were optimal, easy to set weights as 0
- If optimal mapping is closer to identity, easier to find small fluctuations

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

CIFAR-10 experiments

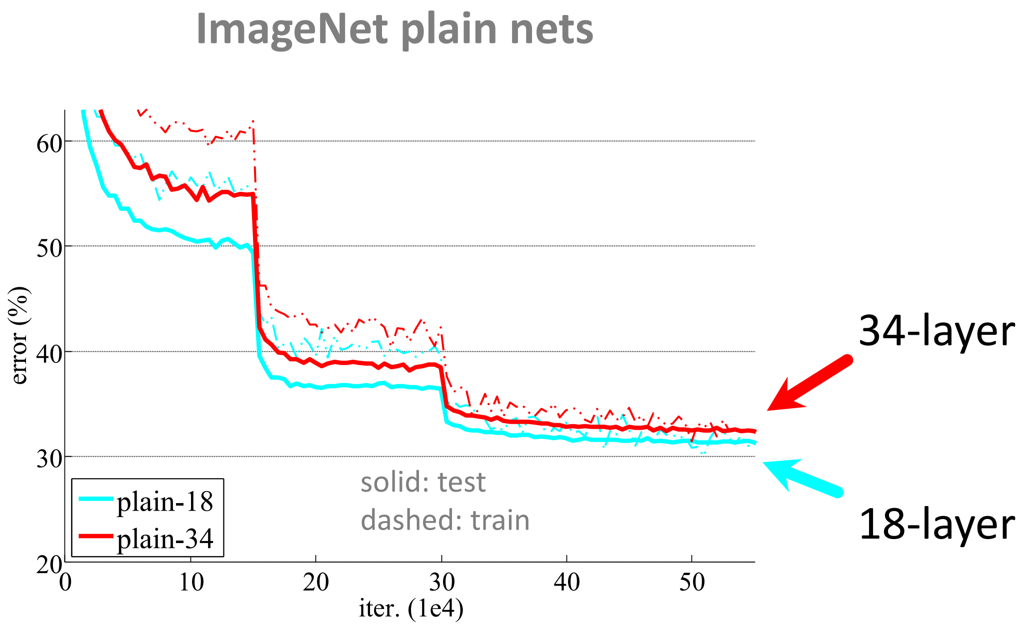


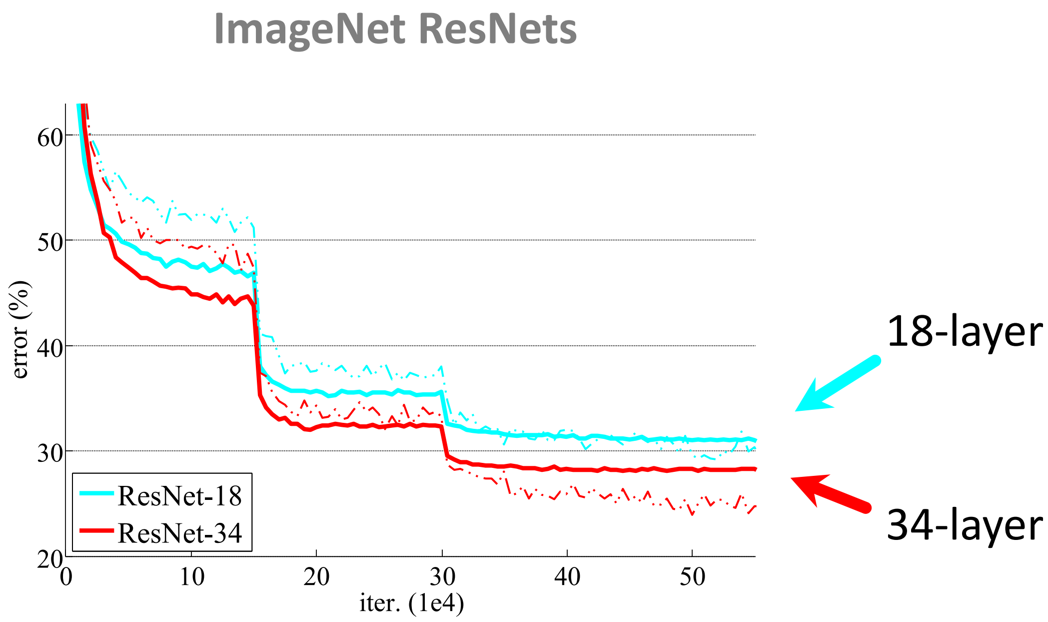


- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

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

ImageNet experiments



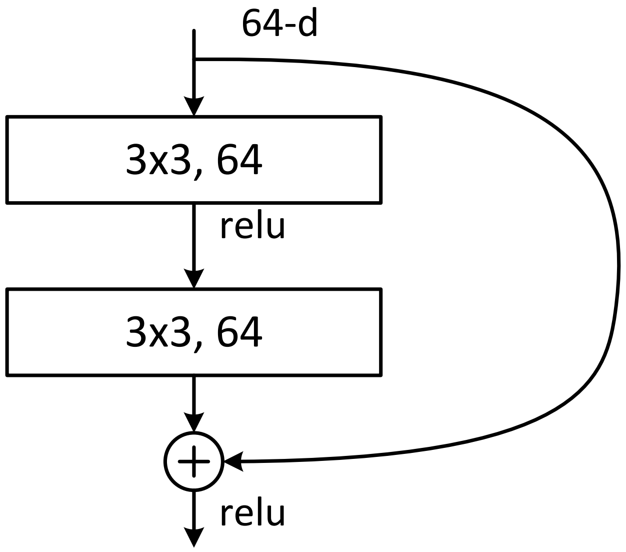


- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

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

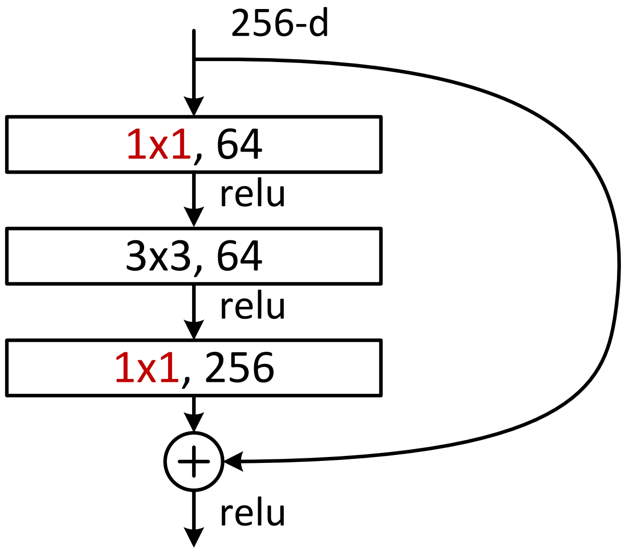
ImageNet experiments

- A practical design of going deeper



all-3x3

similar complexity



bottleneck
(for ResNet-50/101/152)

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

ImageNet experiments

- Deeper ResNets have lower error

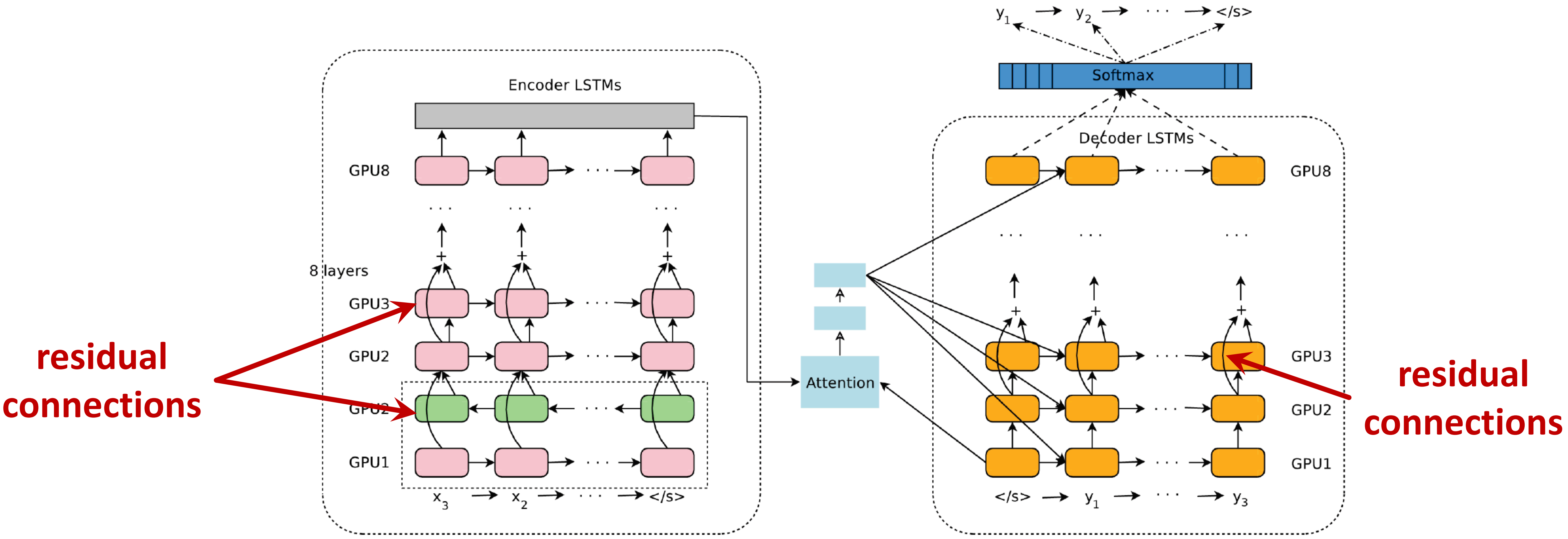
this model has
lower time complexity
than VGG-16/19

