DEEP LEARNING

Lecture 6: CNN Architectures

Dr. Yang Lu

Department of Computer Science and Technology

luyang@xmu.edu.cn

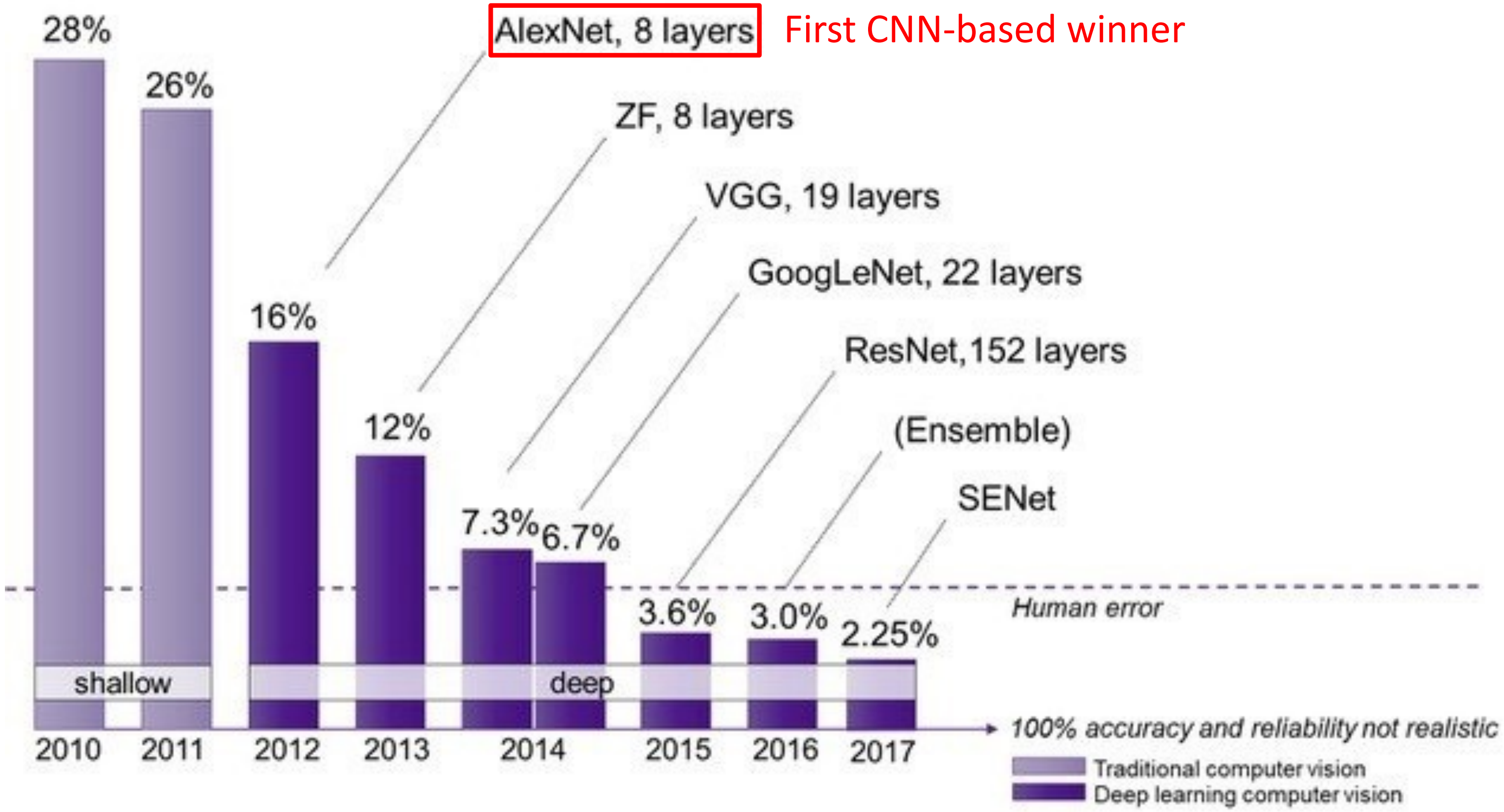
CNN Architectures

- AlexNet
- VGG
- GoogLeNet
- ResNet
- SENet

ALEXNET

ILSVRC Winners

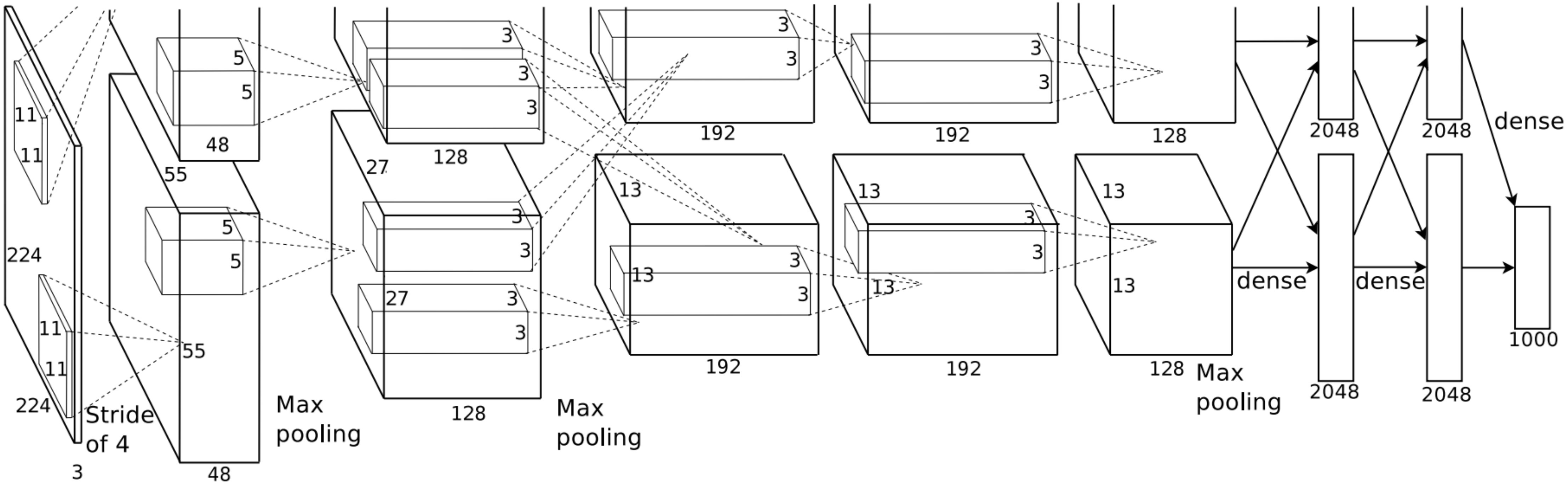
Imagenet classification with **deep** convolutional neural networks
A Krizhevsky, I Sutskever… - Advances in neural …, 2012 - proceedings.neurips.cc
… We trained a large, **deep** convolutional neural network to **classify** the 1.2 million high-resolution images in the **ImageNet** LSVRC-2010 contest into the 1000 different classes. On the test …
Save Cite Cited by 121478 Related articles All 111 versions



First CNN-based winner

Image source: https://semiengineering.com/new-vision-technologies-for-real-world-applications/

AlexNet



CONV1 POOL1 CONV2 POOL2 CONV3 CONV4 CONV5 POOL3 FC6 FC7 FC8

Image source: Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In Advances in neural information processing systems, pp. 1097-1105. 2012.

AlexNet

$$\left\lfloor \frac{n_h + 2p_h - k_h}{s_h} + 1 \right\rfloor \times \left\lfloor \frac{n_w + 2p_w - k_w}{s_w} + 1 \right\rfloor \times c_{out}$$

Input: 227x227x3 images

- First layer (CONV1): 96 filters with size 11x11x3 with a stride of 4.
  - Output height and width: (227-11)/4+1=55.
  - Output volume size: 55x55x96.
- The actual output is not a feature map with size 55x55x96, but two feature maps with size 55x55x48 in different GPUs.
- Total number of parameters: 2x11x11x3x48=35K.

