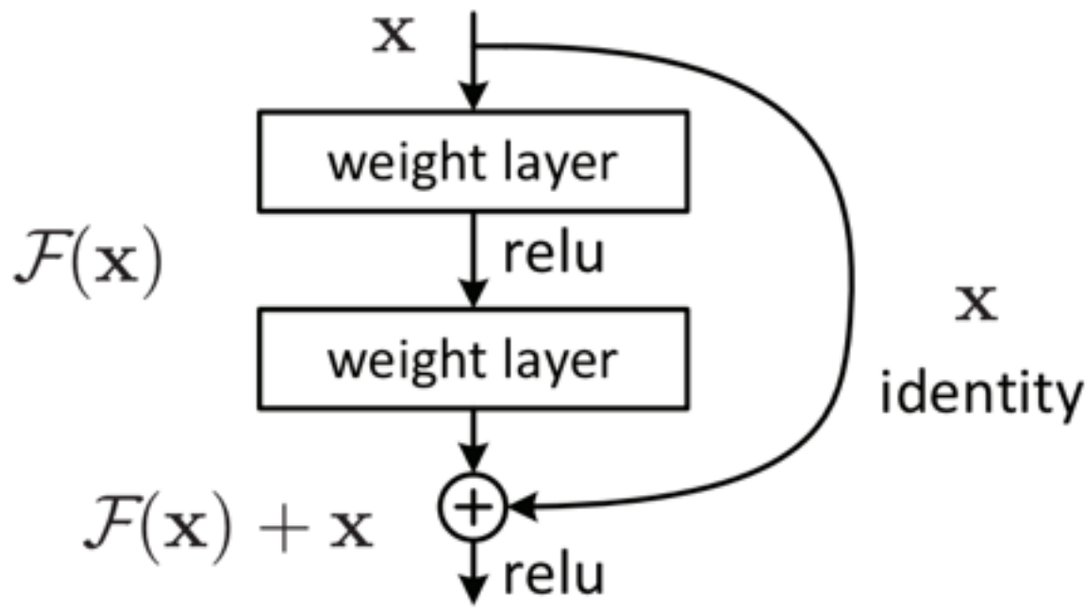
Deep Residual Learning for Image Recognition

