Deep Residual Learning

MSRA @ ILSVRC & COCO 2015 competitions

Kaiming He
with Xiangyu Zhang, Shaoqing Ren, Jifeng Dai, & Jian Sun
Microsoft Research Asia (MSRA)
Presented by : Xiaoyun Wang
MSRA @ ILSVRC & COCO 2015 Competitions

• **1st places in all five main tracks**
  - ImageNet Classification: “Ultra-deep” (quote Yann) **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
3.57
152 layers

ILSVRC'14
GoogleNet
6.7
22 layers

ILSVRC'14
VGG
7.3
19 layers

ILSVRC'13
8 layers
11.7

ILSVRC'12
AlexNet
8 layers
16.4

ILSVRC'11
shallow
25.8

ILSVRC'10
28.2

ImageNet Classification top-5 error (%)

Revolution of Depth

Engines of visual recognition

<table>
<thead>
<tr>
<th>Model</th>
<th>Layers</th>
<th>mAP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOG, DPM</td>
<td>shallow</td>
<td>34</td>
</tr>
<tr>
<td>AlexNet (RCNN)</td>
<td>8</td>
<td>58</td>
</tr>
<tr>
<td>VGG (RCNN)</td>
<td>16</td>
<td>66</td>
</tr>
<tr>
<td>ResNet (Faster RCNN)*</td>
<td>101</td>
<td>86</td>
</tr>
</tbody>
</table>

*with other improvements & more data


PASCAL VOC 2007 **Object Detection** mAP (%)
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)  
ResNet, 152 layers (ILSVRC 2015)

Revolution of Depth

ResNet, 152 layers

Revolution of Depth

ResNet, 152 layers

Revolution of Depth

ResNet, **152 layers**

Revolution of Depth

ResNet, 152 layers

(there was an animation here)
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

A shallower model
(18 layers)

A deeper counterpart
(34 layers)

- 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...

Deep Residual Learning

- Plain net

\[ x \rightarrow \text{weight layer} \rightarrow \text{relu} \rightarrow \text{weight layer} \rightarrow \text{relu} \rightarrow H(x) \]

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

Deep Residual Learning

- Residual net

\[ H(x) = F(x) + x \]

\[ F(x) \]

weight layer

relu

identity

weight layer

relu

\[ x \]

\[ H(x) \]

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

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

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

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

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
  • Simple design; just deep!

• Other remarks:
  • no max pooling (almost)
  • no hidden fc
  • no dropout

Training

• All plain/residual nets are trained from scratch

• All plain/residual nets use Batch Normalization

• Standard hyper-parameters & augmentation

CIFAR-10 experiments

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

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

ImageNet experiments

• A practical design of going deeper

ImageNet experiments

- Deeper ResNets have lower error

<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 (%)

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

ImageNet experiments

ImageNet Classification top-5 error (%)

ILSVRC'15 ResNet: 3.57
ILSVRC'14 GoogleNet: 6.7
ILSVRC'14 VGG: 7.3
ILSVRC'13: 11.7 (8 layers)
ILSVRC'12 AlexNet: 16.4 (8 layers)
ILSVRC'11: 25.8
ILSVRC'10: 28.2 (shallow)

152 layers

Just 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>MSRA</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 absolute 8.5% better!</td>
<td>62.1</td>
<td>16%</td>
</tr>
<tr>
<td>COCO Detection (mAP@.5:.95)</td>
<td>33.5</td>
<td>37.3</td>
<td>11%</td>
</tr>
<tr>
<td>COCO Segmentation (mAP@.5:.95)</td>
<td>25.1</td>
<td>28.2</td>
<td>12%</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@0.5</th>
<th>mAP@0.5:0.95</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)

Object Detection (brief)

• RPN learns proposals by extremely deep nets
  • We use only 300 proposals (no SS/EB/MCG!)

• Add what is just missing in Faster R-CNN...
  • Iterative localization
  • Context modeling
  • Multi-scale testing

• All are based on CNN features; all are end-to-end (train and/or inference)

• All benefit more from deeper features – cumulative gains!

Our results on COCO – too many objects, let’s check carefully!


*the original image is from the COCO dataset*


*the original image is from the COCO dataset*
Instance Segmentation (brief)

- Solely CNN-based ("features matter")
- Differentiable RoI warping layer (w.r.t box coord.)
- Multi-task cascades, exact end-to-end training


*the original image is from the COCO dataset*
Conclusions

• Deeper is still better

• “Features matter”!

• Faster R-CNN is just amazing