A mistake, should be 227

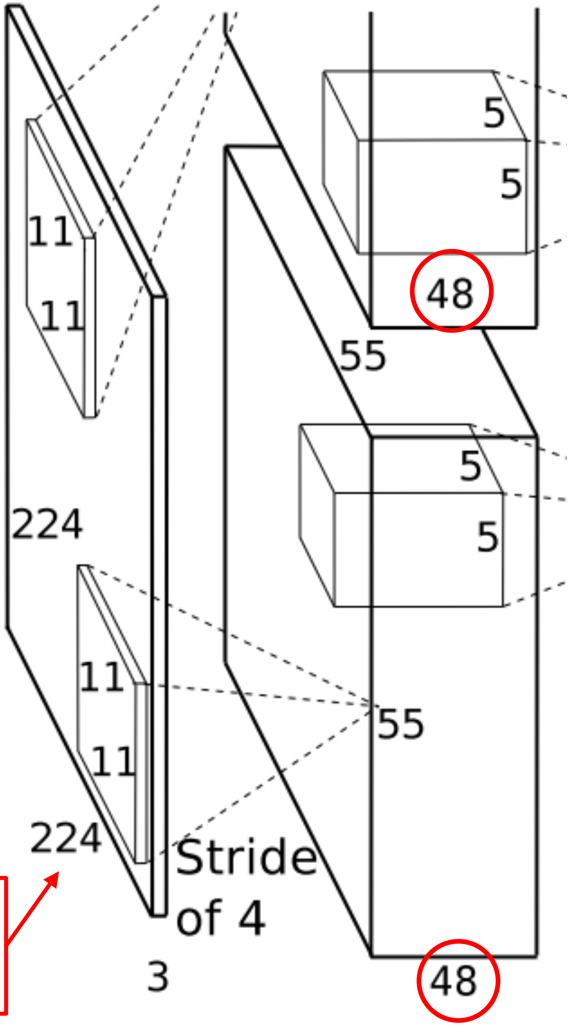


Image source: Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In Advances in neural information processing systems, pp. 1097-1105. 2012.

AlexNet

$$\left\lfloor \frac{n_h + 2p_h - k_h}{s_h} + 1 \right\rfloor \times \left\lfloor \frac{n_w + 2p_w - k_w}{s_w} + 1 \right\rfloor \times c_{out}$$

Input: 227x227x3 images.

After CONV1: 2x55x55x48.

- Second layer (POOL1): 3x3 filter with a stride of 2.
  - Output height and width: (55-3)/2+1=27.
  - Output volume size: 2x27x27x48.