K. He, X. Zhang, S. Ren and J. Sun
Microsoft Research

ILSVRC 2015
MS COCO 2015
WINNER

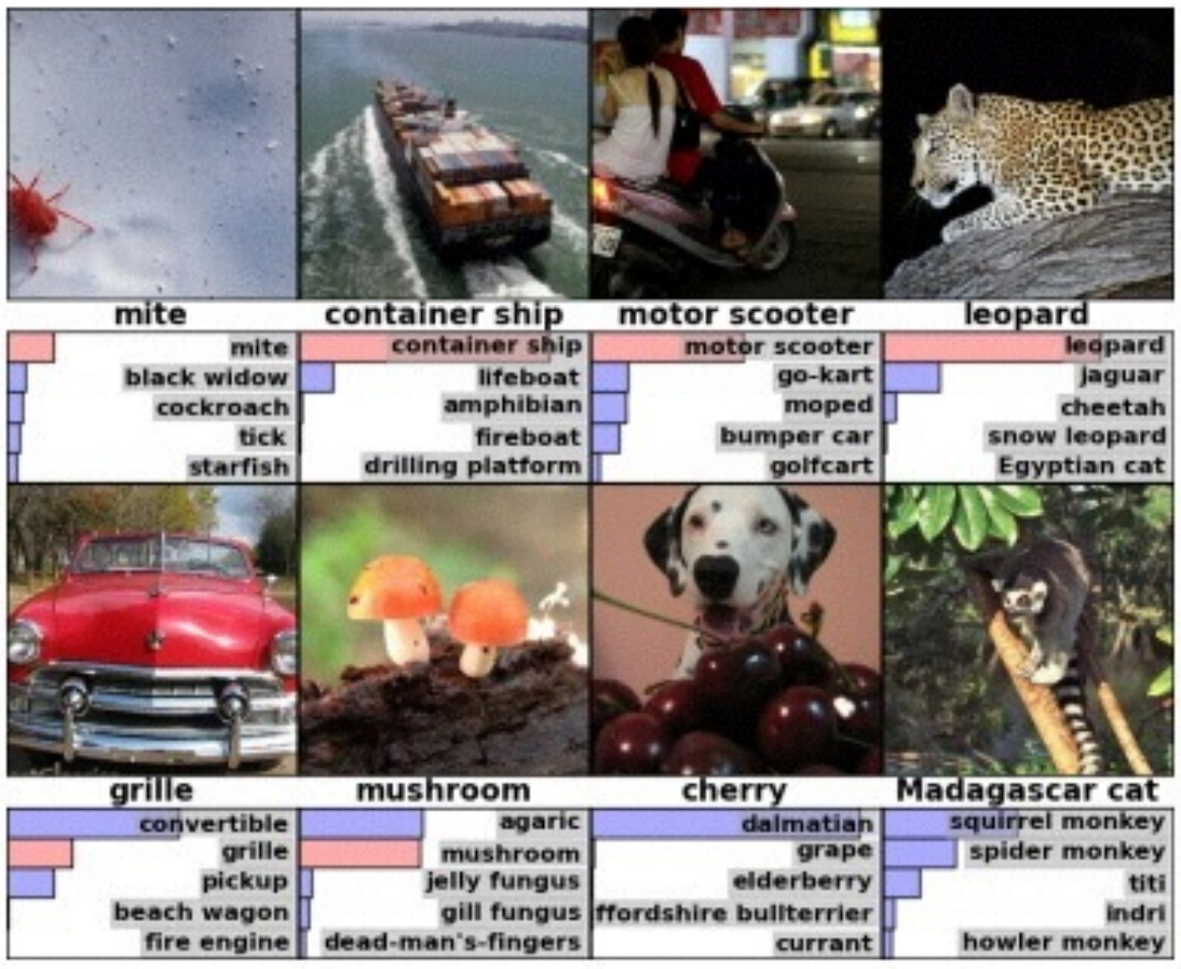
Article overview by
Ilya Kuzovkin

Computational Neuroscience Seminar
University of Tartu
2016


x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

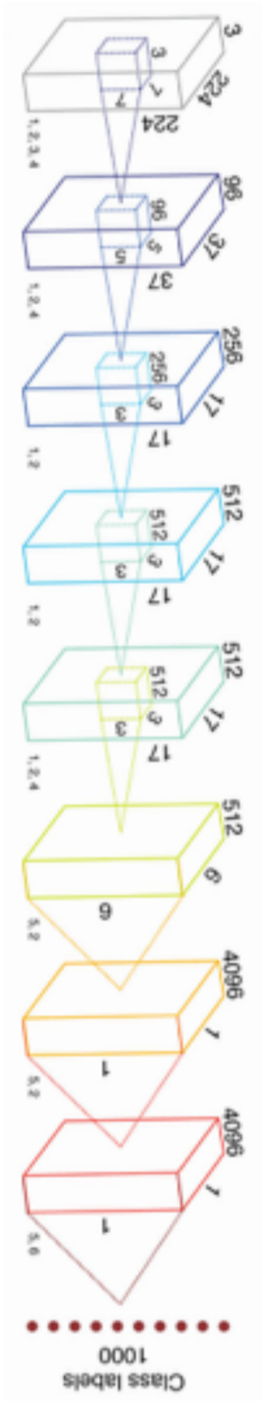
The Idea

IMAGENET



1000 classes

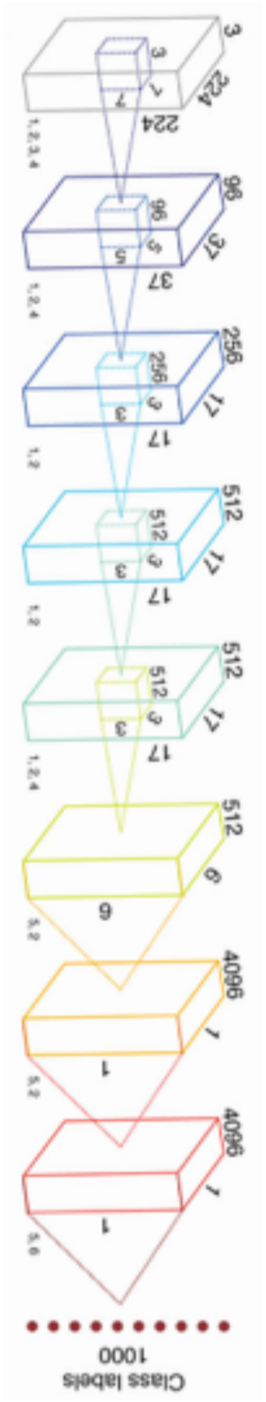
2012



8 layers

15.31% error

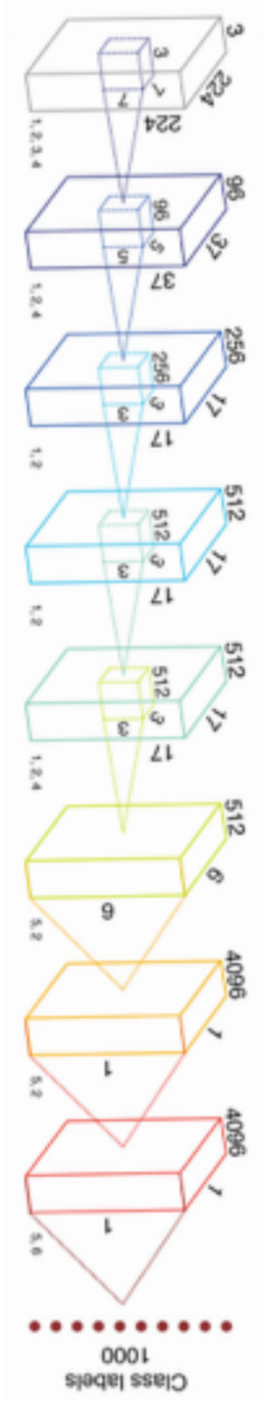
2012



8 layers

15.31% error

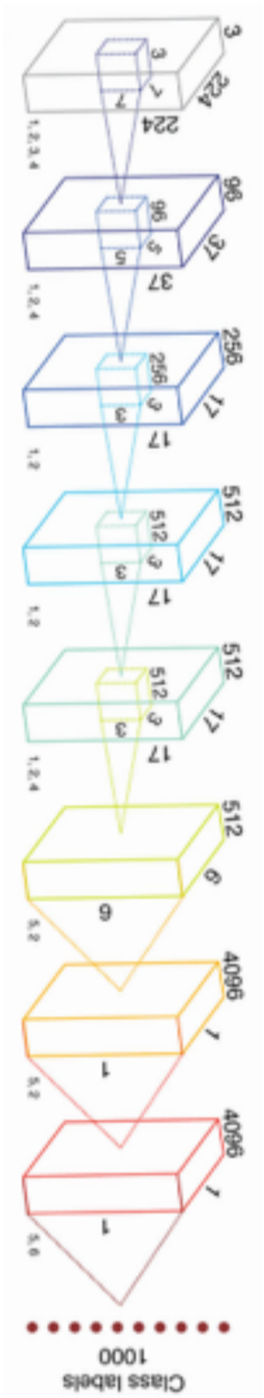
2013



9 layers, 2x params

11.74% error

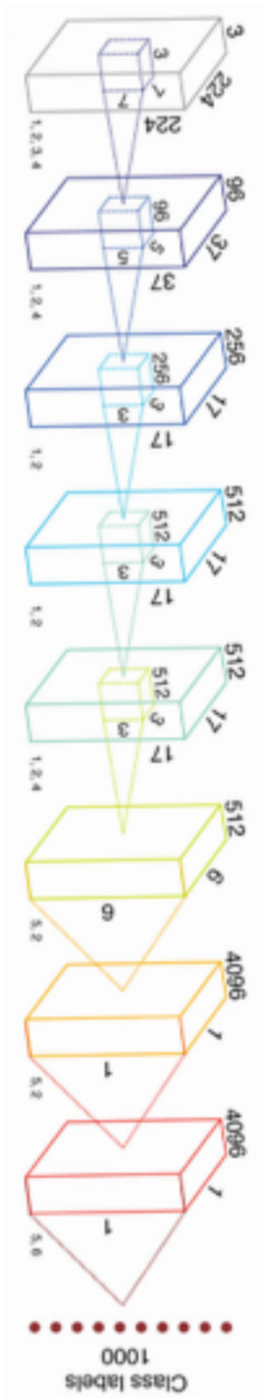
2012



8 layers

15.31% error

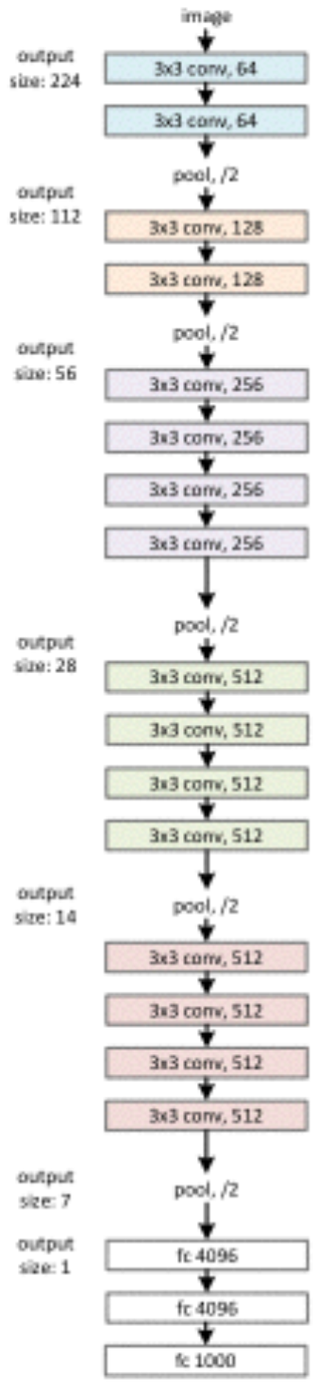
2013



9 layers, 2x params

11.74% error

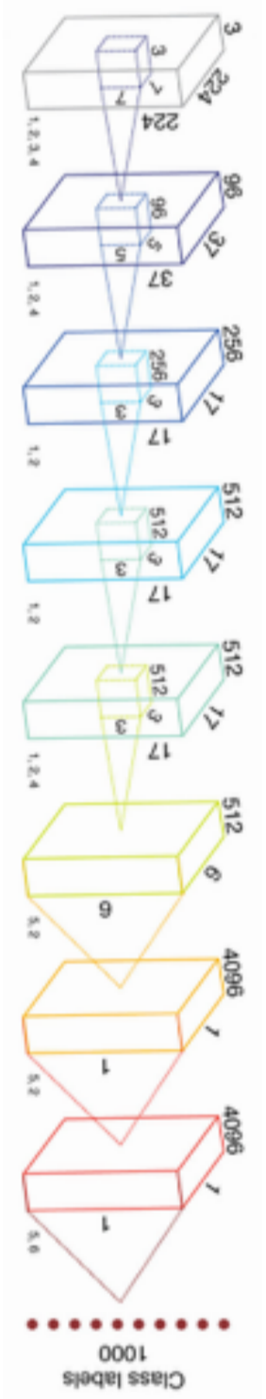
2014



19 layers

7.41% error

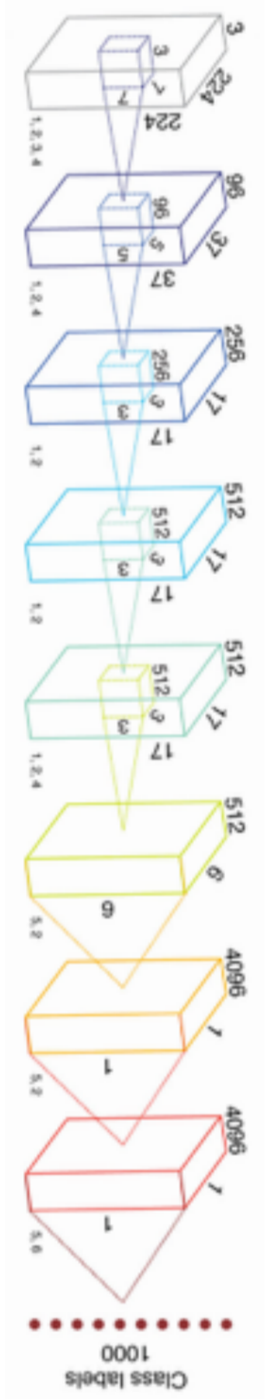
2012



8 layers

15.31% error

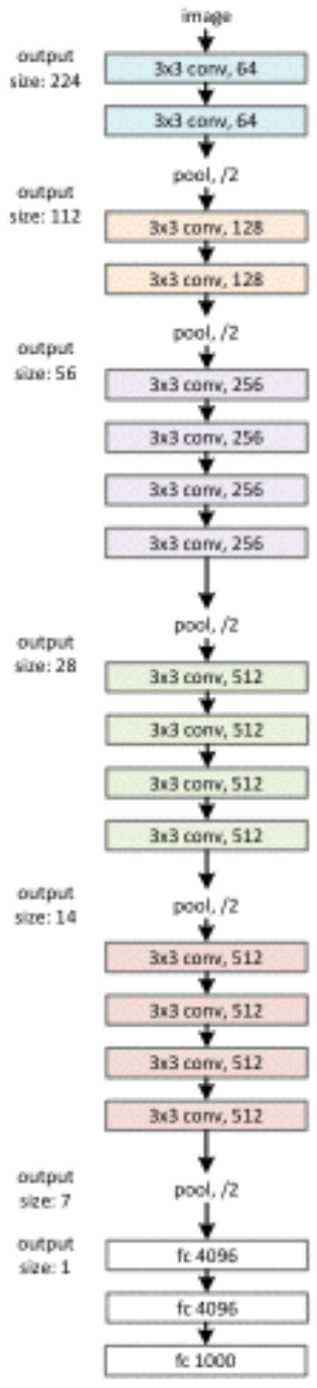
2013



9 layers, 2x params

11.74% error

2014



19 layers

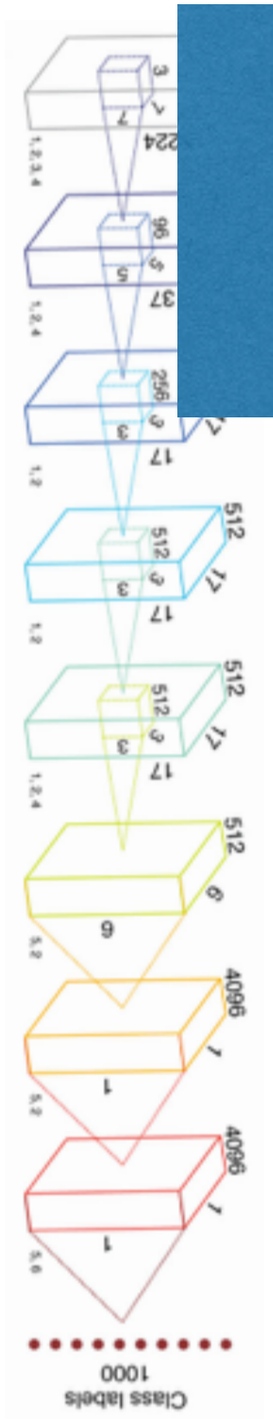
7.41% error

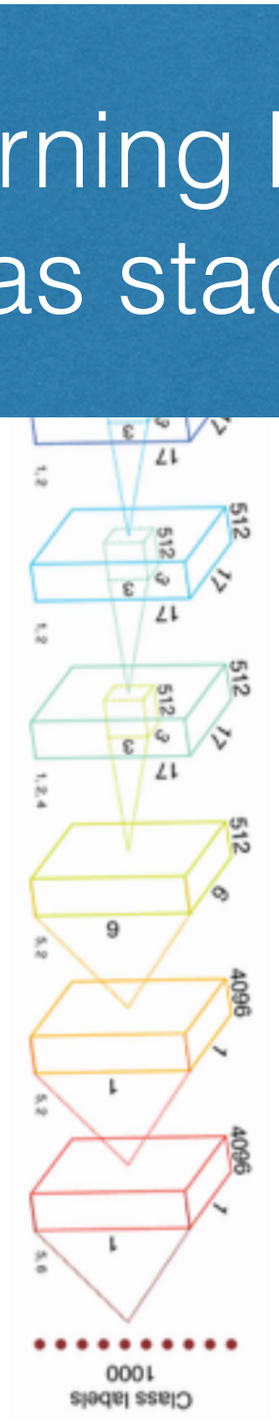
2015

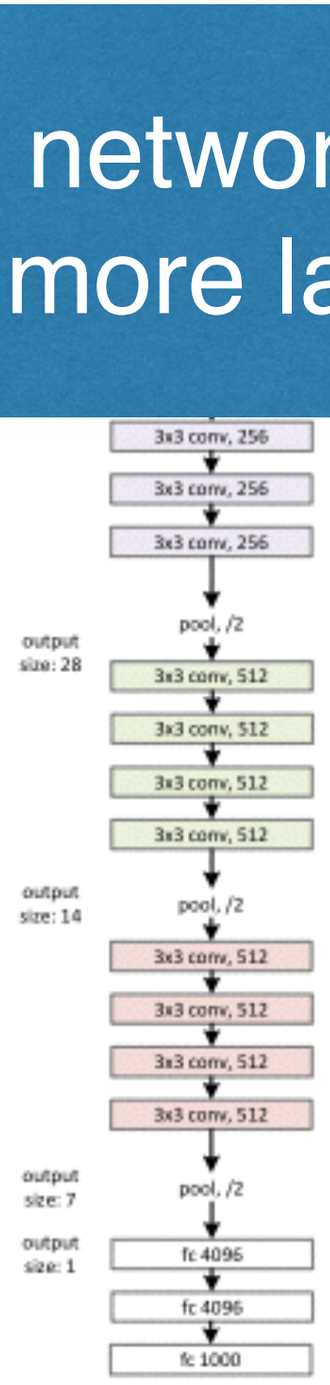
?

2012 2013 2014 2015

Is learning **better networks** as easy as stacking **more layers**?







?

8 layers
15.31% error

9 layers, 2x params
11.74% error

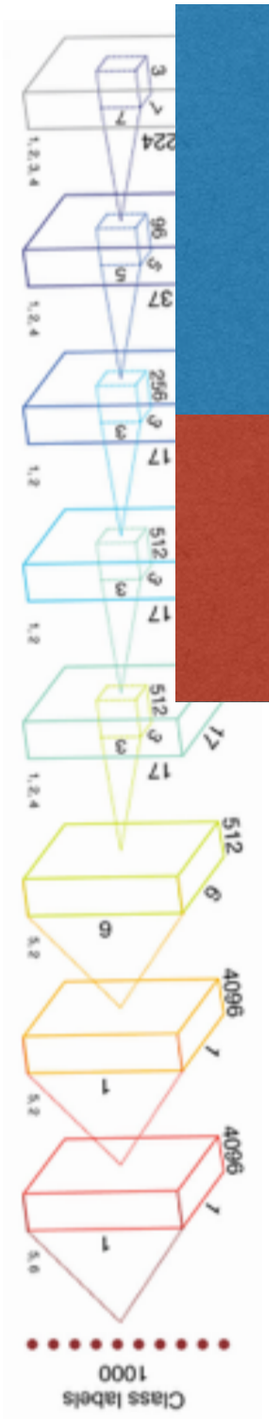
19 layers
7.41% error

2012 2013 2014 2015

Is learning **better networks** as easy as stacking **more layers**?

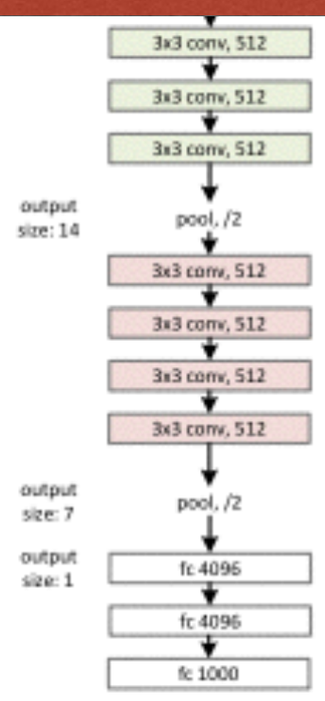
Vanishing / exploding gradients

?



8 layers
15.31% error

9 layers, 2x params
11.74% error



19 layers
7.41% error

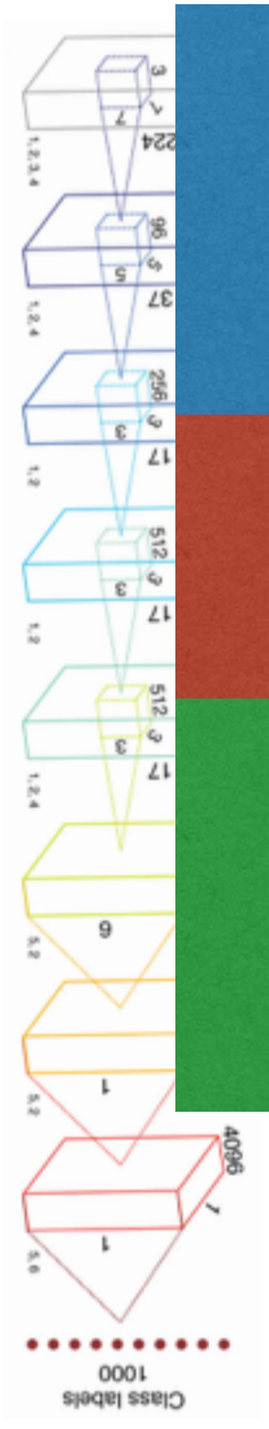
2012 2013 2014 2015

Is learning **better networks** as easy as stacking **more layers**?

