Deep Residual Learning for Image Recognition

Authors: **Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun**
Microsoft Research

Presented by: Kang Zhao and Yar Rouf

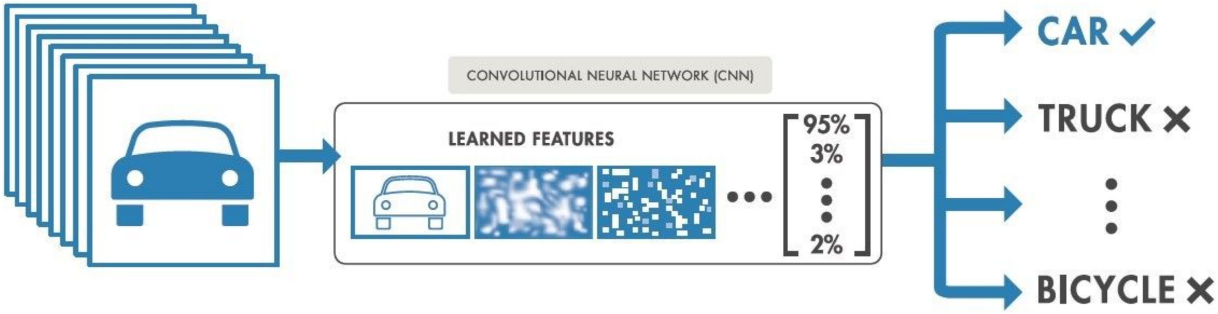
Overview

- Background
- Problem Domain
- Related Works
- Deep Residual Learning
- Experiments & Results
- Conclusions

Background

Introduction of CNN

- Deep Convolutional Neural Networks
  - Breakthrough for Image Classification
  - Integrates low/mid/high-level features and classifiers in an end-to-end multilayer fashion
  - **“...Network depth is of crucial importance”**