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

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

ResNets beyond computer vision

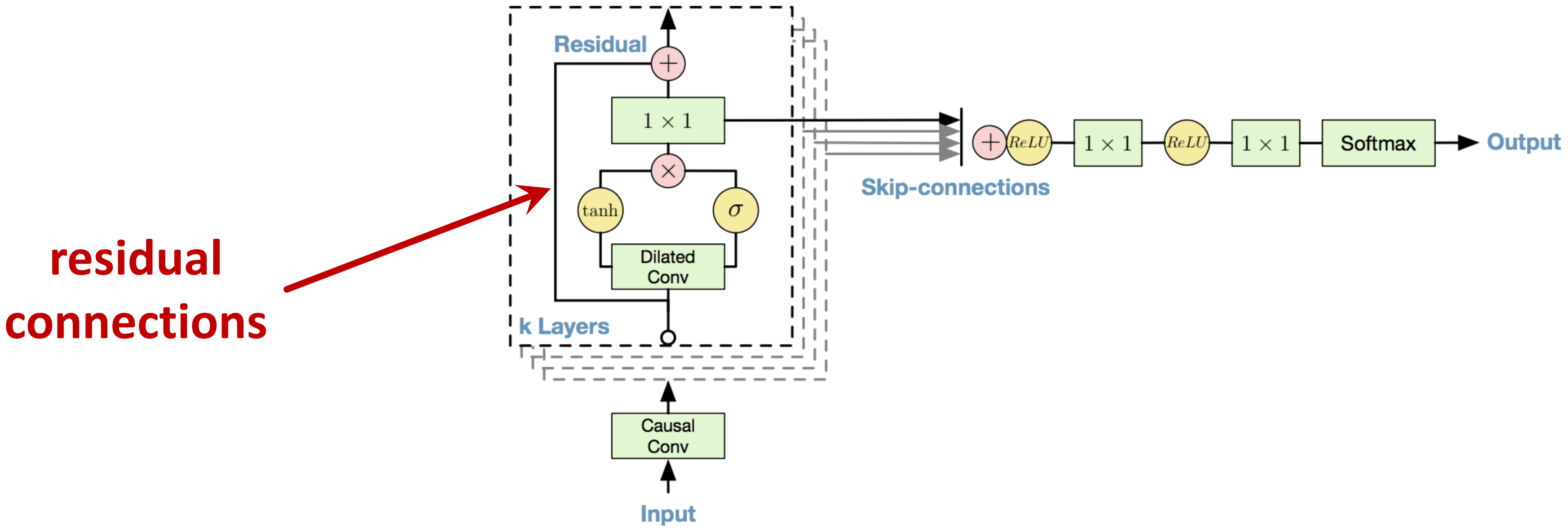
- **Neural Machine Translation** (NMT): 8-layer LSTM!



Wu et al. “Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”. arXiv 2016.