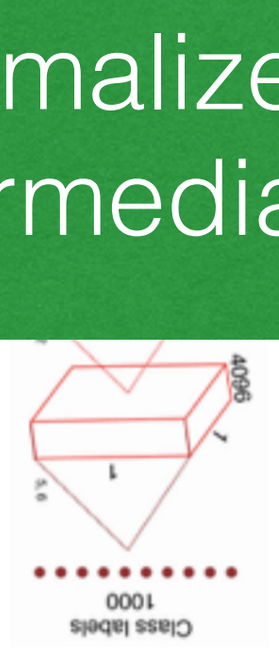
Vanishing / exploding gradients

Normalized initialization & intermediate normalization

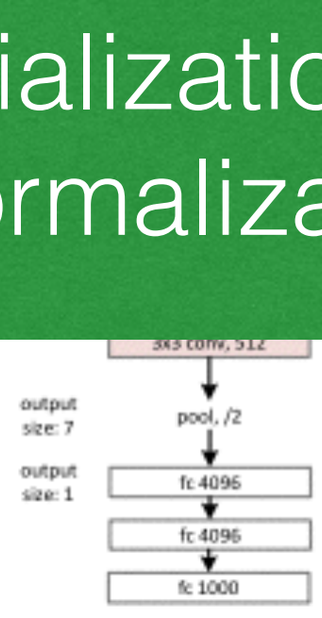
?



8 layers
15.31% error

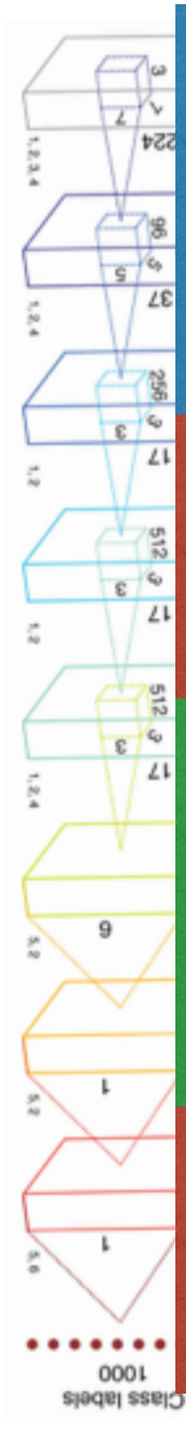


9 layers, 2x params
11.74% error



19 layers
7.41% error

2012 2013 2014 2015



Is learning **better networks** as easy as stacking **more layers**?

Vanishing / exploding gradients

Normalized initialization & intermediate normalization

Degradation problem

?

8 layers
15.31% error

9 layers, 2x params
11.74% error

19 layers
7.41% error

Degradation problem

"with the network depth increasing, accuracy gets saturated"

Degradation problem

"with the network depth increasing, accuracy gets saturated"

Not caused by overfitting:

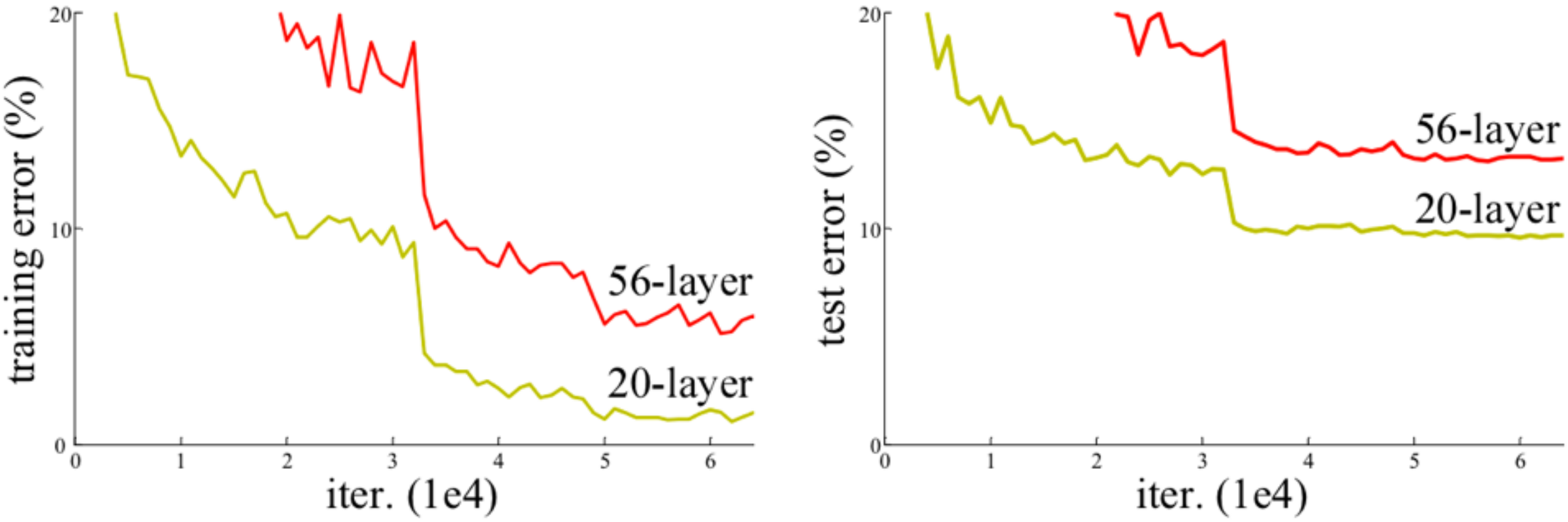
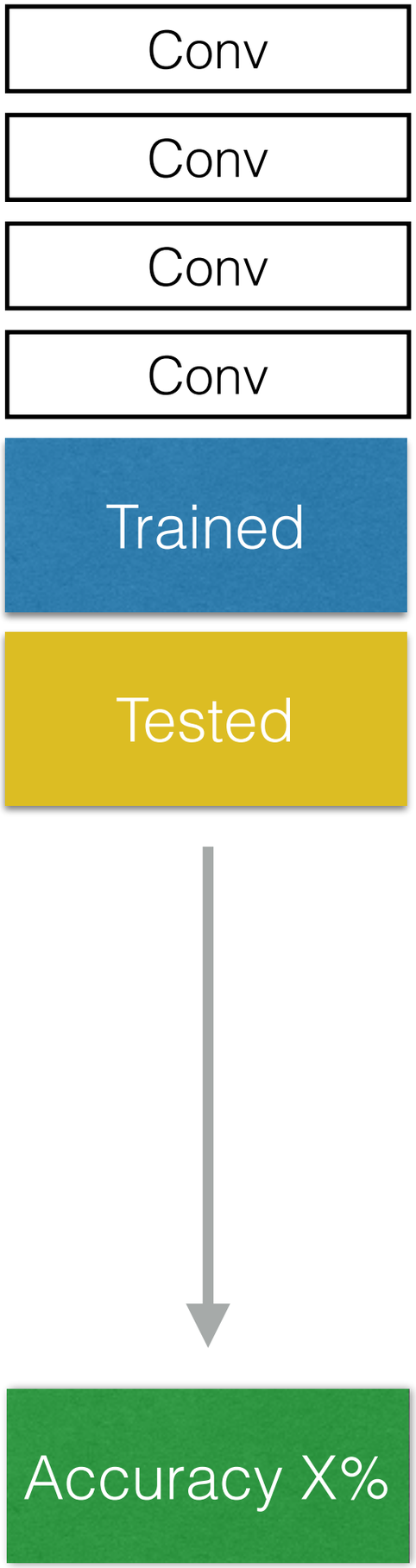


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error.


Conv
Conv
Conv
Conv
Trained
Tested
Accuracy X%

Conv

Conv

Conv

Conv

Trained

Tested

Accuracy X%

Conv

Conv

Conv

Conv

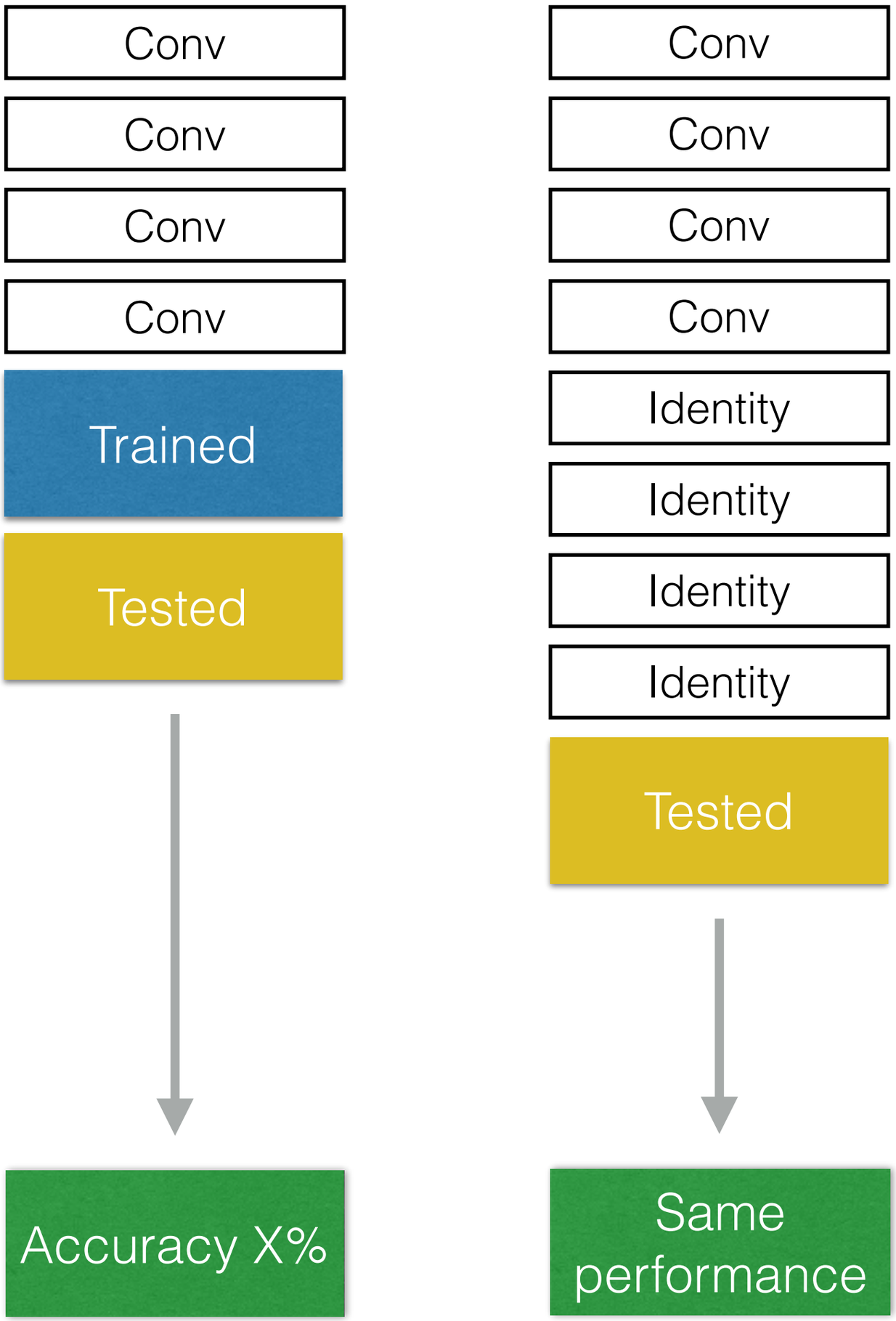
Identity

Identity

Identity

Identity

Tested


Conv
Conv
Conv
Conv
Trained
Tested
Accuracy X%
Conv
Conv
Conv
Conv
Identity
Identity
Identity
Identity
Tested
Same performance