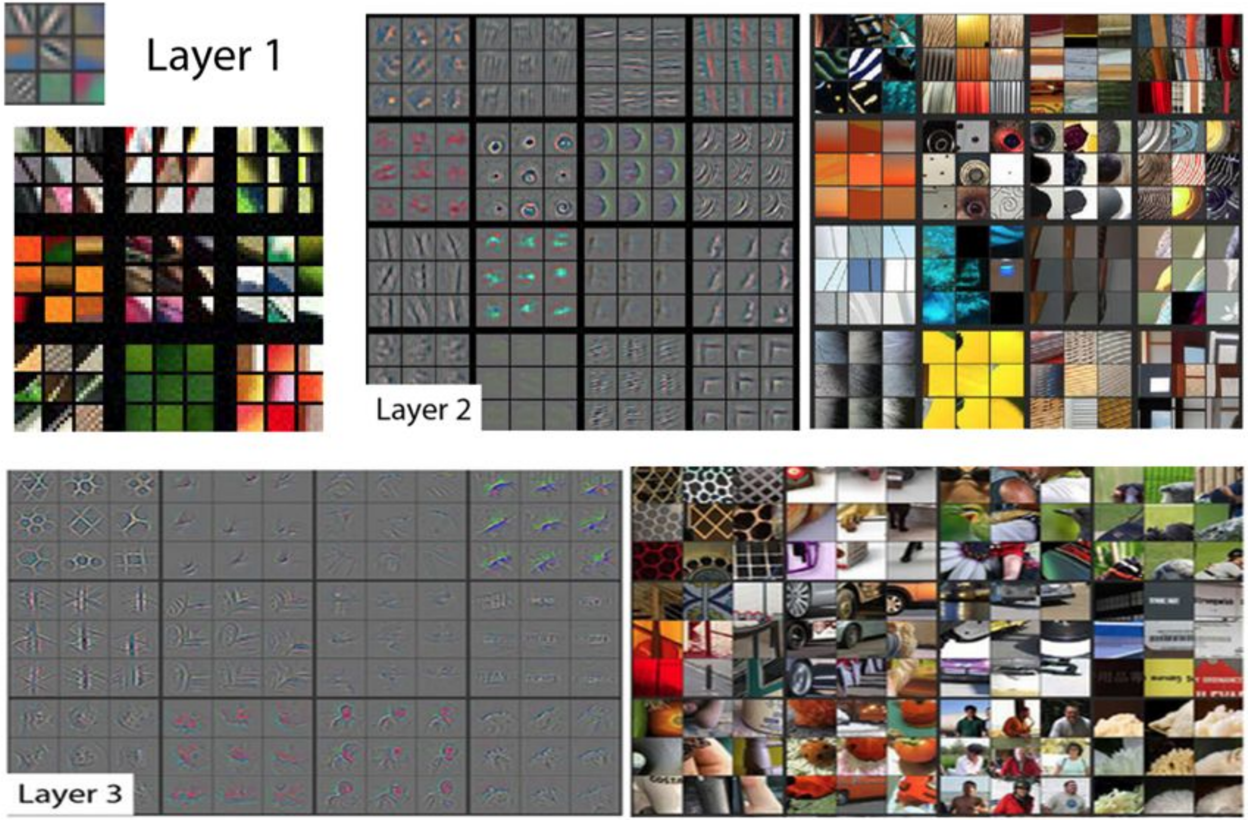


Benefits of Deeper Networks

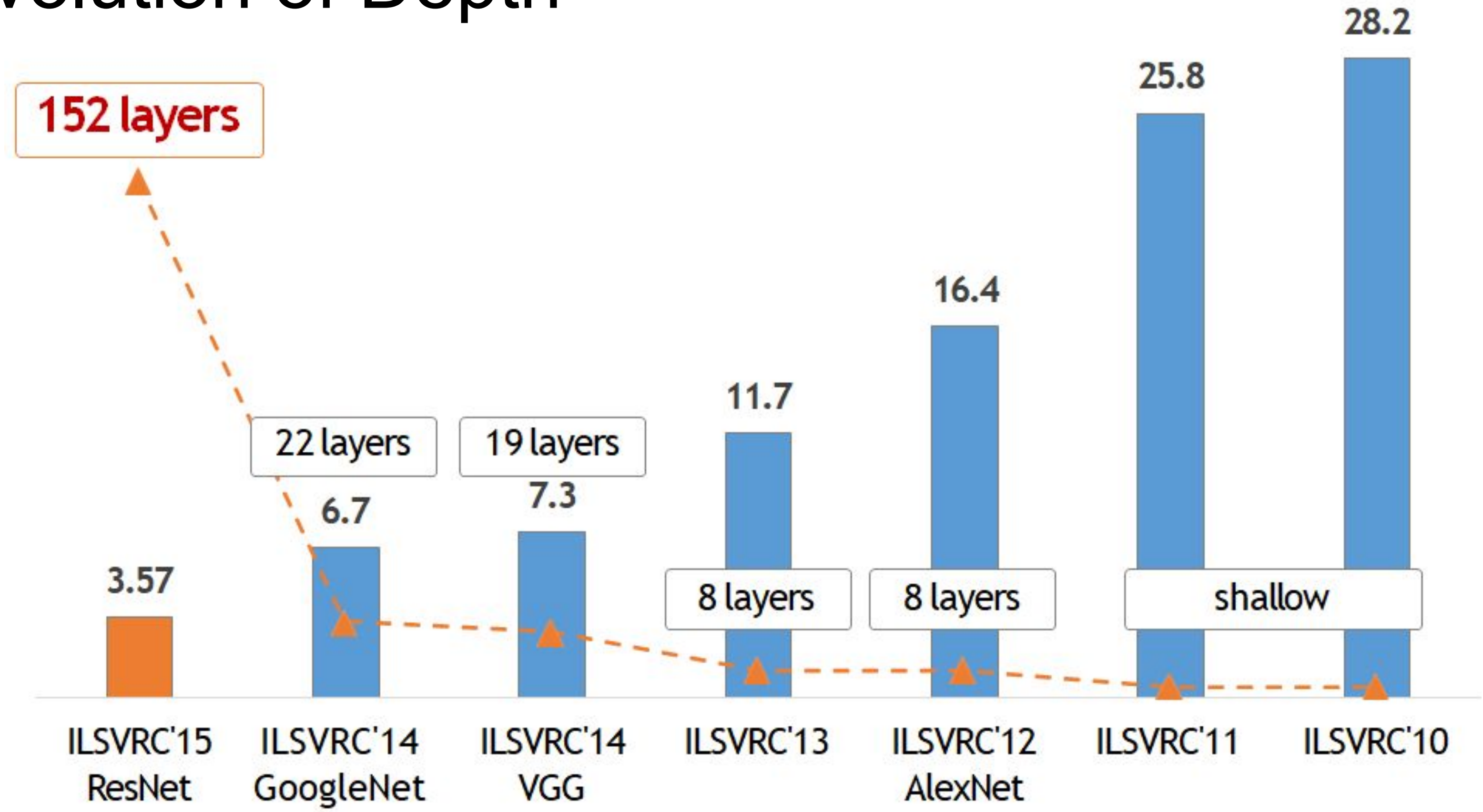
- Deeper features achieve better representation of data

- Deeper network can cover more complex problems
  - Receptive field size ↑
  - Non-linearity ↑

Benefits of Deeper Networks

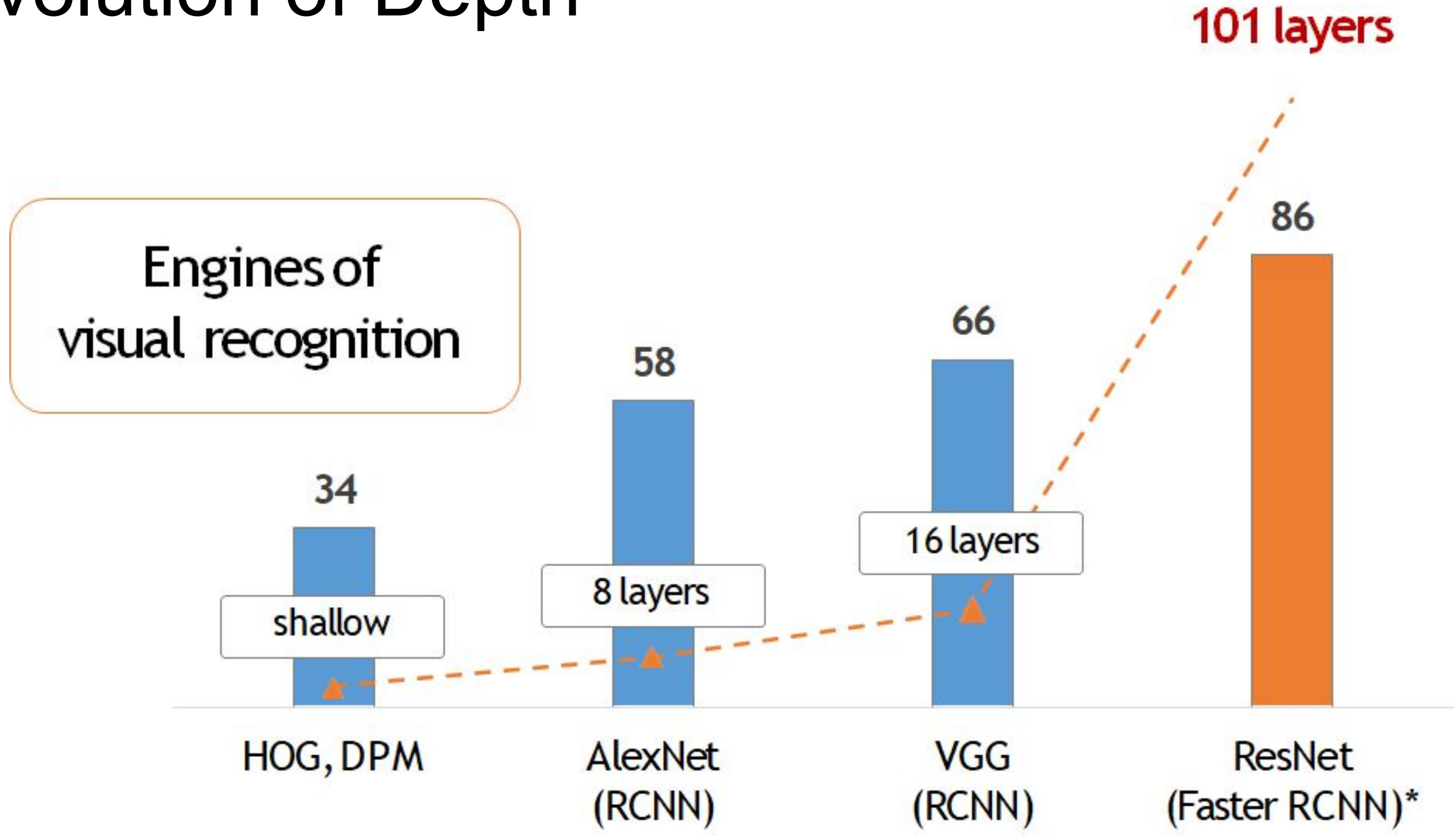


Revolution of Depth



ImageNet Classification top-5 error (%)