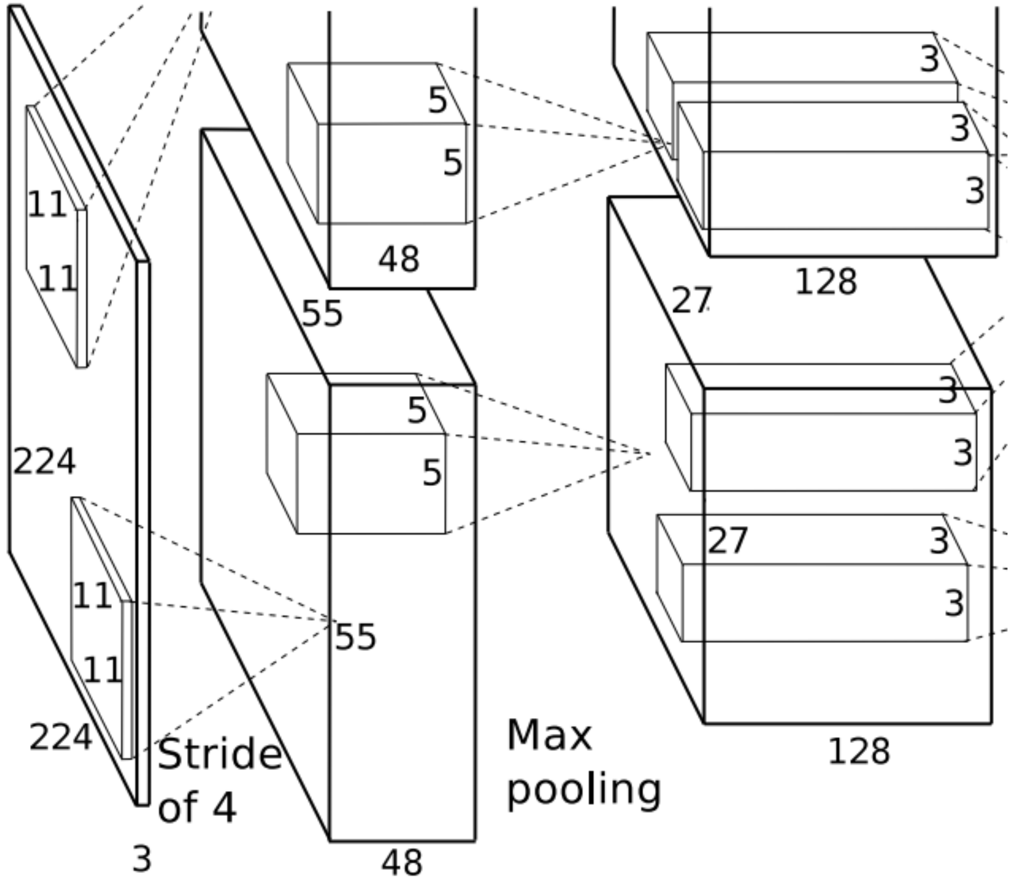
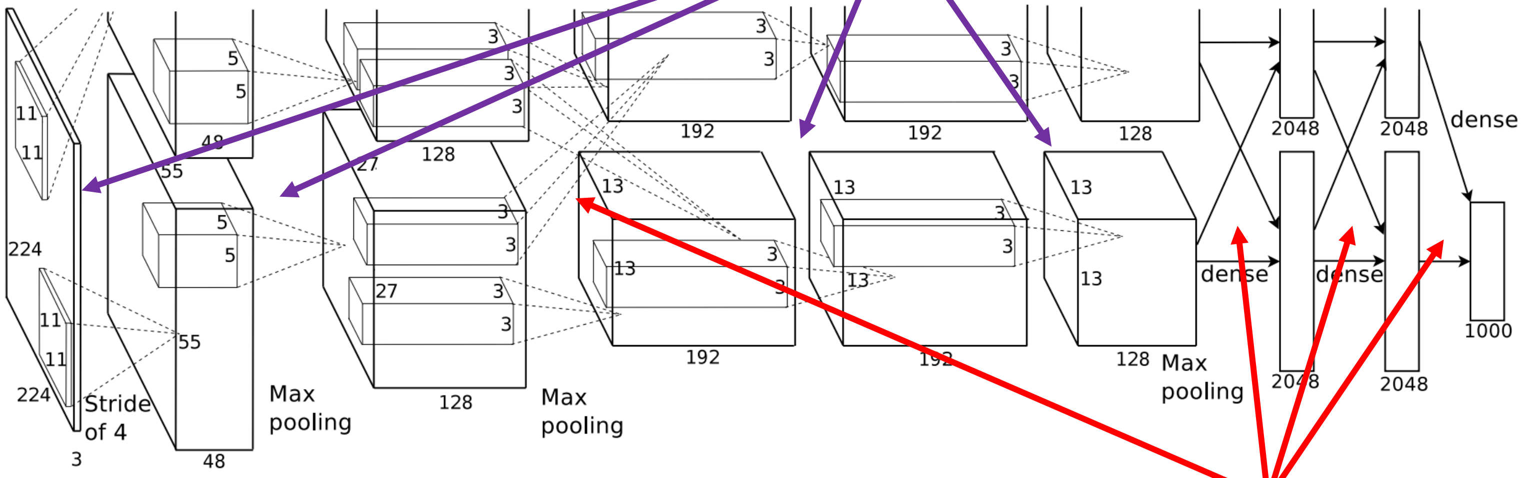


Image source: Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In Advances in neural information processing systems, pp. 1097-1105. 2012.