Conv

Conv

Conv

Conv

Trained

Tested

Accuracy X%

Conv

Conv

Conv

Conv

Identity

Identity

Identity

Identity

Tested

Same performance

Conv

Conv

Conv

Conv

Conv

Conv

Conv

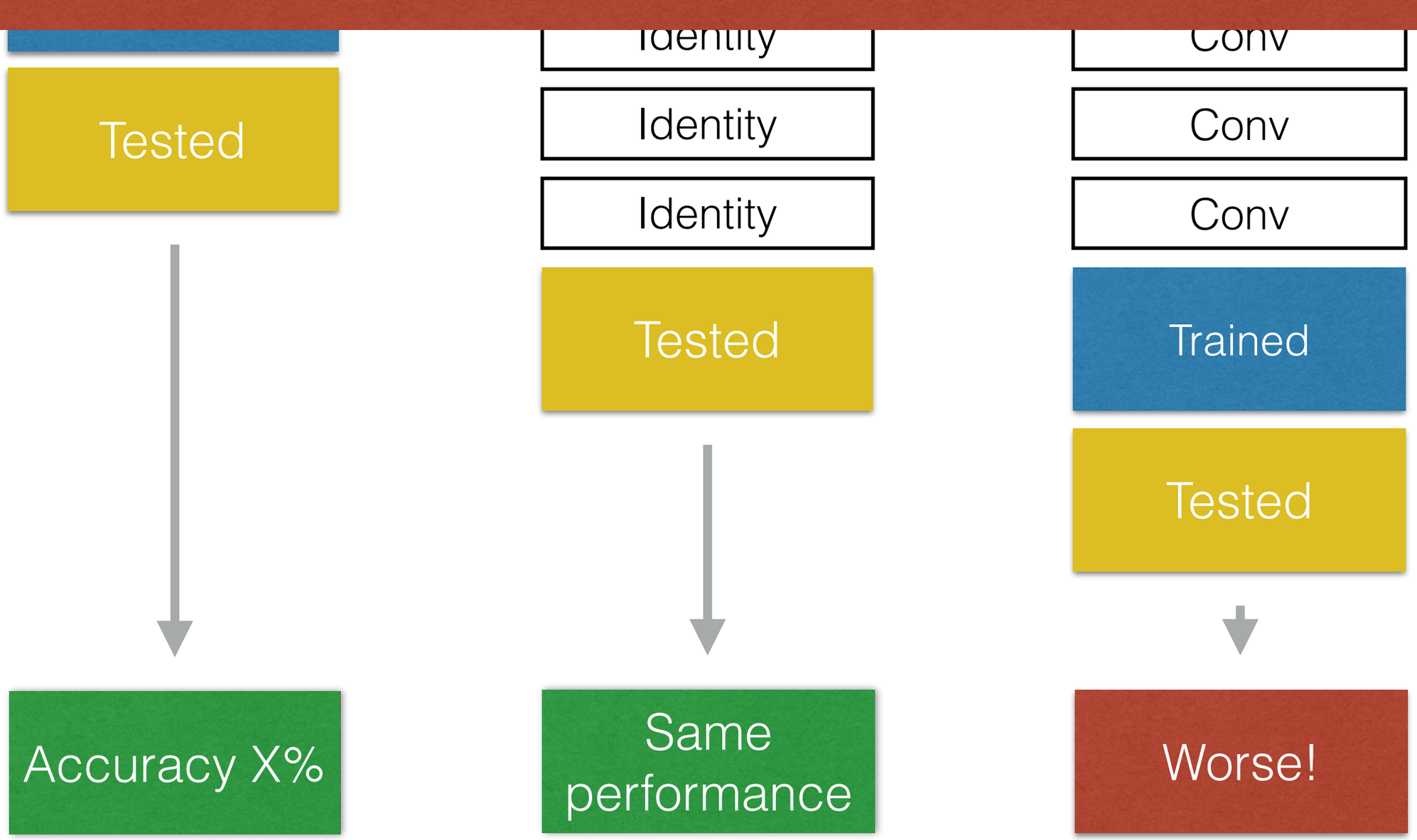
Conv

Trained

Tested

Conv	Conv	Conv
Conv	Conv	Conv
Conv	Conv	Conv
Conv	Conv	Conv
Trained	Identity	Conv
Tested	Identity	Conv
	Identity	Conv
	Identity	Conv
	Tested	Trained
		Tested
Accuracy X%	Same performance	Worse!

*"Our **current solvers** on hand are **unable to find solutions** that are comparably good or better than the **constructed solution** (or unable to do so in feasible time)"*



*"Our **current solvers** on hand are **unable to find solutions** that are comparably good or better than the **constructed solution** (or unable to do so in feasible time)"*

*"Solvers might have **difficulties in approximating identity mappings** by multiple nonlinear layers"*

Accuracy X%

Same performance

Tested

Worse!

*"Our **current solvers** on hand are **unable to find solutions** that are comparably good or better than the **constructed solution** (or unable to do so in feasible time)"*

*"Solvers might have **difficulties in approximating identity mappings** by multiple nonlinear layers"*

Add explicit **identity connections** and *"solvers may simply drive the **weights** of the multiple nonlinear layers **toward zero**"*

Add explicit **identity connections** and *"solvers may simply drive the **weights** of the multiple nonlinear layers **toward zero**"*

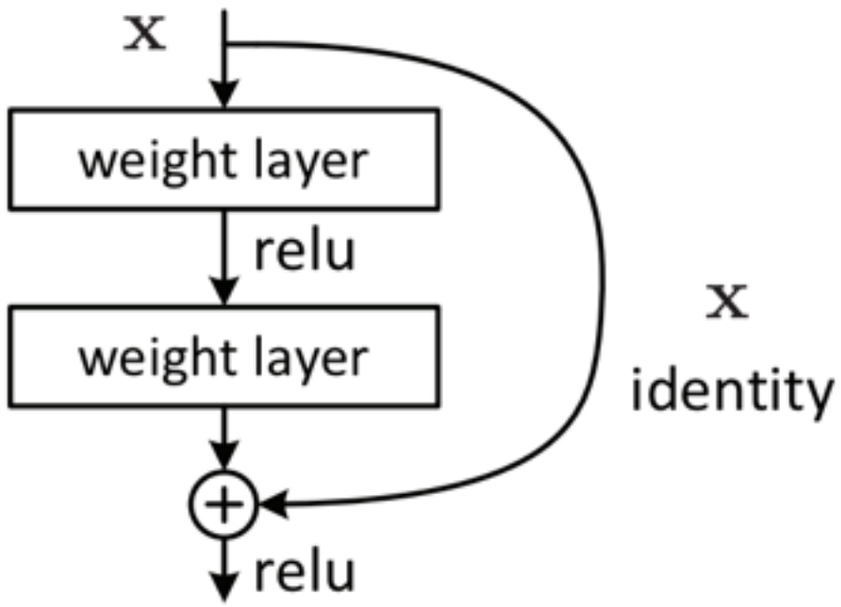


Figure 2. Residual learning: a building block.

$\mathcal{H}(\mathbf{x})$ is the true function we want to learn

Add explicit **identity connections** and *"solvers may simply drive the **weights** of the multiple nonlinear layers **toward zero**"*

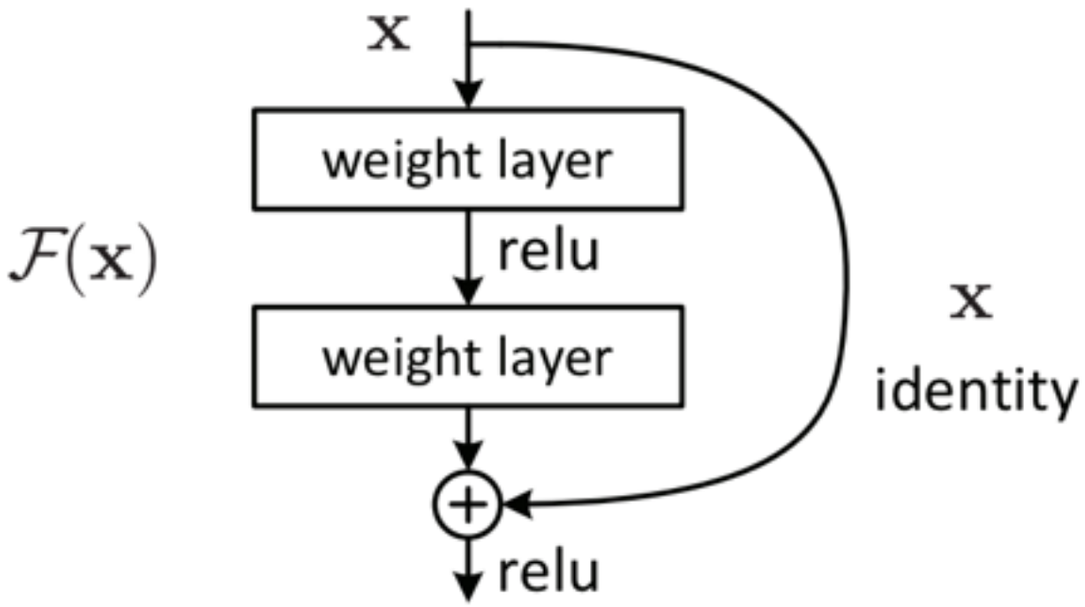


Figure 2. Residual learning: a building block.

$\mathcal{H}(\mathbf{x})$ is the true function we want to learn

Let's pretend we want to learn

$$\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$$

instead.

Add explicit **identity connections** and *"solvers may simply drive the **weights** of the multiple nonlinear layers **toward zero**"*

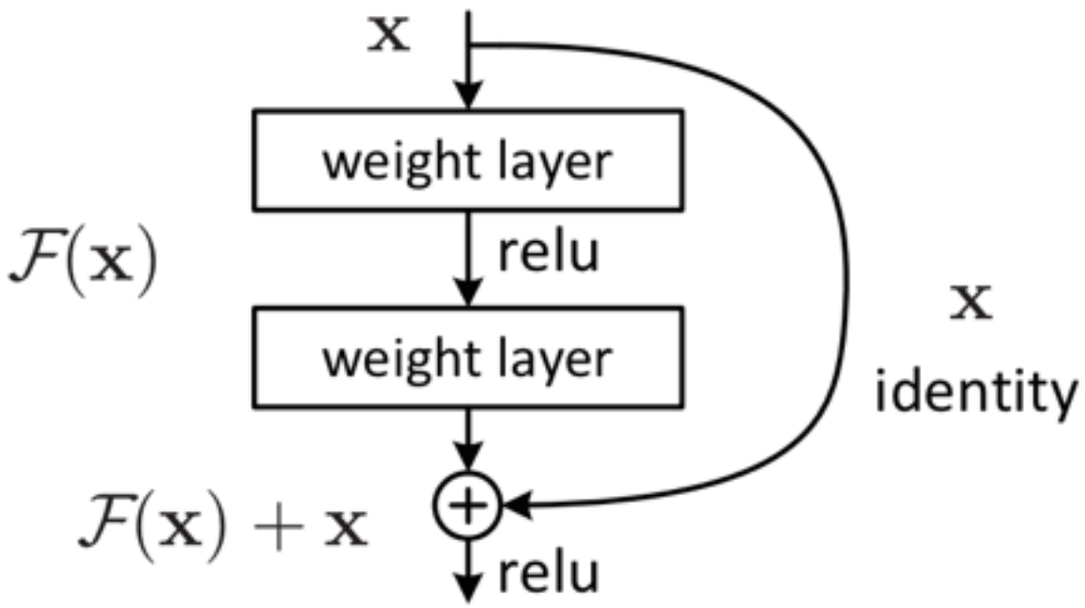


Figure 2. Residual learning: a building block.

$\mathcal{H}(\mathbf{x})$ is the true function we want to learn

Let's pretend we want to learn

$$\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$$

instead.

The original function is then

$$\mathcal{F}(\mathbf{x}) + \mathbf{x}$$

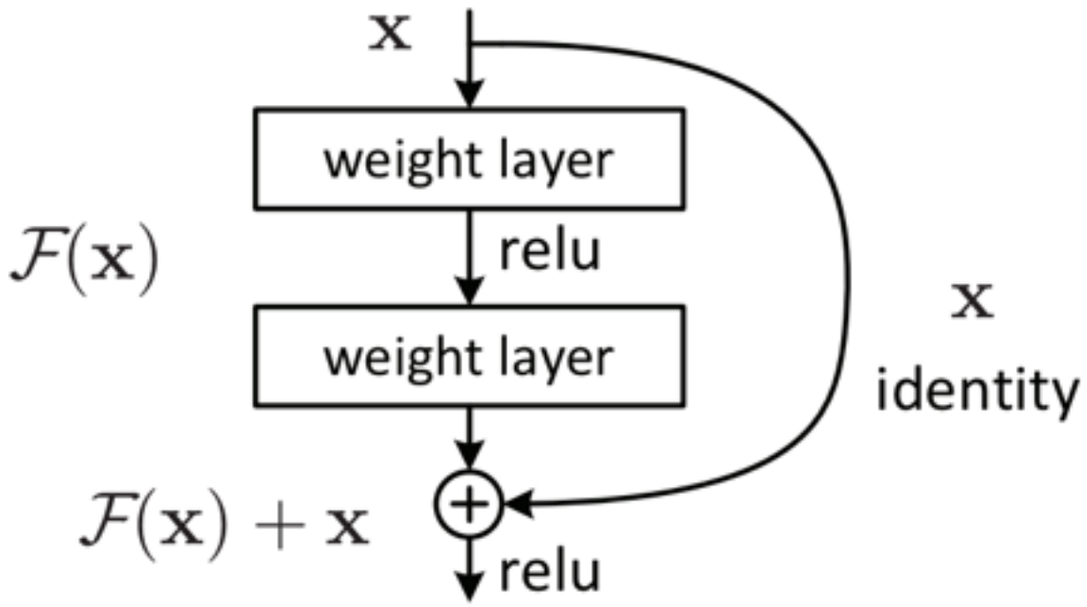
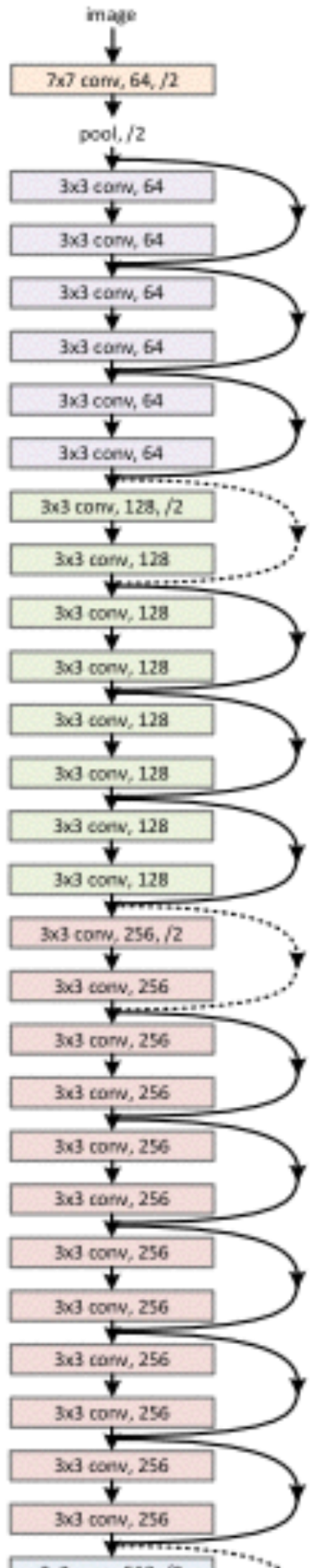


Figure 2. Residual learning: a building block.