Revolution of Depth



PASCAL VOC 2007 Object Detection mAP (%)

Problem Domain

The Original Obstacle for deeper network

- Overfitting
- Vanishing and Exploding gradient problem has been largely addressed...
  - Normalized initialization
  - Intermediate normalization layers (batch normalization)

- There is a New Problem!
  - **DEGRADATION**

The Degradation Problem

- When the network depth increases, accuracy gets saturated, and then **degrades rapidly**
- Adding more layers leads to higher training error

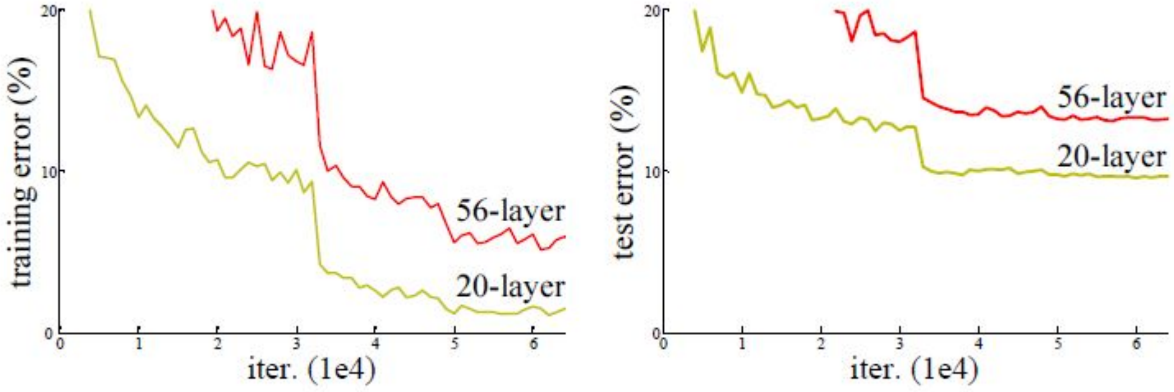
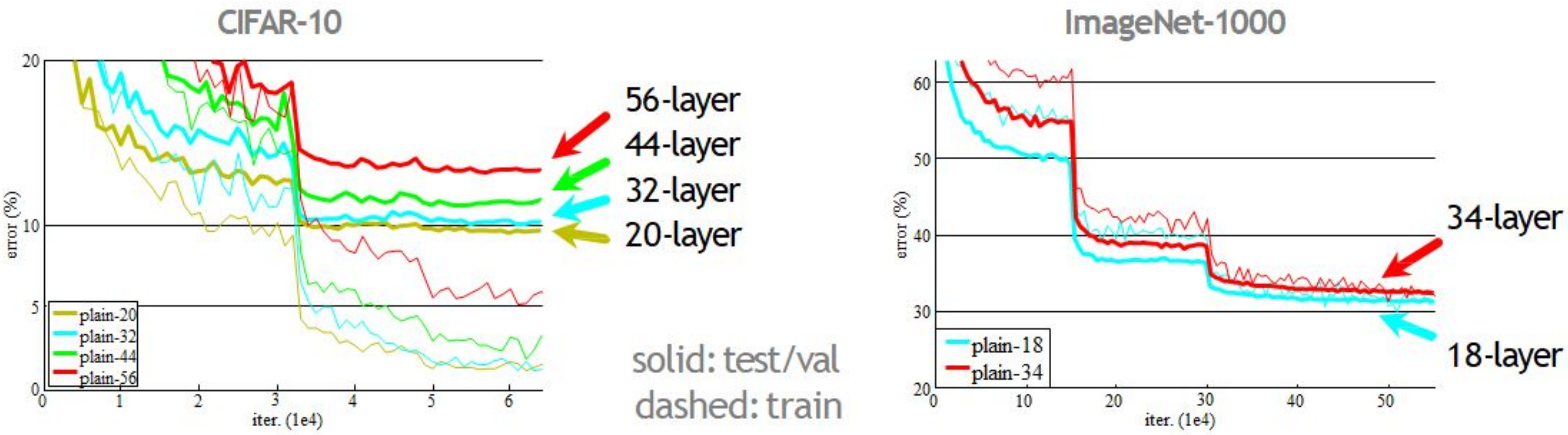


Figure 1: Training Error (left) and Test error (right) on
CIFAR-10 with “Plain” networks

The Degradation Problem



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

The Response to Degradation

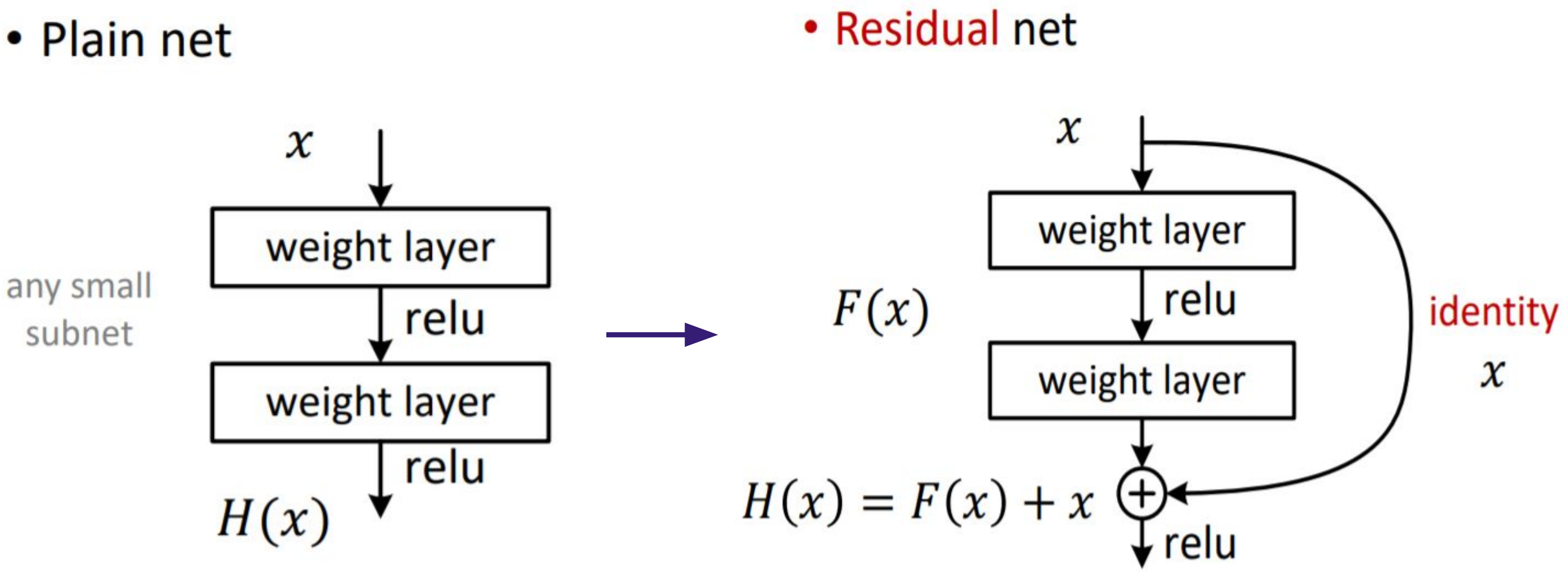
- Shallow Networks and Deeper Networks
- Solution **by Construction**
  - By added layers that are Identity mappings to learned shallow model
- Constructed solution indicated that…
  - Deeper Model should produce no higher training error than its shallower counterpart (superset of solution space)
  - **Experiments** are unable to find solutions that are comparably better than the **constructed solution**