ResNets beyond computer vision

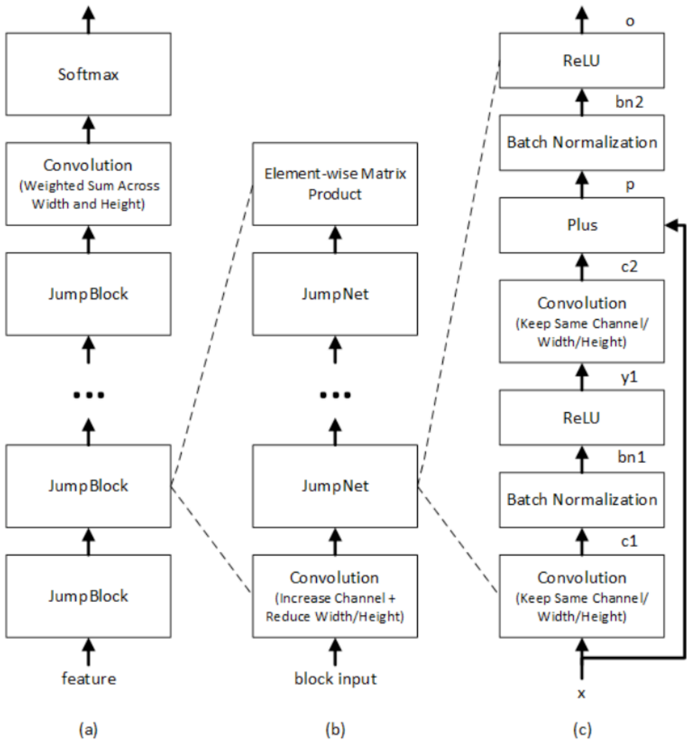
- **Speech Synthesis** (WaveNet): Residual CNNs on 1-d sequence



van den Oord et al. “WaveNet: A Generative Model for Raw Audio”. arXiv 2016.

ResNets beyond computer vision

- **Speech Recognition** – Residual CNNs on 1-d sequence



residual connections

Xiong et al. "The Microsoft 2016 Conversational Speech Recognition System". arXiv 2016.

ResNeXt

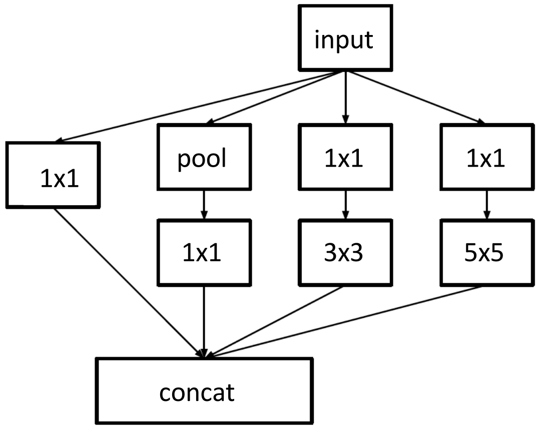
to be presented in CVPR 2017

"Aggregated Residual Transformations for Deep Neural Networks"

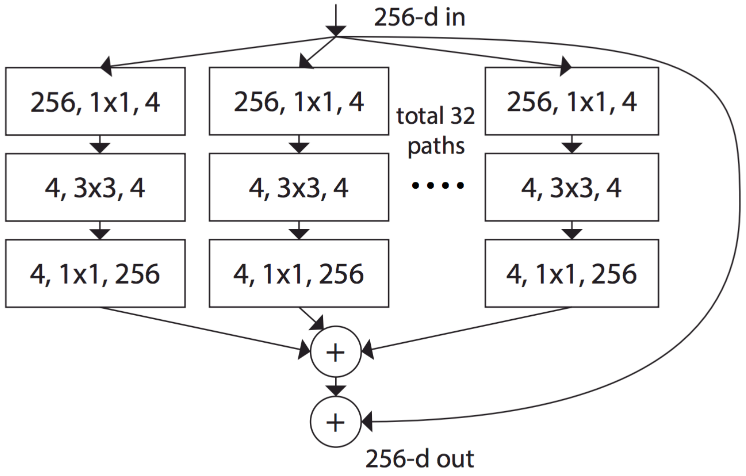
Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He.