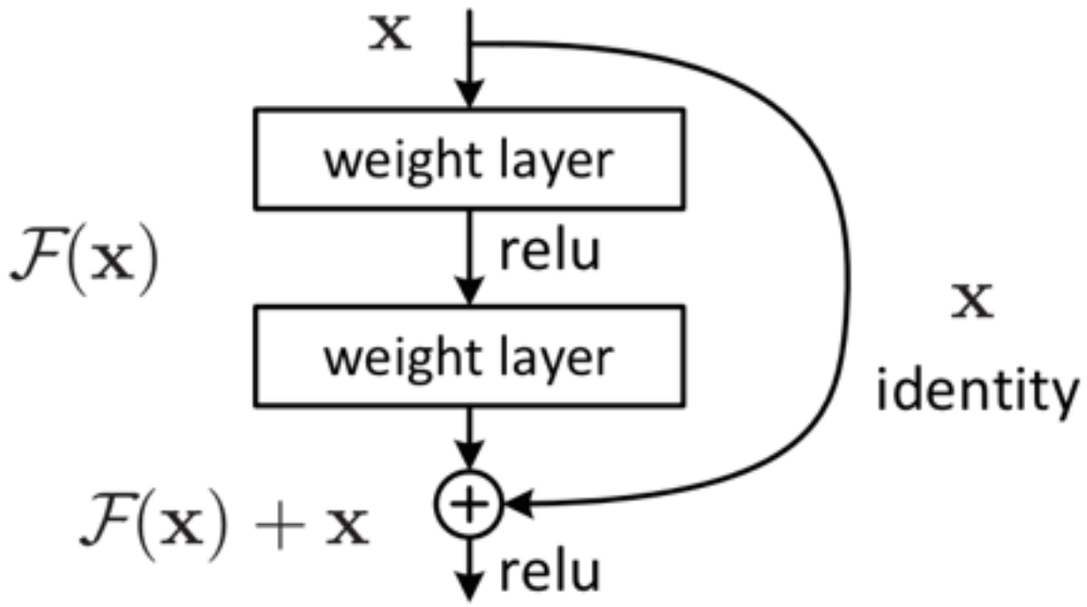
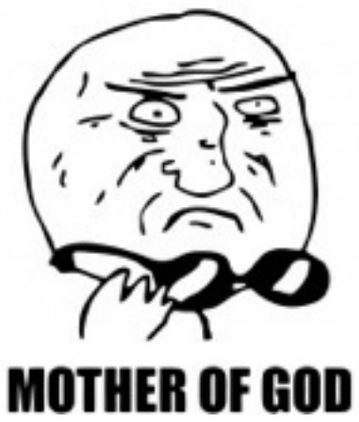
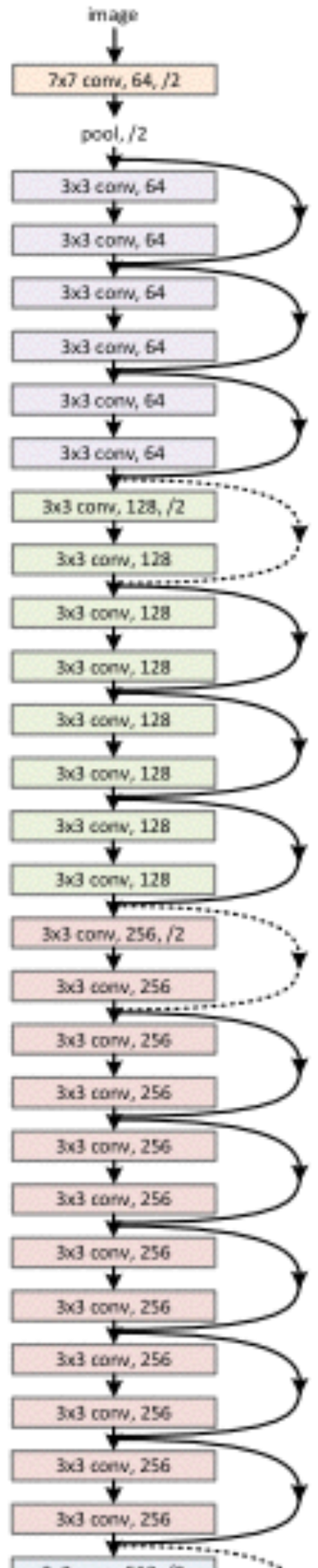


Figure 2. Residual learning: a building block.



Network can decide how deep it needs to be…



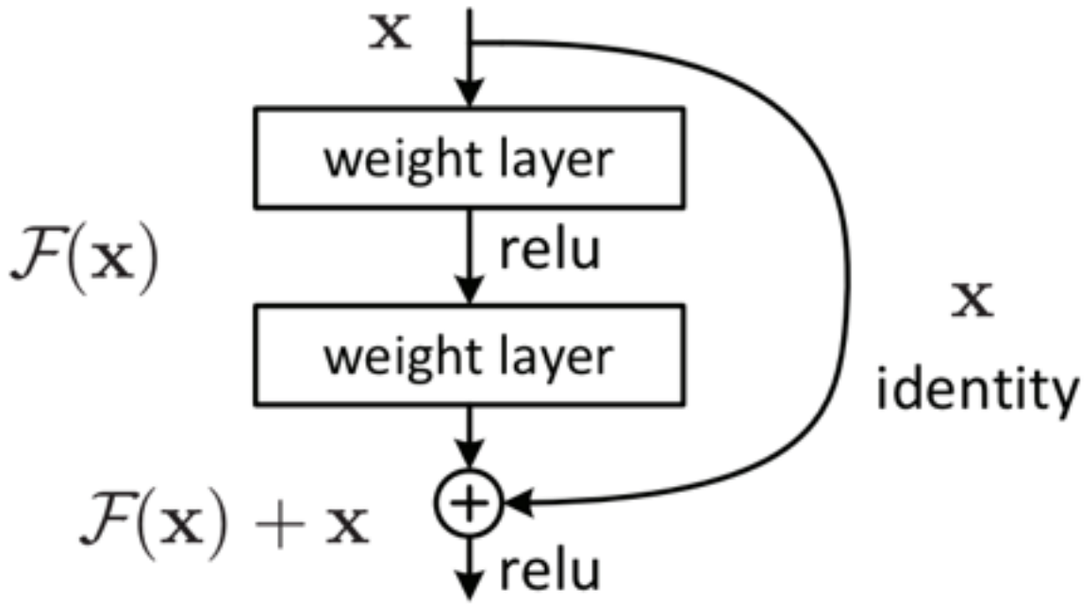
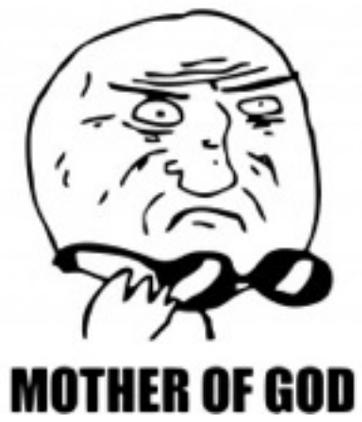
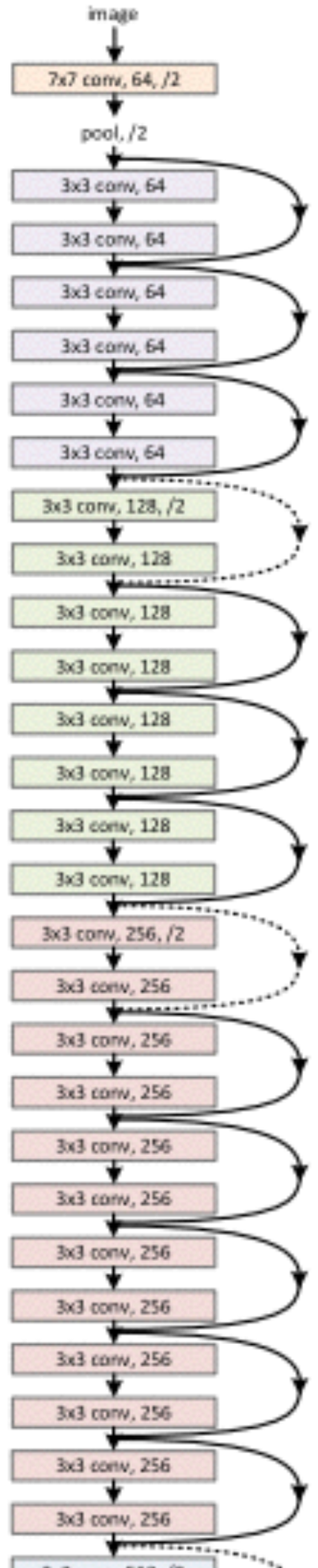


Figure 2. Residual learning: a building block.

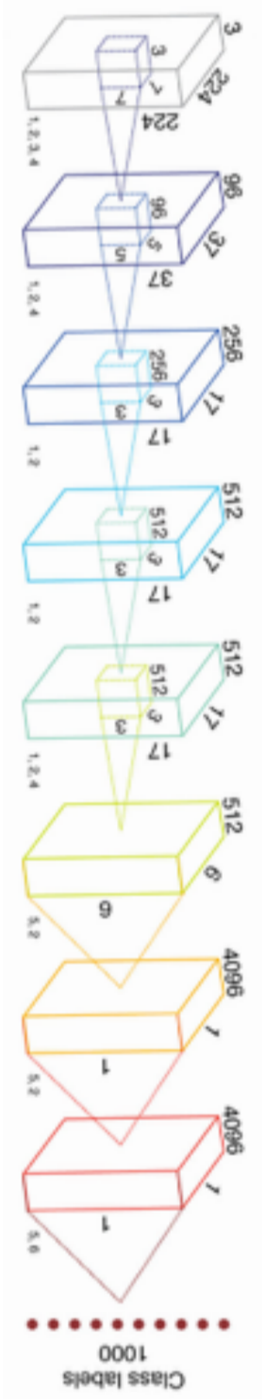


Network can decide how deep it needs to be…

“The identity connections introduce neither extra parameter nor computation complexity”



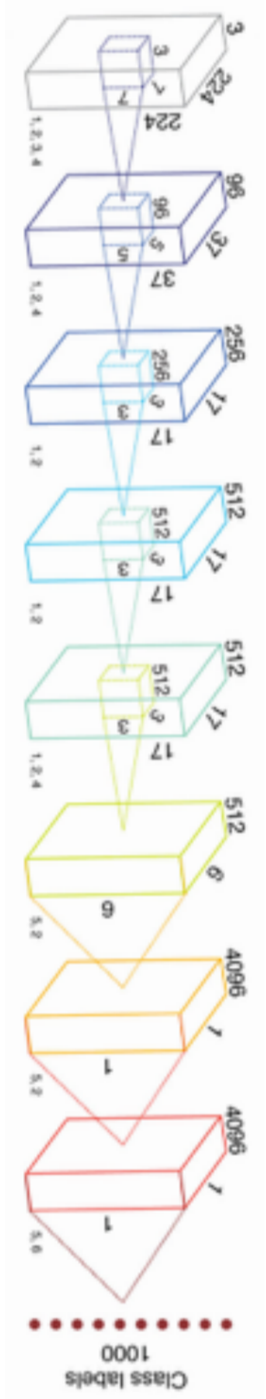
2012



8 layers

15.31% error

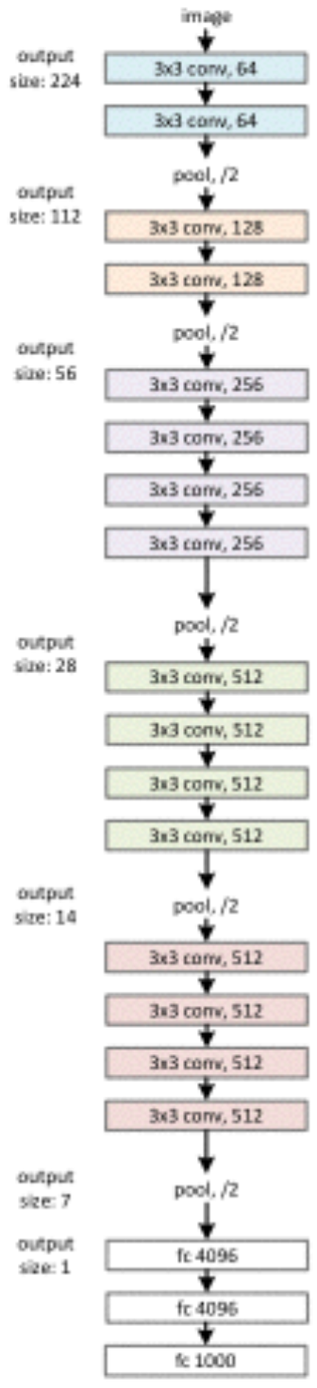
2013



9 layers, 2x params

11.74% error

2014



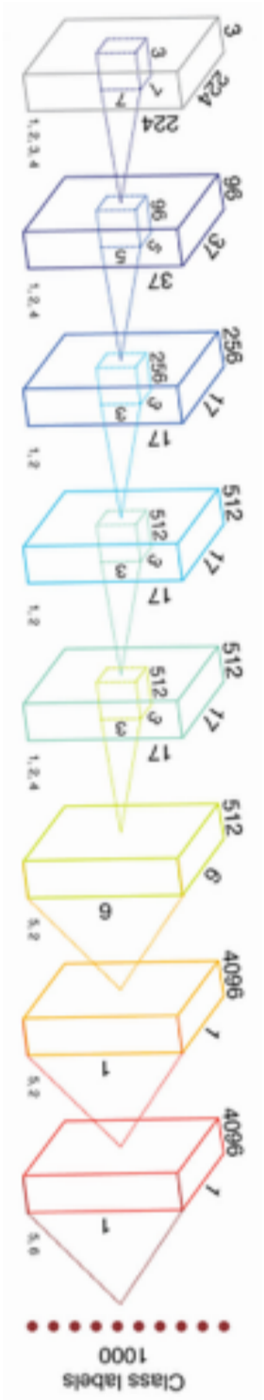
19 layers

7.41% error

2015

?