Related Works

- Residual Representations
  - VLAD
  - Fisher Vector
  - Multigrid and Hierarchical Precondition
- Shortcut Connections
  - Highway Networks
    - Are data dependent and require parameters

Deep Residual Learning

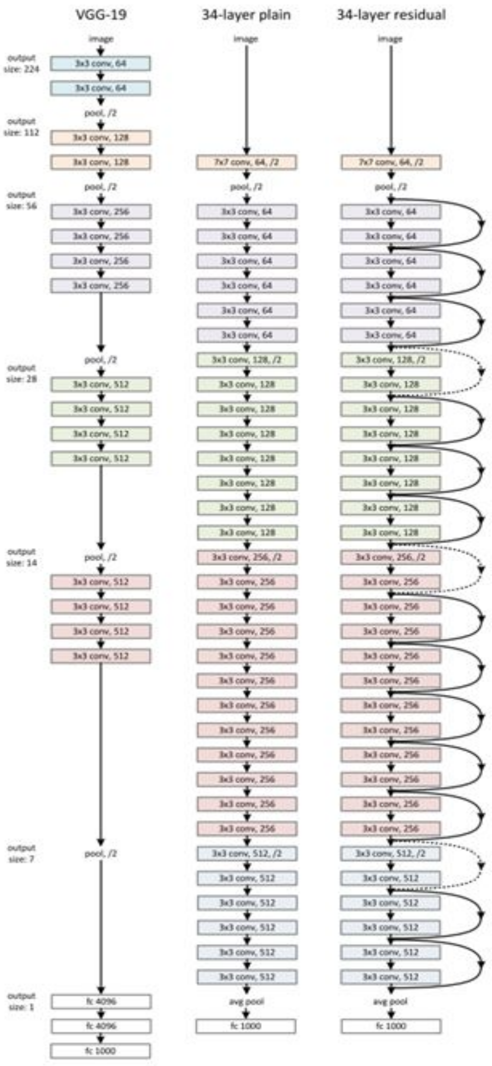
Deep Residual Learning



Learn the residual mapping F(x) rather than unreferenced H(x)

Design Deep Residual Network

- Keep it simple, just deep
- Design based on VGG style
  - All 3*3 conv (almost)
  - Batch normalization and ReLU
  - Downsampling: cov with stride of 2
  - Spatial size/2 => # filters *2
- No hidden layer, no dropout



Experiments and Results

Experiments: Dataset

- ImageNet dataset (2012, image recognition)
    - Classification Dataset consisting of 1000 classes
    - Models were trained on 1.28 million training images, and validated on 40k validation images
    - Final result on 100k test images
        - Top-1 and Top-5 Error rates are evaluated
- CIFAR-10 Dataset (image recognition)
    - 50k Training images
    - 10k Testing images
    - 10 classes
- MS COCO (object detection)
    - 80 Object Categories
    - 80k training images
    - 40k images for evaluation

Training

- All plain/ residual nets are trained from scratch

- All plain/ residual nets use Batch normalization

- Standard hyper-parameters & augmentation

Experiments: Plain Net

- **Degradation problem**

- Deeper 34-layer plain net has higher validation error than 18 layer plain net

- 34-layer plain net has higher training error through the whole training procedure

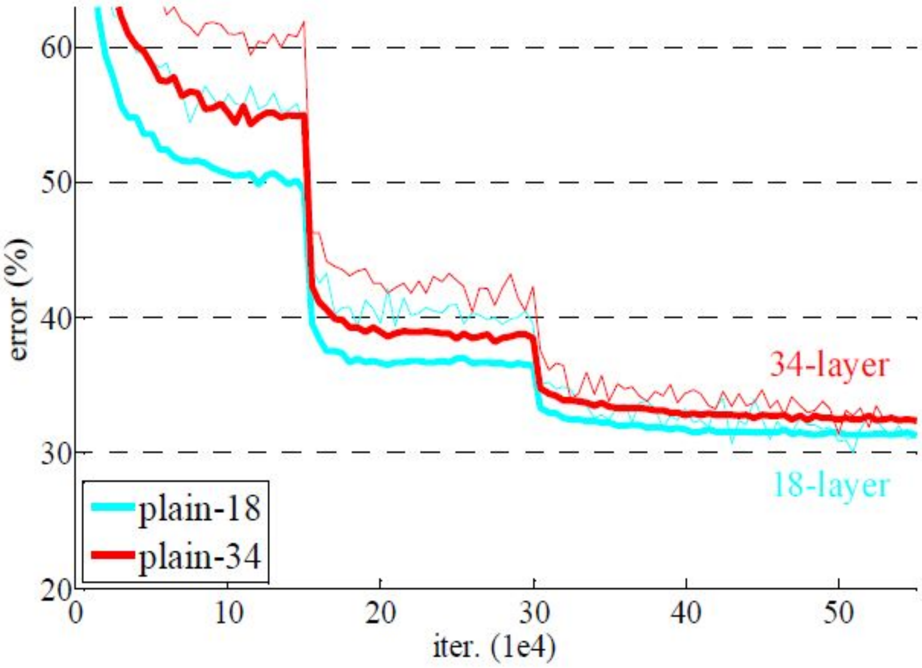


Figure 4: Training on Image Net

Experiments: ResNet

34-Layer ResNet is better than 18-layer ResNet (**by 2.8%)**

- lower training error
- **Degradation problem is addressed**

34-layer ResNet reduced top-1 error by 3.5 percent compared to Plain Net

18-layer plain/residual nets are comparably accurate, but…

- 18-layer Resnet converges faster
- ResNet eases optimization by providing faster convergences

