Deep Residual Learning for Image Recognition

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

work done at
Microsoft Research Asia
ResNet @ ILSVRC & COCO 2015 Competitions

1st places in all five main tracks

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

*improvements are relative numbers

Revolution of Depth

ILSVRC'15 ResNet
ILSVRC'14 GoogleNet
ILSVRC'14 VGG
ILSVRC'13
ILSVRC'12 AlexNet
ILSVRC'11
ILSVRC'10

ImageNet Classification top-5 error (%)

Revolution of Depth

Engines of visual recognition

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

- HOG, DPM: shallow
- AlexNet (RCNN): 8 layers, 58
- VGG (RCNN): 16 layers, 66
- ResNet (Faster RCNN)*: 101 layers, 86

*with other improvements & more data

Revolution of Depth

AlexNet, 8 layers (ILSVRC 2012)

11x11 conv, 96, /4, pool/2
5x5 conv, 256, pool/2
3x3 conv, 384
3x3 conv, 384
3x3 conv, 256, pool/2
fc, 4096
fc, 4096
fc, 1000

Revolution of Depth

AlexNet, 8 layers  
(ILSVRC 2012)

VGG, 19 layers  
(ILSVRC 2014)

GoogleNet, 22 layers  
(ILSVRC 2014)

Revolution of Depth

AlexNet, 8 layers (ILSVRC 2012)

VGG, 19 layers (ILSVRC 2014)

ResNet, 152 layers (ILSVRC 2015)

Is learning better networks as simple as stacking more layers?

Simply stacking layers?

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

Simply stacking layers?

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


- 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...

a shallower model (18 layers)

a deeper counterpart (34 layers)
Deep Residual Learning

- Plaint net

\[ H(x) \] is any desired mapping, hope the 2 weight layers fit \( H(x) \)

Any two stacked layers

Deep Residual Learning

• Residual net

\[ H(x) \text{ is any desired mapping,} \]
\[ \text{hope the 2 weight layers fit } H(x) \]
\[ \text{hope the 2 weight layers fit } F(x) \]
\[ \text{let } H(x) = F(x) + x \]

Deep Residual Learning

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

![Diagram](https://example.com/diagram.png)

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

$$H(x) = F(x) + x$$
Network “Design”

- Keep it simple

- Our basic design (VGG-style)
  - all 3x3 conv (almost)
  - spatial size /2 => # filters x2
  - Simple design; just deep!
CIFAR-10 experiments

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

ImageNet experiments

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

ImageNet experiments

- Deeper ResNets have lower error

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-152</td>
<td>5.7</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>6.1</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>6.7</td>
</tr>
<tr>
<td>ResNet-34</td>
<td>7.4</td>
</tr>
</tbody>
</table>

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

Beyond classification

A treasure from ImageNet is on learning features.

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

<table>
<thead>
<tr>
<th>task</th>
<th>2nd-place winner</th>
<th>ResNets</th>
<th>margin (relative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageNet Localization (top-5 error)</td>
<td>12.0</td>
<td>9.0</td>
<td>27%</td>
</tr>
<tr>
<td>ImageNet Detection (mAP@.5)</td>
<td>53.6</td>
<td>62.1</td>
<td><strong>16%</strong></td>
</tr>
<tr>
<td>COCO Detection (mAP@.5:.95)</td>
<td>33.5</td>
<td>37.3</td>
<td><strong>11%</strong></td>
</tr>
<tr>
<td>COCO Segmentation (mAP@.5:.95)</td>
<td>25.1</td>
<td>28.2</td>
<td><strong>12%</strong></td>
</tr>
</tbody>
</table>

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

Object Detection (brief)

- Simply “Faster R-CNN + ResNet”

<table>
<thead>
<tr>
<th>Faster R-CNN baseline</th>
<th>mAP@.5</th>
<th>mAP@.5:.95</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-16</td>
<td>41.5</td>
<td>21.5</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>48.4</td>
<td>27.2</td>
</tr>
</tbody>
</table>

COCO detection results
(ResNet has 28% relative gain)

Our results on MS COCO

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


this video is available online: https://youtu.be/WZmSMkK9VuA
More Visual Recognition Tasks

ResNets lead on these benchmarks (incomplete list):

- **ImageNet** classification, detection, localization
- **MS COCO** detection, segmentation
- **PASCAL VOC** detection, segmentation
- **VQA** challenge 2016

- Depth estimation [Laina et al 2016]
- Segment proposal [Pinheiro et al 2016]

...
Potential 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 (preliminary results)
- Advertising, user prediction (preliminary results)

Conclusions

• Deep Residual Networks:
  • Easy to train
  • Simply gain accuracy from depth
  • Well transferrable

• Follow-up [He et al. arXiv 2016]
  • 200 layers on ImageNet, 1000 layers on CIFAR

Resources

• Models and Code
  • Our ImageNet models in Caffe: https://github.com/KaimingHe/deep-residual-networks

• Many available implementations:
  (list in 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
    • .......