Multi-branch

- (Recap): shortcut, bottleneck, and multi-branch



Inception:
heterogeneous multi-branch

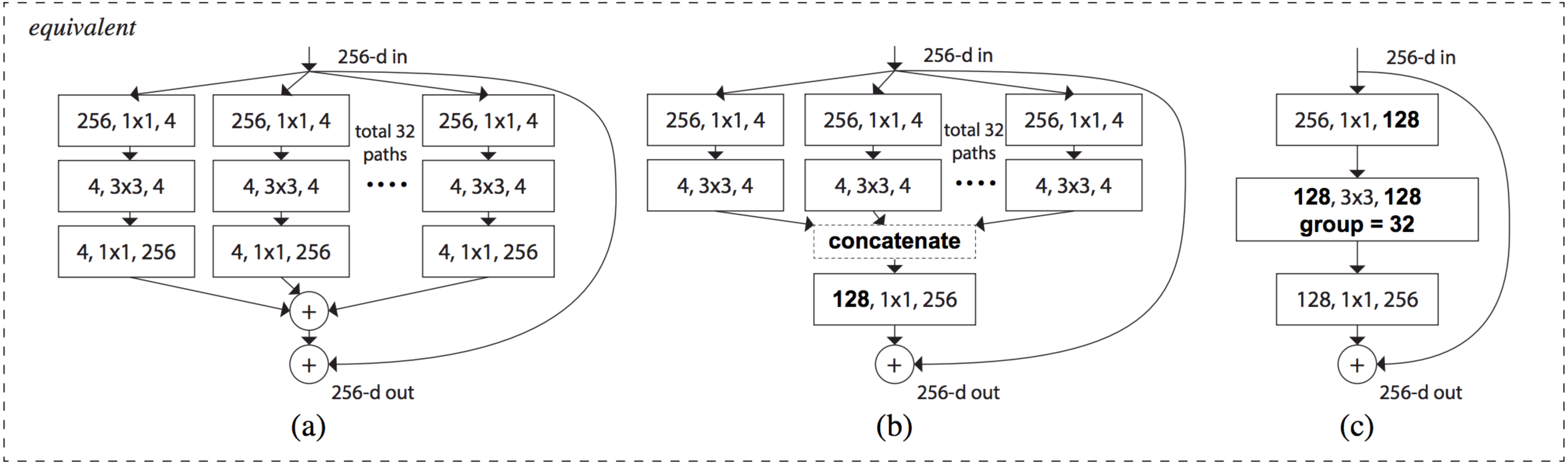


ResNeXt:
uniform multi-branch

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. "Aggregated Residual Transformations for Deep Neural Networks". arXiv 2016 (CVPR 2017).

ResNeXt

- Concatenation and Addition are interchangeable
  - General property for DNNs; not only limited to ResNeXt
- Uniform multi-branching can be done by group-conv