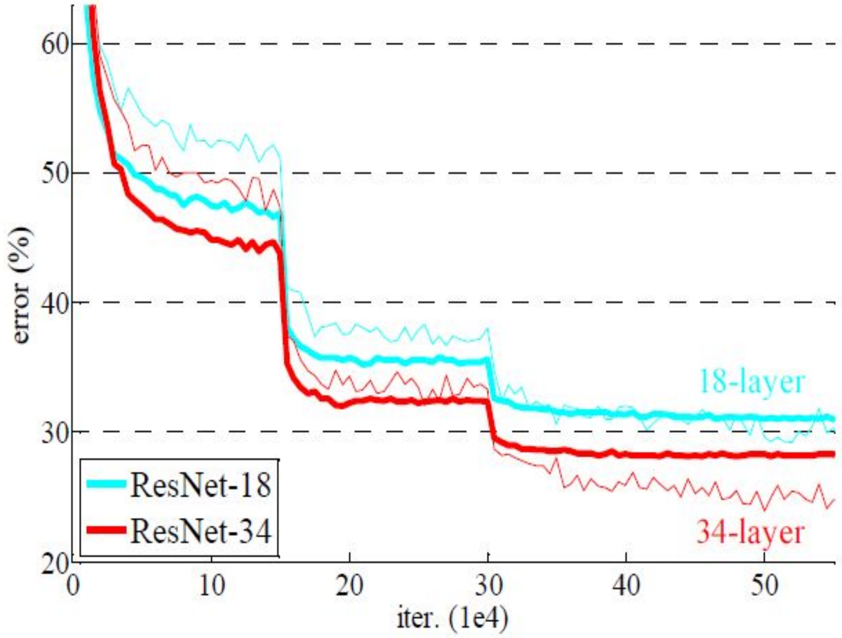


Figure 5: Training on Image Net

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

Table 2: Top-1 Error on ImageNet Validation

Identity vs Projection Shortcuts

(A) Zero Padding Shortcuts

(B) Projection Shortcuts for increasing Dimensions

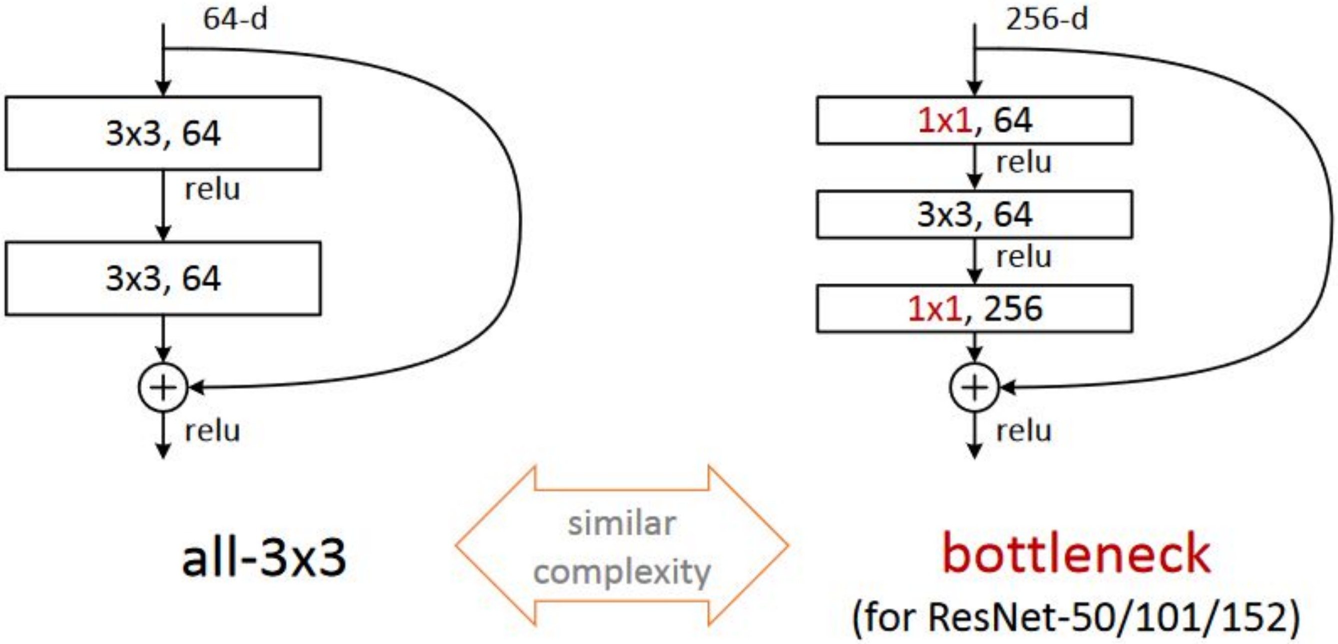
(C) All Shortcuts are projections

Accuracy Increases for Training

model	top-1 err.	top-5 err.
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40