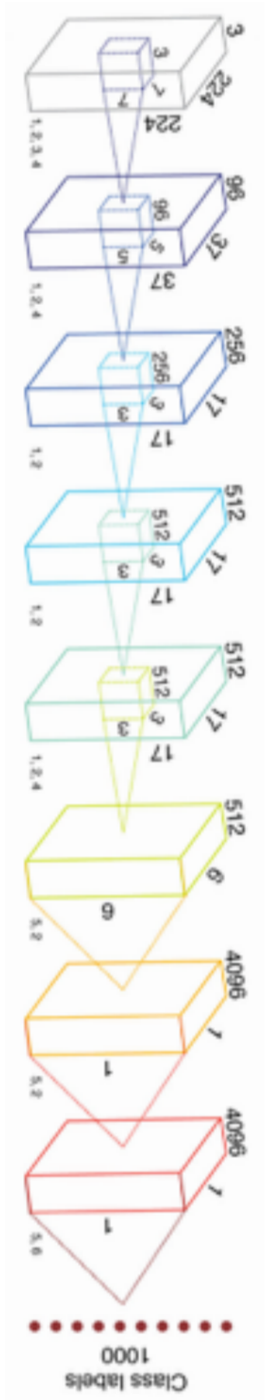
2012



8 layers

15.31% error

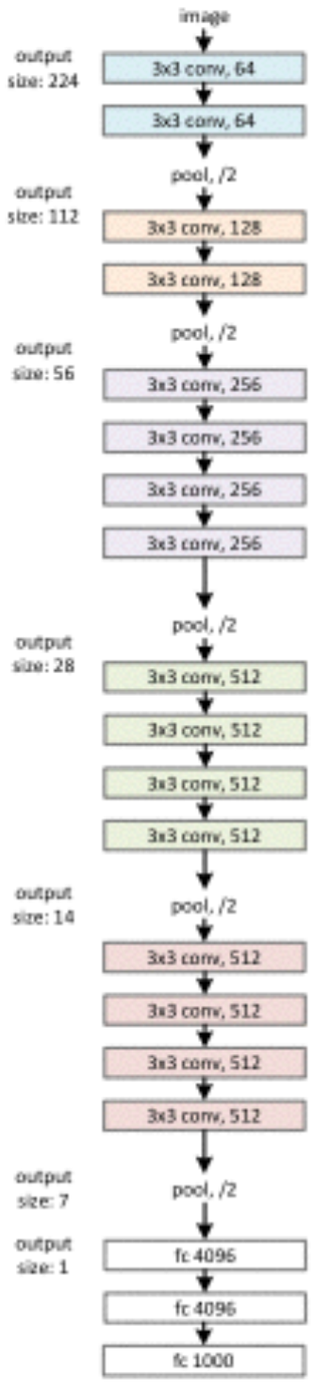
2013



9 layers, 2x params

11.74% error

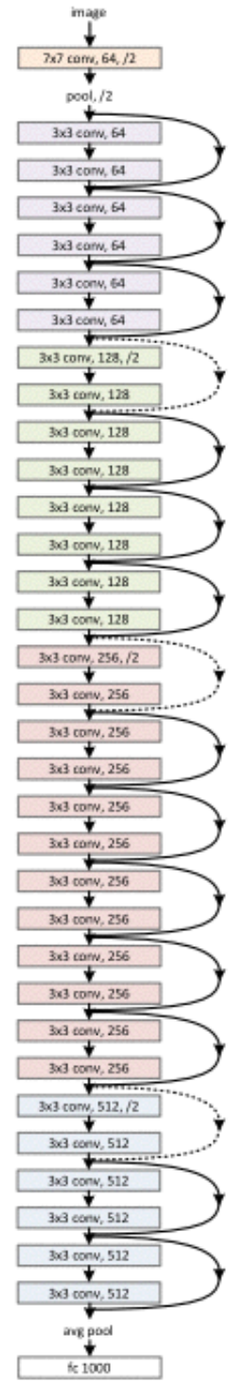
2014



19 layers

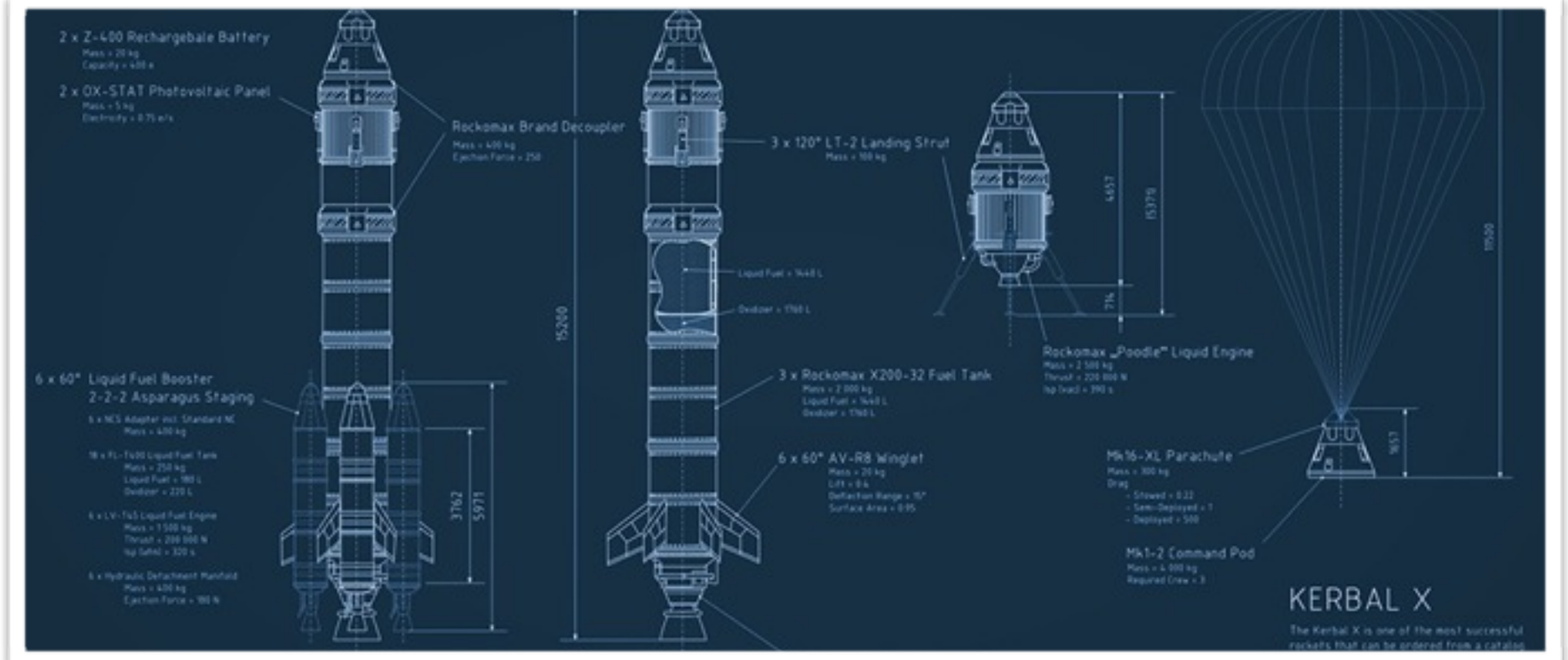
7.41% error

2015



152 layers

3.57% error



Experiments and Details

Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

- Lots of convolutional 3x3 layers
- VGG complexity is 19.6 billion FLOPs
  34-layer-ResNet is 3.6 bln. FLOPs

Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

- Lots of convolutional 3x3 layers
- VGG complexity is 19.6 billion FLOPs
  34-layer-ResNet is 3.6 bln. FLOPs

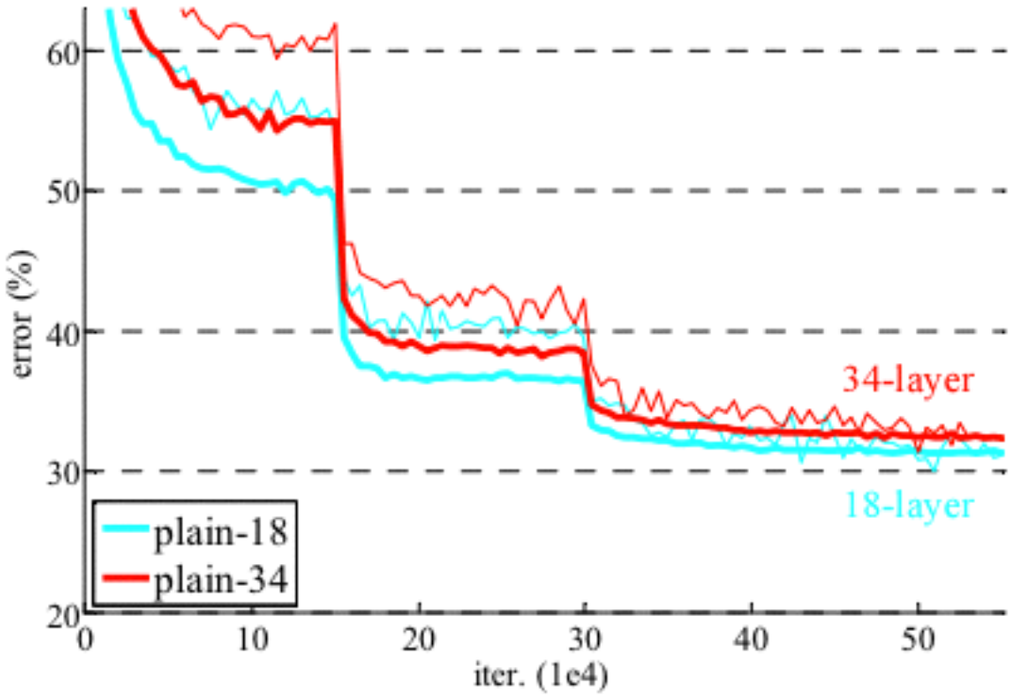
- Batch normalization
- SGD with batch size 256
- (up to) 600,000 iterations
- LR 0.1 (divided by 10 when error plateaus)
- Momentum 0.9
- No dropout
- Weight decay 0.0001

Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

- Lots of convolutional 3x3 layers
- VGG complexity is 19.6 billion FLOPs
  34-layer-ResNet is 3.6 bln. FLOPs

- Batch normalization
- SGD with batch size 256
- (up to) 600,000 iterations
- LR 0.1 (divided by 10 when error plateaus)
- Momentum 0.9
- No dropout
- Weight decay 0.0001