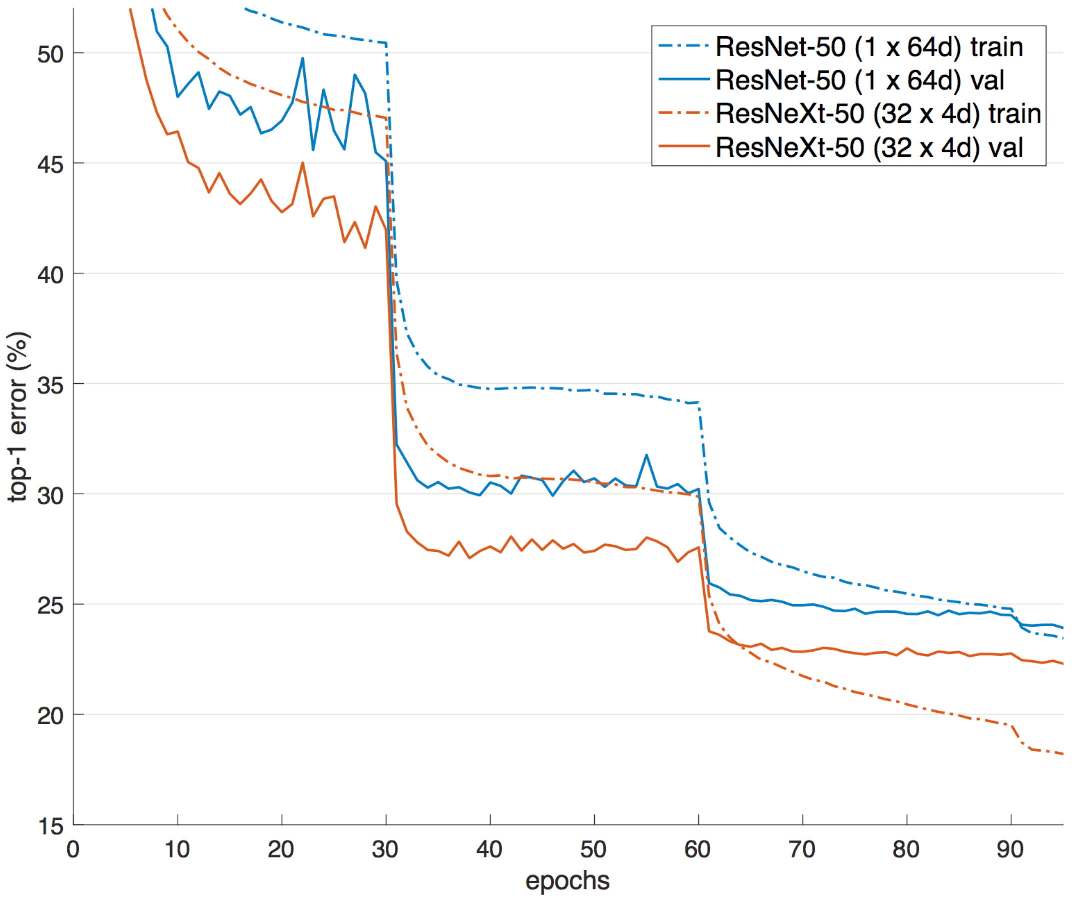


Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. "Aggregated Residual Transformations for Deep Neural Networks". arXiv 2016 (CVPR 2017).

ResNeXt

- Better accuracy
  - when having the same FLOPs/#params as ResNet

- Better trade-off of larger models



Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. "Aggregated Residual Transformations for Deep Neural Networks". arXiv 2016 (CVPR 2017).

ResNeXt for Mask R-CNN

	backbone	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{bb}_{S}	AP^{bb}_{M}	AP^{bb}_{L}
Faster R-CNN+++ [19]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [27]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [21]	Inception-ResNet-v2 [37]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [36]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
Faster R-CNN, RoIAlign	ResNet-101-FPN	37.3	59.6	40.3	19.8	40.2	48.8
Mask R-CNN	ResNet-101-FPN	38.2	60.3	41.7	20.1	41.1	50.2
Mask R-CNN	ResNeXt-101-FPN	**39.8**	**62.3**	**43.4**	**22.1**	**43.2**	51.2

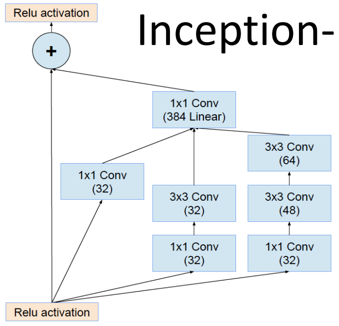
ResNeXt improves 1.6 bbox AP (and 1.4 mask AP) on COCO

Feature still matters!

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask R-CNN". ICCV 2017.
Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. "Aggregated Residual Transformations for Deep Neural Networks". arXiv 2016 (CVPR 2017).

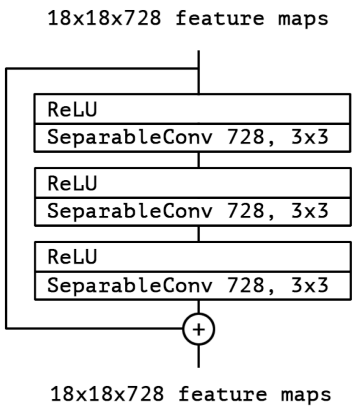
More architectures (not covered in this tutorial)