Table 3: Training Error Rates on ImageNet

Deeper Bottleneck Architecture



Deeper Bottleneck Architecture

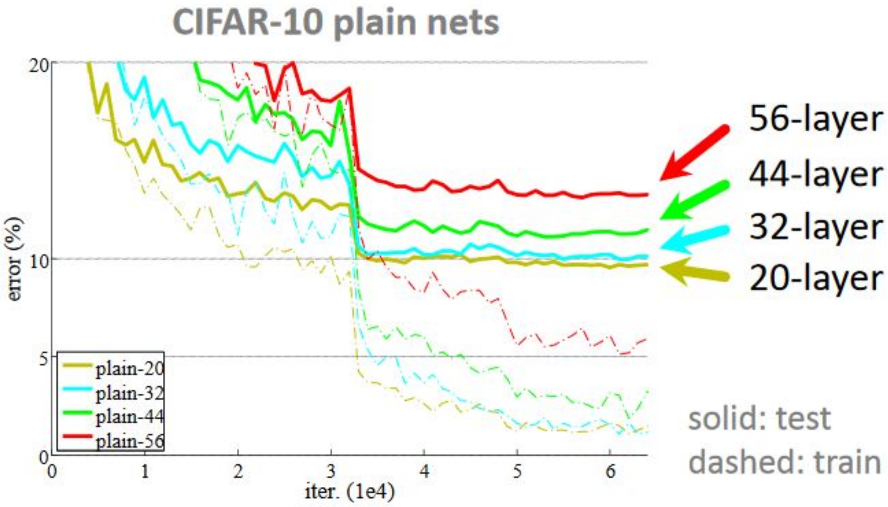
Results in **50-Layer ResNet**

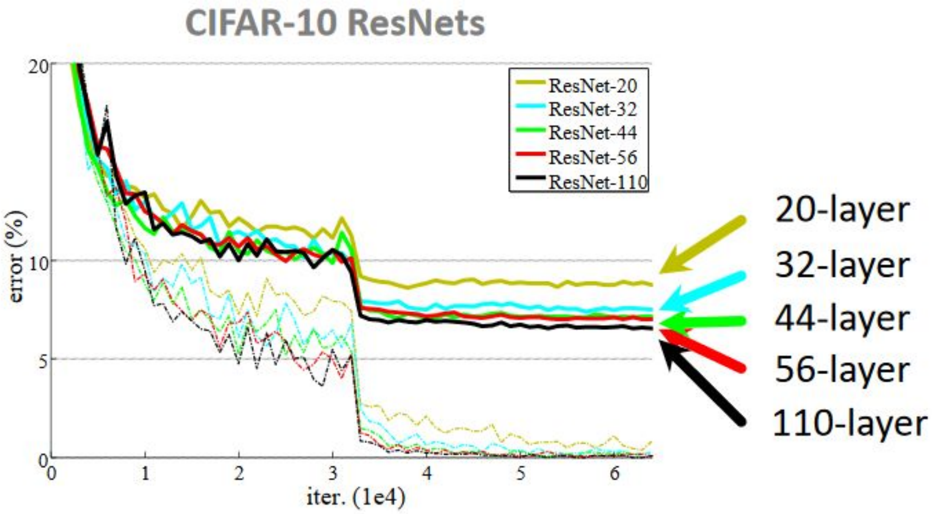
Using more 3-Layer Blocks, we can construct…

101-Layer and 152-Layer ResNet

model	top-1 err.	top-5 err.
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

Experiments: CIFAR-10



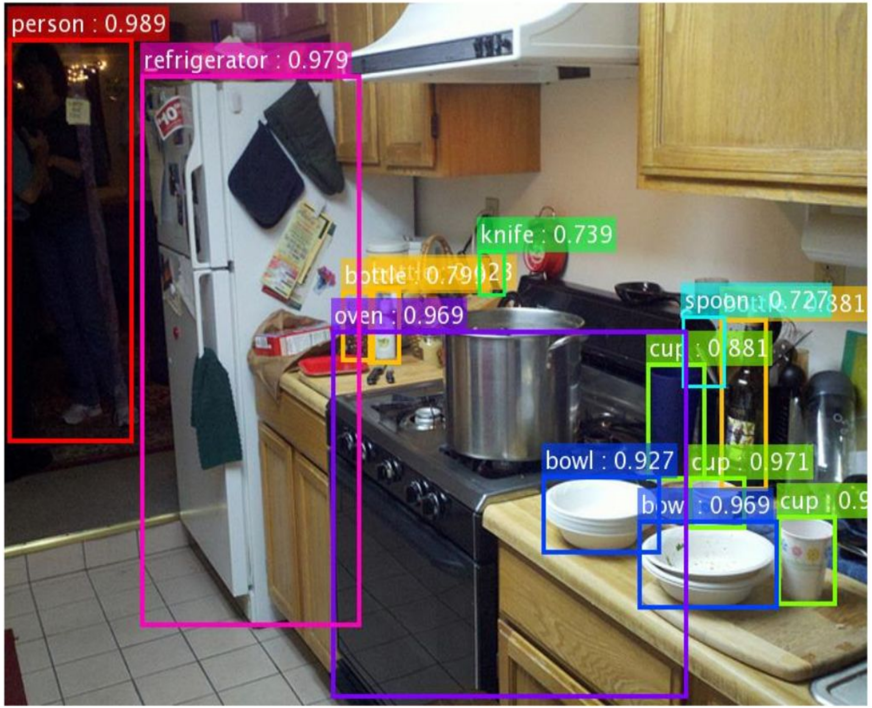


Experiments: Object Detections

- Improved performance for Object Detection on Pascal and COCO
  - Faster-R-CNN
    - Replace VGG-net Method with ResNet

metric	mAP@.5	mAP@[.5, .95]
VGG-16	41.5	21.2
ResNet-101	48.4	27.2

Table 8: Object Detection Map on COCO Validation set



COCO Results from ICCV15 Slides

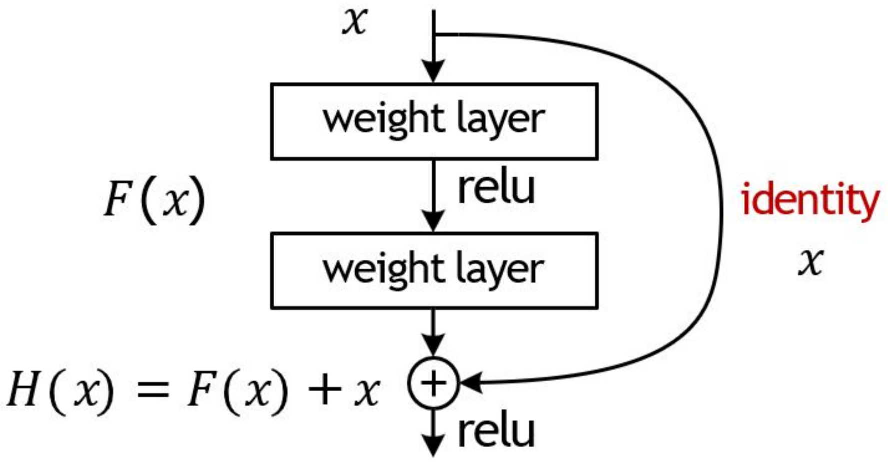
Accomplishments

- ILSVRC 2015 and COCO 2015 Competitions
- 1st place in Classification Competition
- 1st place in ImageNet Detection
- 1st place in Imagenet Localization
- 1st place in COCO Detection
- 1st place in COCO Segmentation
- Most cited paper in Google Scholar Metrics 2018 (over 10k citations)