- 1.28 million training images
- 50,000 validation
- 100,000 test



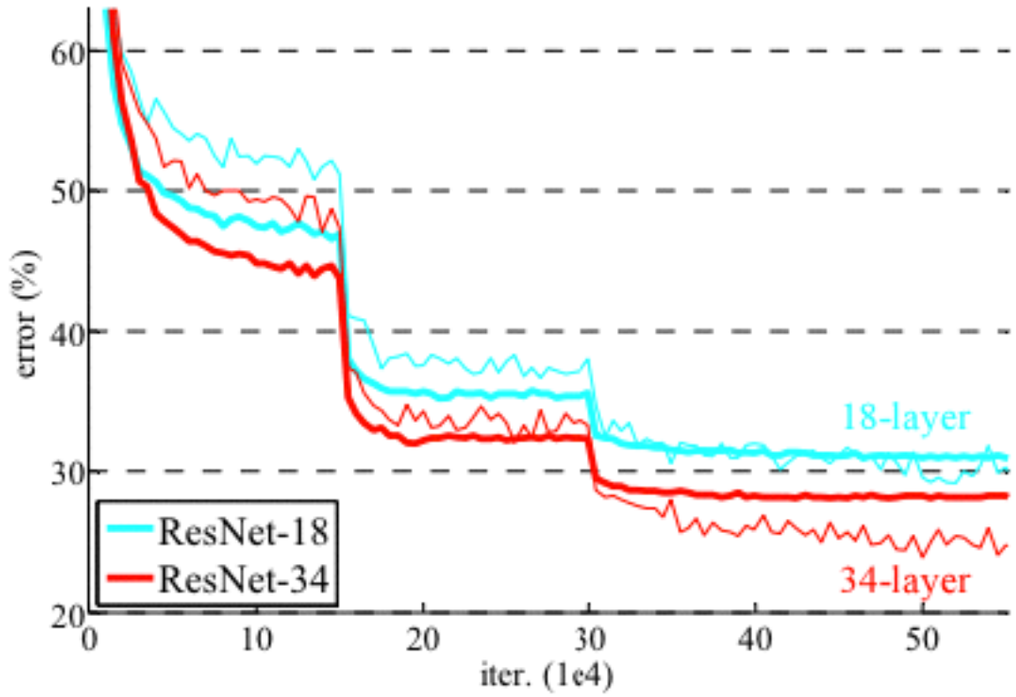
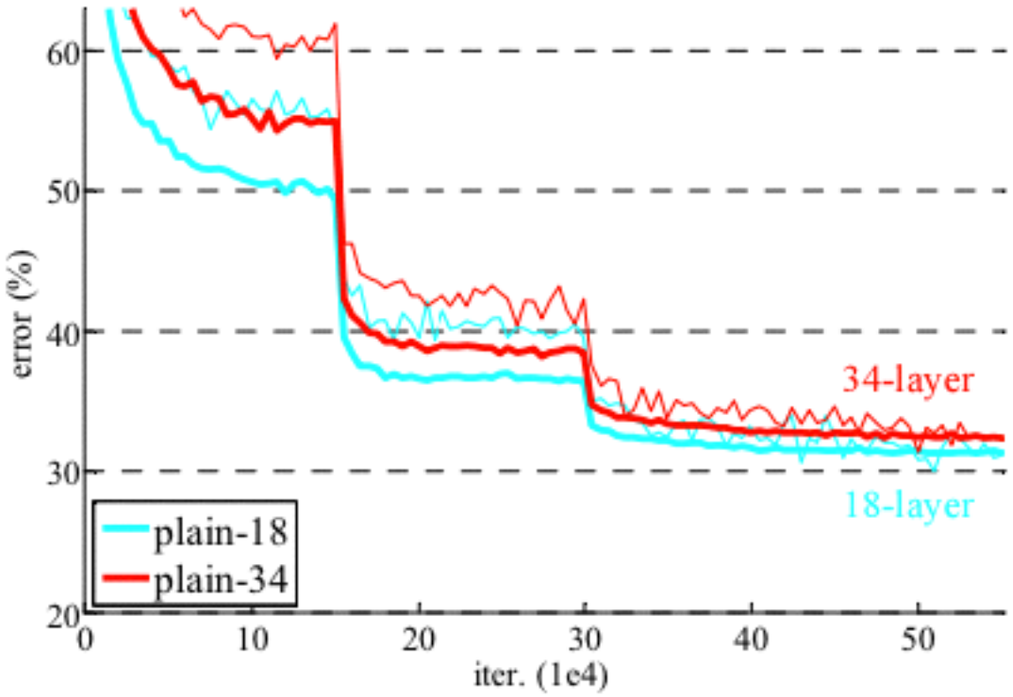


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

Table 2. Top-1 error (%, 10-crop testing) on ImageNet validation. Here the ResNets have no extra parameter compared to their plain counterparts. Fig. 4 shows the training procedures.

- 34-layer ResNet has **lower training error**. This indicates that the **degradation problem is well addressed** and we manage to obtain accuracy **gains from increased depth**.



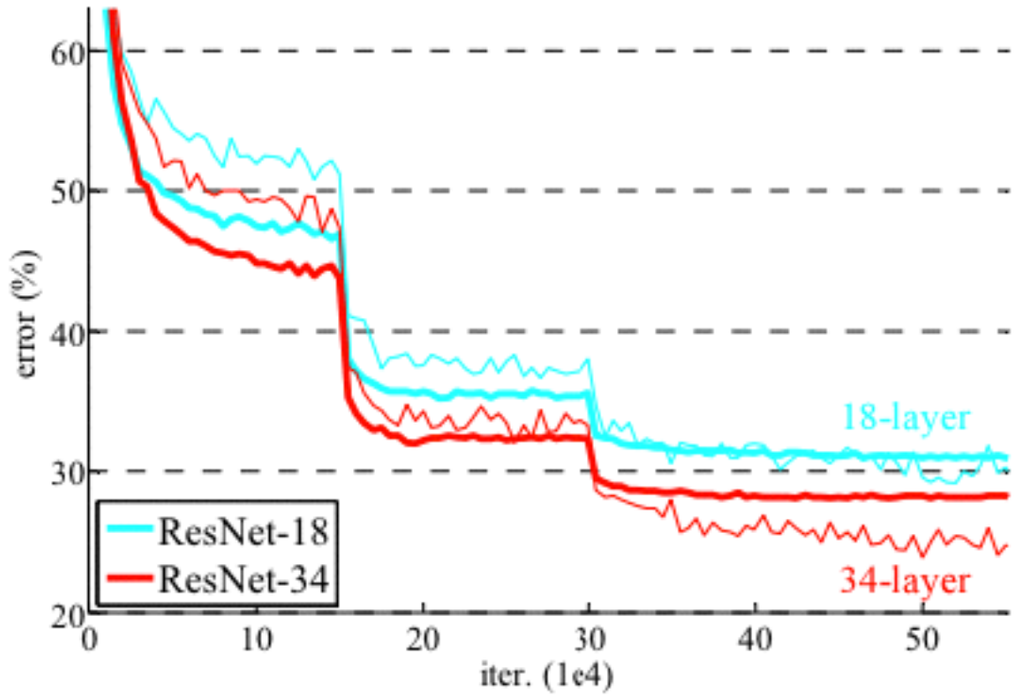
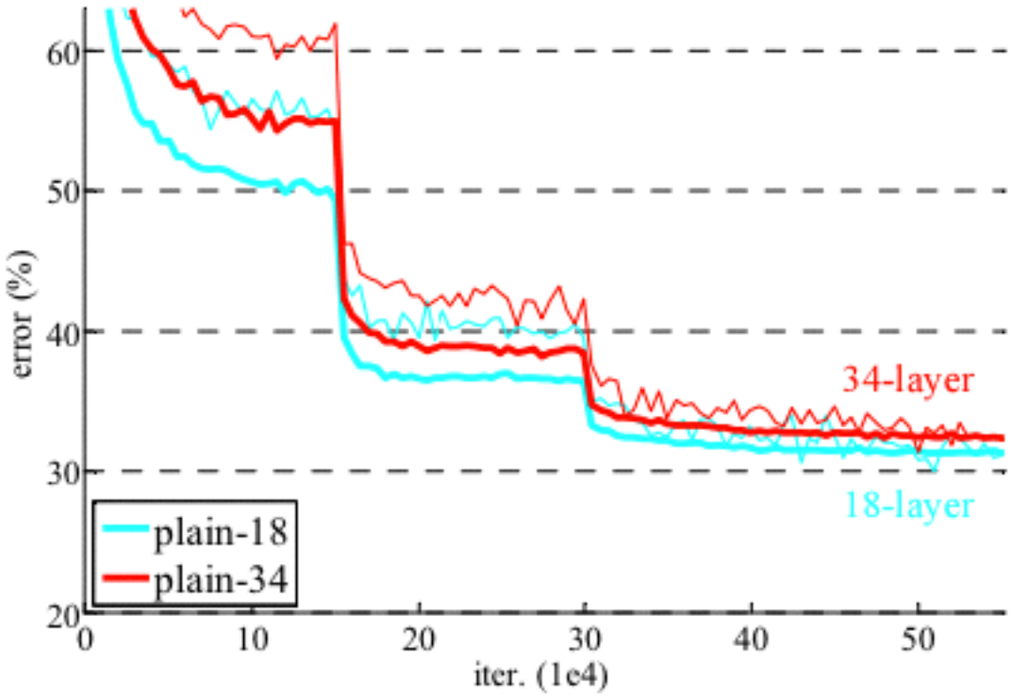


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

Table 2. Top-1 error (%, 10-crop testing) on ImageNet validation. Here the ResNets have no extra parameter compared to their plain counterparts. Fig. 4 shows the training procedures.

- 34-layer ResNet has **lower training error**. This indicates that the **degradation problem is well addressed** and we manage to obtain accuracy **gains from increased depth**.
- 34-layer-ResNet **reduces the top-1 error** by 3.5%



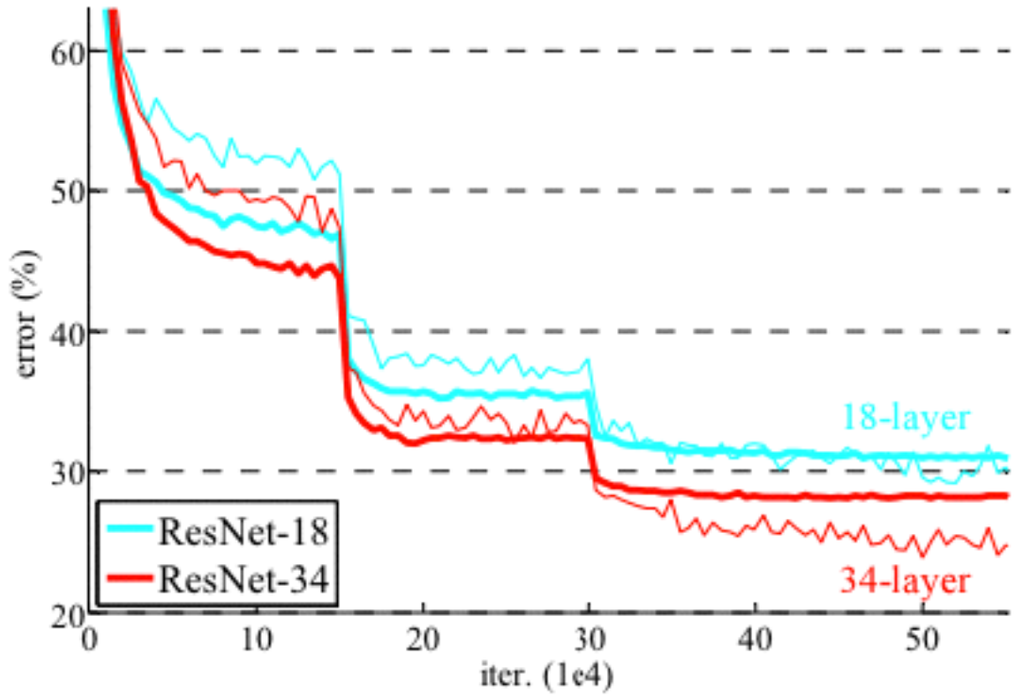


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

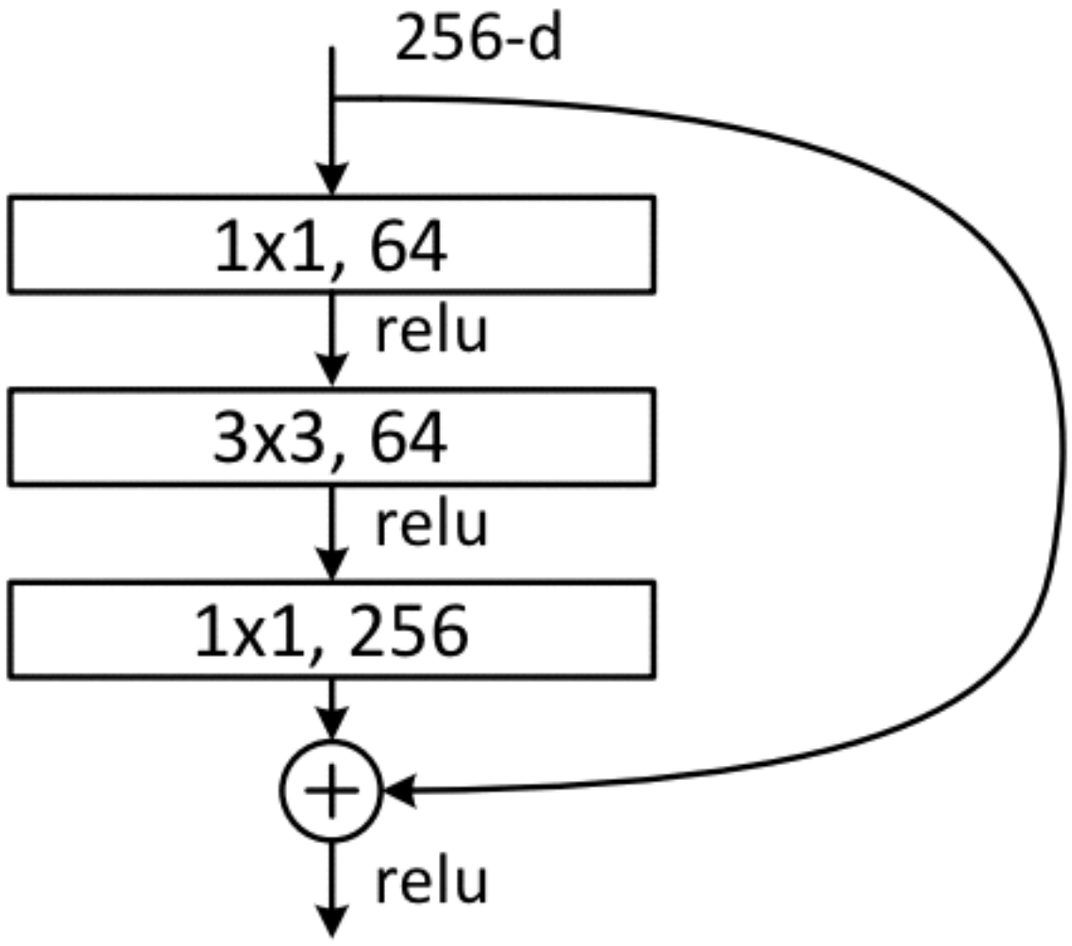
	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

Table 2. Top-1 error (%, 10-crop testing) on ImageNet validation. Here the ResNets have no extra parameter compared to their plain counterparts. Fig. 4 shows the training procedures.