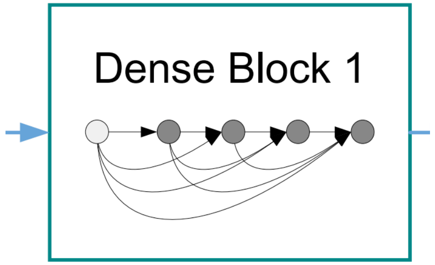
- Inception-ResNet [Szegedy et al 2017]
  - Inception as transformation + residual connection
- DenseNet [Huang et al CVPR 2017]
  - Densely connected shortcuts w/ concat.
- Xception [Chollet CVPR 2017], MobileNets [Howard et al 2017]
  - DepthwiseConv (i.e., GroupConv with #group=#channel)
- ShuffleNet [Zhang et al 2017]
  - More Group/DepthwiseConv + shuffle
- ……



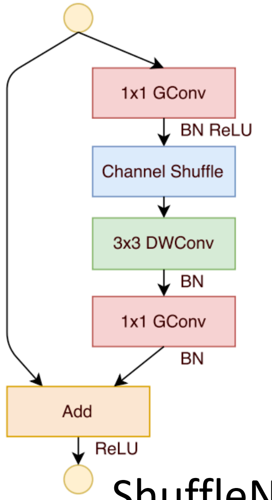
Inception-ResNet

Xception





DenseNet



ShuffleNet

Training ImageNet in 1 Hour

- 256 GPUs
- 8,192 mini-batch size
- ResNet-50
- No loss of accuracy

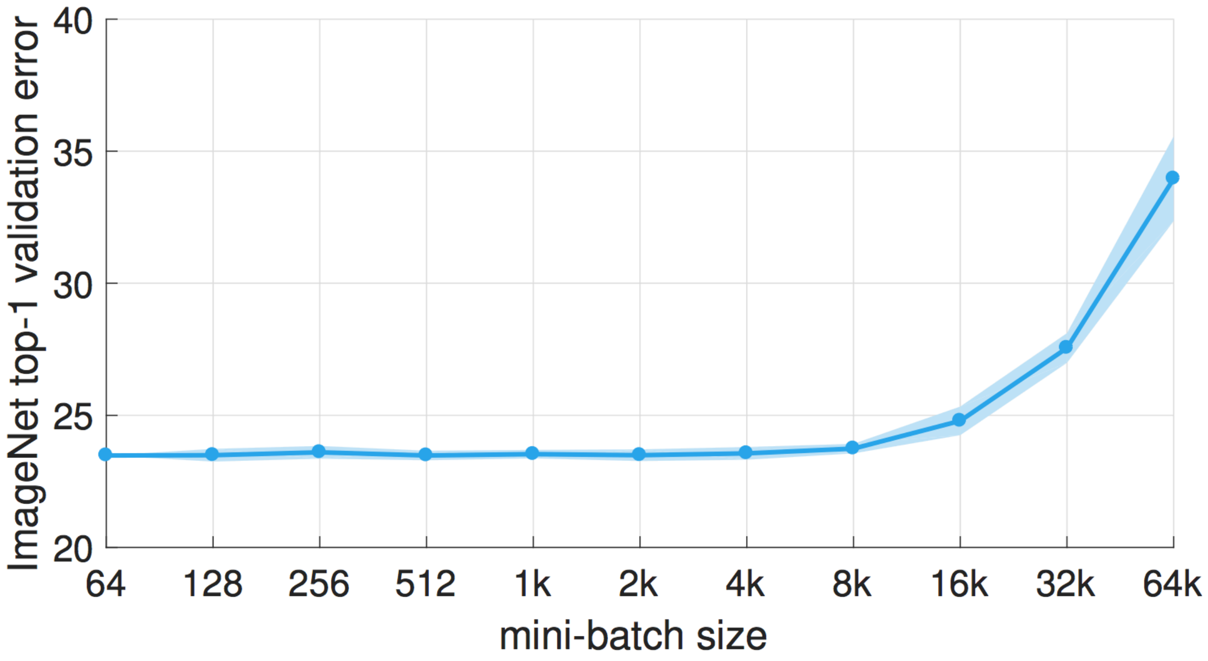


Figure 1. **ImageNet top-1 validation error *vs*. minibatch size.**

Key factors

- Linear scaling learning rate in minibatch size
- Warmup
- Implement things correctly in multiple GPUs/machines!

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, Kaiming He. "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour". arXiv 2017.

Conclusion: Features Matter!

Deep features empower amazing visual recognition results

(Mask R-CNN w/ ResNet101; more in next talk)

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. “Mask R-CNN”. ICCV 2017.