Insight of ResNet

Identity shortcut

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



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

Very Smooth Feedforward

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

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

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

Very Smooth Feedforward

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

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

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

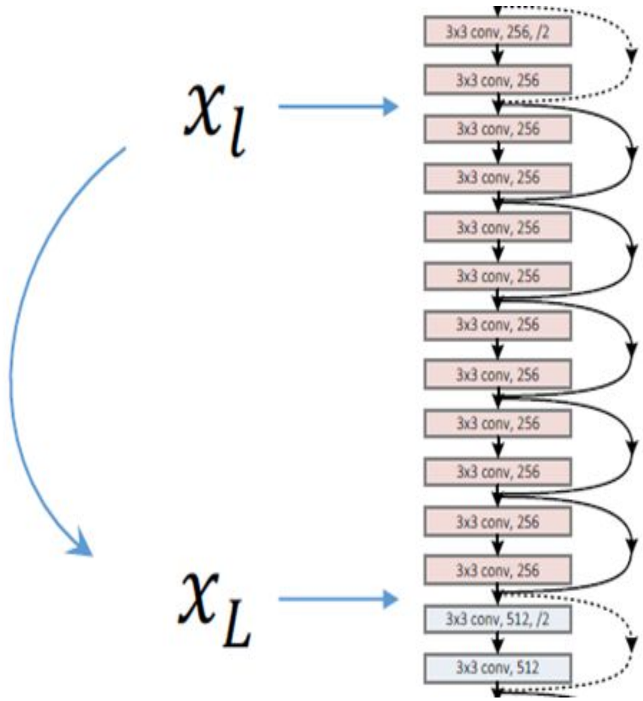
$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$$

Very Smooth Feedforward

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

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

Features from early layers are reused!



Very Smooth Backforward

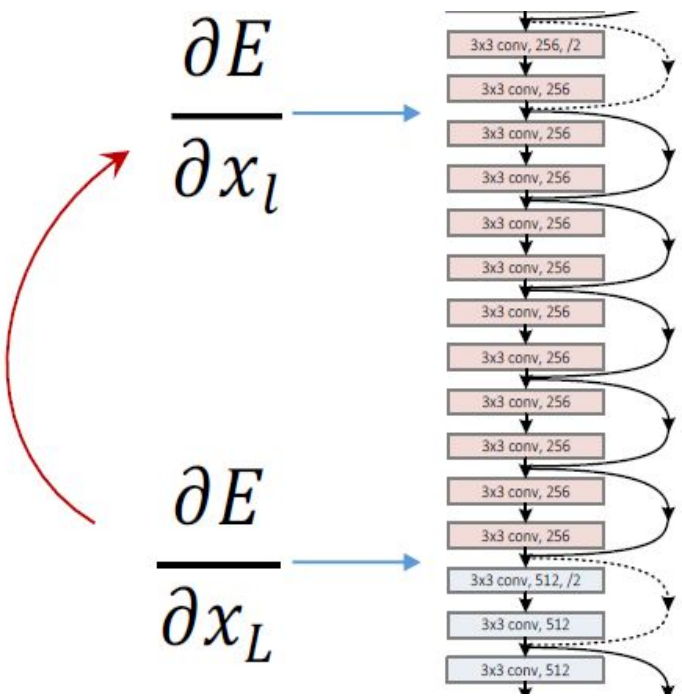
$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i)$$

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

Very Smooth Backforward

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

- 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: $\frac{\partial E}{\partial x_l} = \prod_{i=l}^{L-1} W_i \frac{\partial E}{\partial x_L}$



Easier and Faster Optimization

- At deeper stage, easy to find solution to F(x) as simply zero
- Residual learning converges faster at early stage because of zero initialization of weight
- Gradient flow back to early layers avoiding the gradient vanishing problem

Limitations and Debates

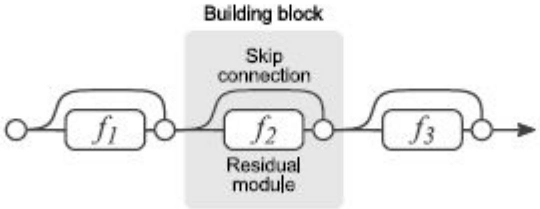
Expensive training with deeper networks

- 152-layer ResNet requires several weeks to converge on the ImageNet dataset

- For too deep network, layers at later stages is suspected to contribute almost nothing for some specific task

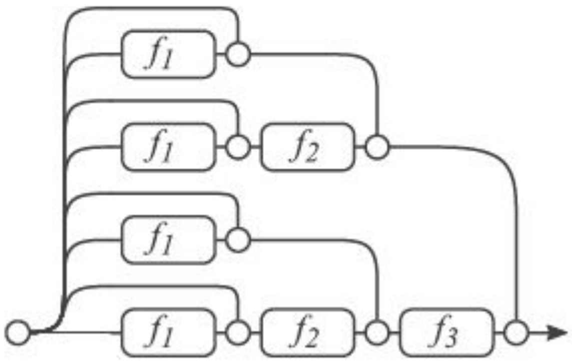
Different Interpretation of ResNet

- ResNet is not a single ultra-deep networks, but very large implicit ensemble of many shallow networks