- 34-layer ResNet has **lower training error**. This indicates that the **degradation problem is well addressed** and we manage to obtain accuracy **gains from increased depth**.
- 34-layer-ResNet **reduces the top-1 error** by 3.5%
- 18-layer ResNet **converges faster** and thus ResNet eases the optimization by providing faster convergence at the early stage.

Going Deeper

Due to time complexity the usual building block is replaced by *Bottleneck Block*



50 / 101 / 152 - layer ResNets are build from those blocks

method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

Table 4. Error rates (%) of **single-model** results on the ImageNet validation set (except † reported on the test set).

method	top-5 err. (**test**)
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66
VGG [41] (v5)	6.8
PReLU-net [13]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

Table 5. Error rates (%) of **ensembles**. The top-5 error is on the test set of ImageNet and reported by the test server.

Analysis on CIFAR-10

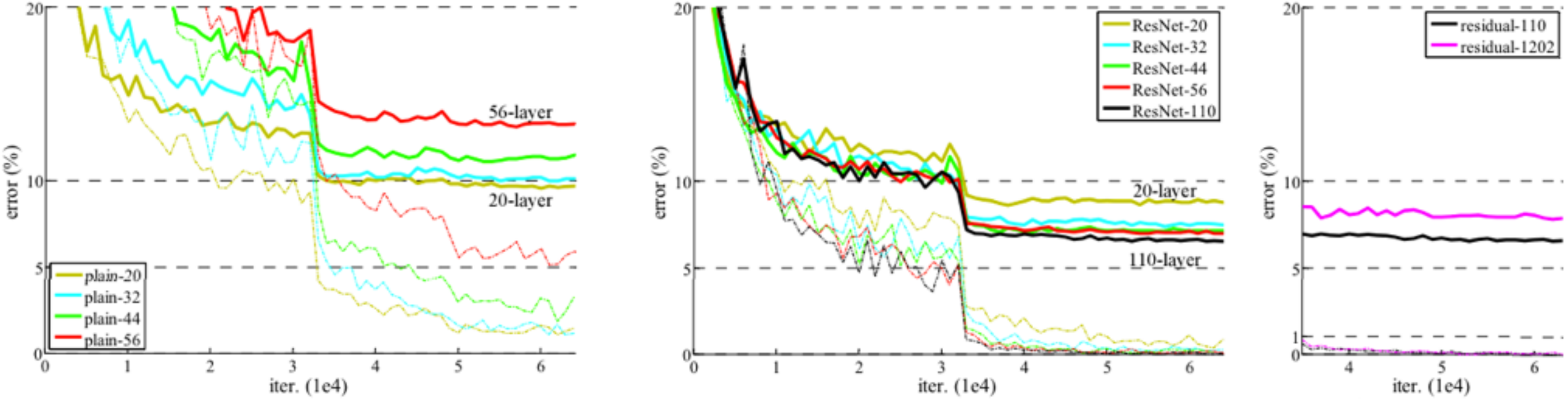


Figure 6. Training on **CIFAR-10**. Dashed lines denote training error, and bold lines denote testing error. **Left**: plain networks. The error of plain-110 is higher than 60% and not displayed. **Middle**: ResNets. **Right**: ResNets with 110 and 1202 layers.

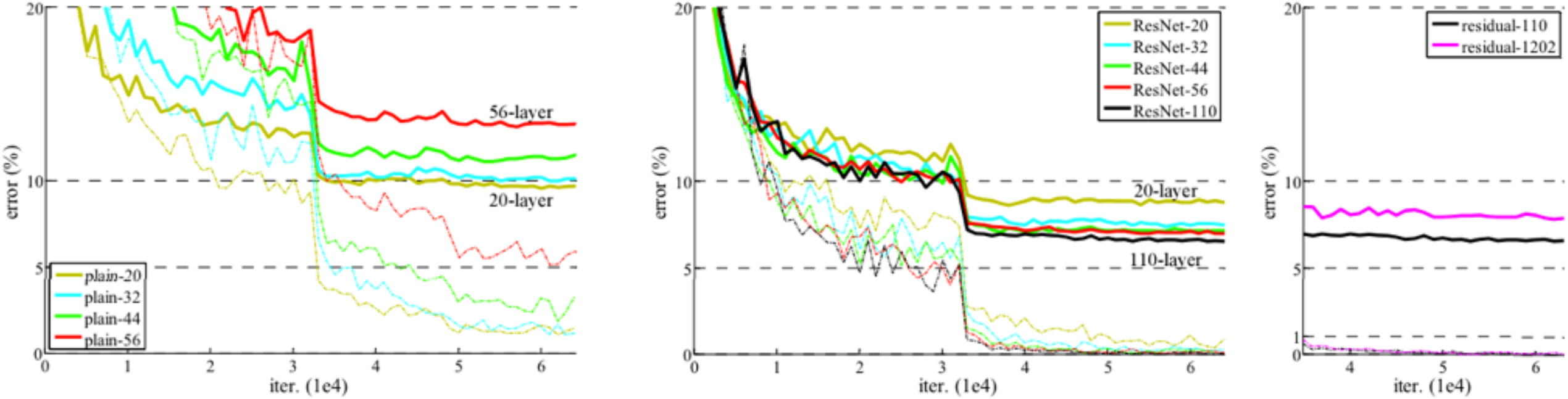
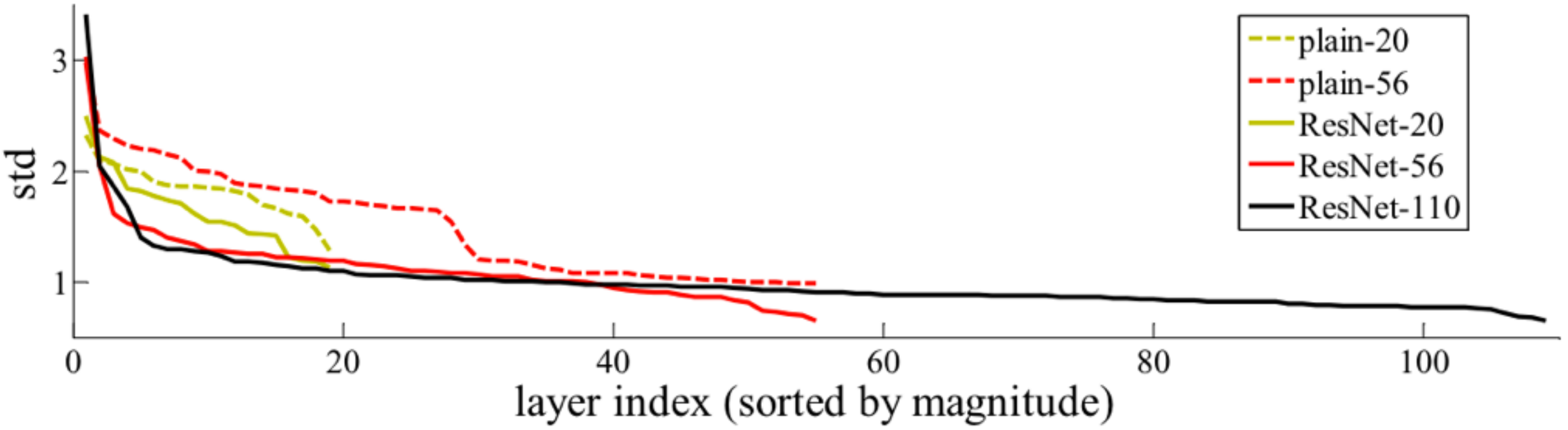
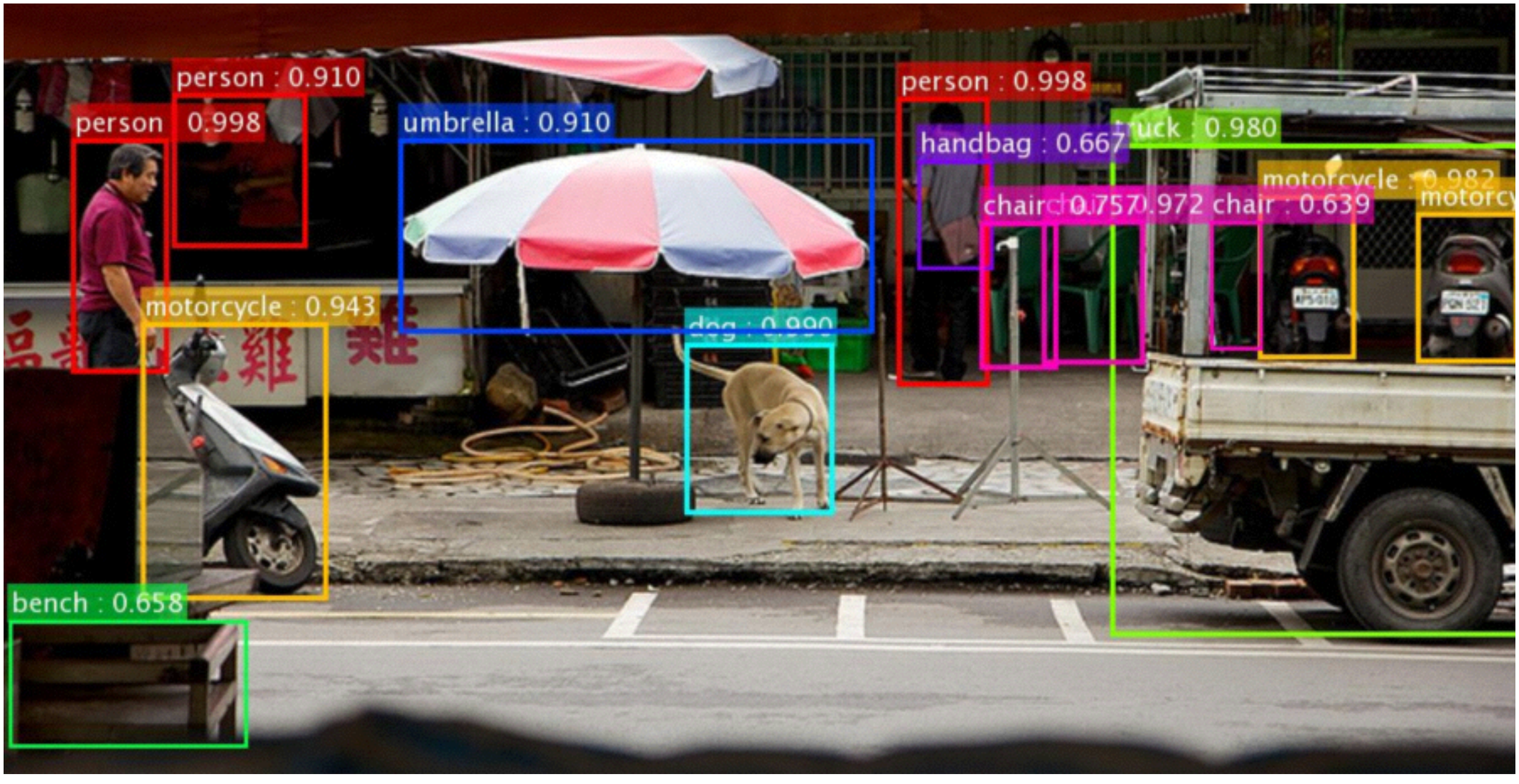


Figure 6. Training on **CIFAR-10**. Dashed lines denote training error, and bold lines denote testing error. **Left**: plain networks. The error of plain-110 is higher than 60% and not displayed. **Middle**: ResNets. **Right**: ResNets with 110 and 1202 layers.



ImageNet Classification 2015	1st	3.57% error
ImageNet Object Detection 2015	1st	194 / 200 categories
ImageNet Object Localization 2015	1st	9.02% error
COCO Detection 2015	1st	37.3%
COCO Segmentation 2015	1st	28.2%



http://research.microsoft.com/en-us/um/people/kahe/ilsvrc15/ilsvrc2015_deep_residual_learning_kaiminghe.pdf