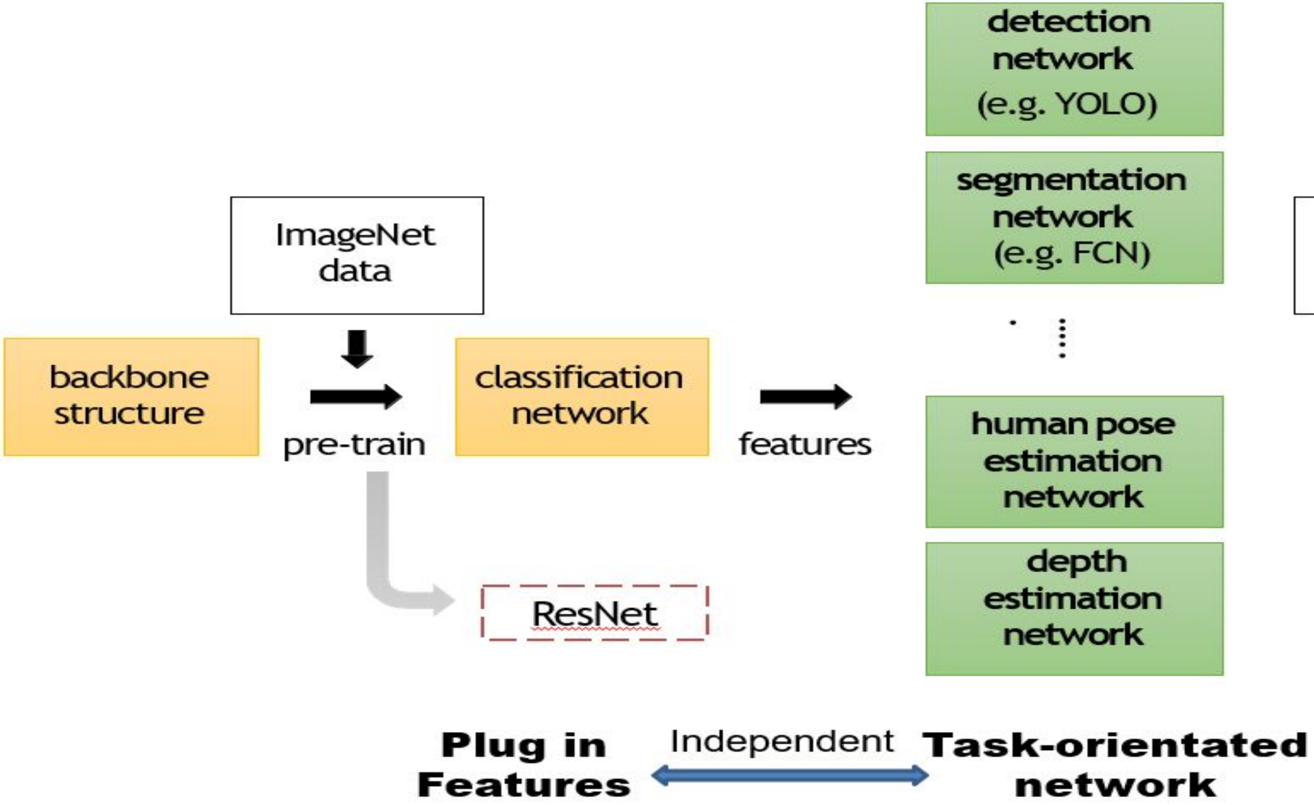
$$
\begin{aligned}
y_3 &= y_2 + f_3(y_2) \\
&= [y_1 + f_2(y_1)] \;+\; f_3(y_1 + f_2(y_1)) \\
&= \big[y_0 + f_1(y_0) + f_2(y_0 + f_1(y_0))\big] \;+\; f_3\big(y_0 + f_1(y_0) + f_2(y_0 + f_1(y_0))\big)
\end{aligned}
$$

- Depth may not be the key of deep learning



Applications of ResNet

From classification to general vision tasks

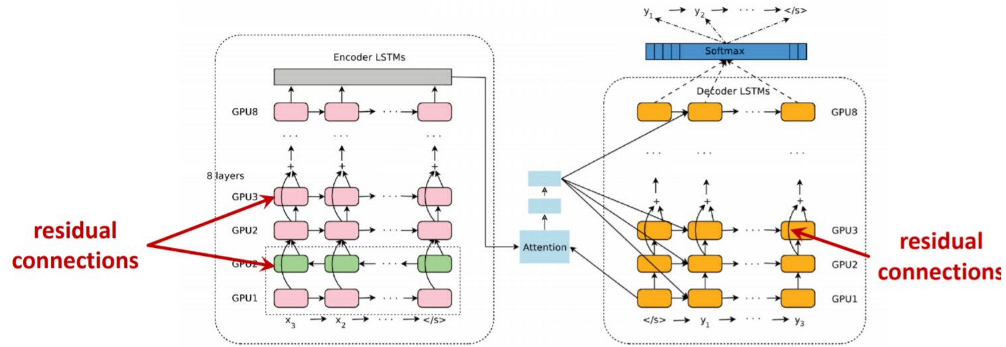


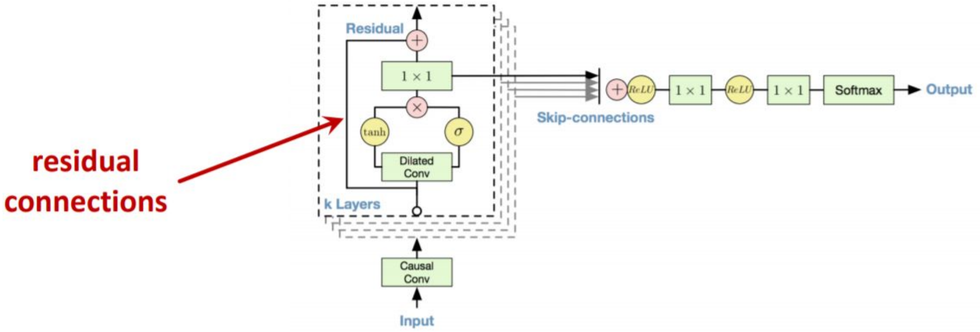
From classification to general vision tasks

https://gkioxari.github.io/Tutorials/eccv2018/index.html

From vision to general machine learning tasks

- Visual Recognition
- Image Generation
- Natural Language Processing
- Speech Recognition
- Advertising, User Prediction
- AlphaGo Zero: 40 Residual Blocks





Conclusions

- Residual Learning!
  - Degradation problem is addressed!
  - Deeper networks are easier to train via residual learning
  - Less error when increasing network depth
  - More accuracy gained from depth!
- Experiments
  - Image Recognition (ImageNet, CIFAR-10)
  - Object Detection (MS COCO)
- Model and Code:
  - https://github.com/KaimingHe/deep-residual-networks

Reference

"Very Deep Convolutional Networks for Large-Scale Image Recognition", Simonyan & Zisserman. ICLR 2015

"Deep Residual Learning for Image Recognition", Kaiming He et al. CVPR 2016

"Identity Mappings in Deep Residual Networks", Kaiming He et al ECCV 2016

"Deep Networks with Stochastic Depth", G. Huang et al, ECCV 2016

"Residual Networks are Exponential Ensembles of Relatively Shallow Networks" , Andreas Veit et al, NIPS 2016

"Learning Deep Representations for Visual Recognition", Kaiming He, Tutorial on Visual Recognition and Beyond, ECCV 2018

Any Questions?