AlexNet

CONV1, CONV2, CONV4, CONV5 connect only with feature maps on same GPU



CONV3, FC6, FC7, FC8 connect across GPUs

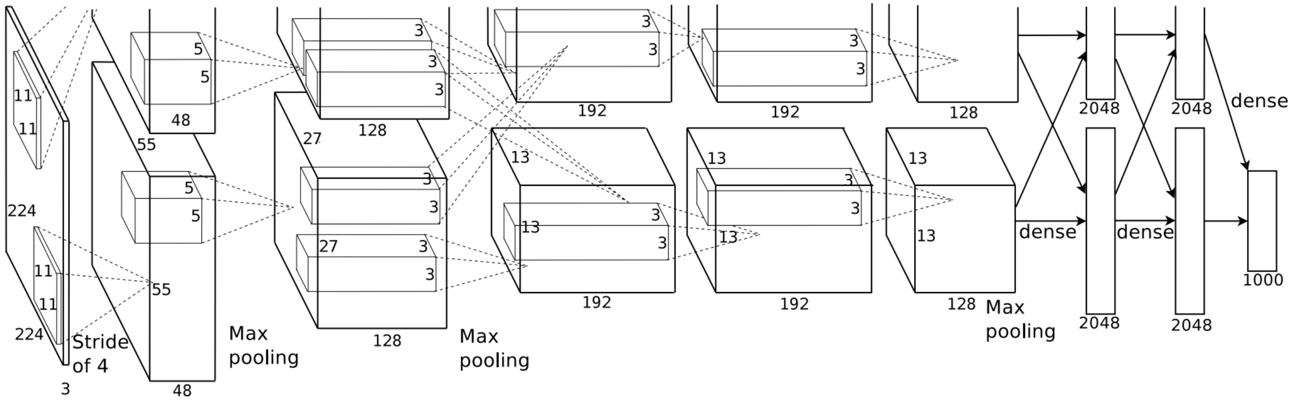
Input: 227x227x3 images.

After CONV1: 2x55x55x48. After POOL1: 2x27x27x48.

- Third layer (CONV2): 256 filters with size 5x5x48 with stride 1 and padding 2.
  - Output height and width: (27+2x2-5)/1+1=27.
  - Output volume size: 2x27x27x128.
- Total number of parameters: 5x5x48x128x2=307K.

Image source: Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In Advances in neural information processing systems, pp. 1097-1105. 2012.

AlexNet



AlexNet architecture:

- [227x227x3] INPUT.
- [2x55x55x48] CONV1: 96 filters with size 11x11x3, stride 4, padding 0.
- [2x27x27x48] POOL1: 3x3 filters with stride 2.
- [2x27x27x128] CONV2: 256 filters with size 5x5x48, stride 1, padding 2.
- [2x13x13x128] POOL2: 3x3 filters with stride 2.
- [2x13x13x192] CONV3: 384 filters with size 3x3x256, stride 1, padding 1.
- [2x13x13x192] CONV4: 384 filters with size 3x3x192, stride 1, padding 1.
- [2x13x13x128] CONV5: 256 filters with size 3x3x192, stride 1, padding 1.
- [2x6x6x128] POOL3: 3x3 filters with stride 2.
- [4096] FC6: 4096 neurons.
- [4096] FC7: 4096 neurons.
- [1000] FC8: 1000 neurons (class scores).

Details:

- First use of ReLU
- Heavy data augmentation
- Dropout rate 0.5
- Batch size 128.
- SGD with momentum 0.9.
- Learning rate 1e-2, reduced by 10 manually when the validation error rate stopped improving with the current learning rate.
- L2 weight decay 5e-4.

Image source: Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In Advances in neural information processing systems, pp. 1097-1105. 2012.

ReLU vs. Sigmoid

- A large change in the input of the sigmoid function will cause a small change in the output.
- Hence, the derivative becomes small. This phenomenon is called vanishing gradient problem.

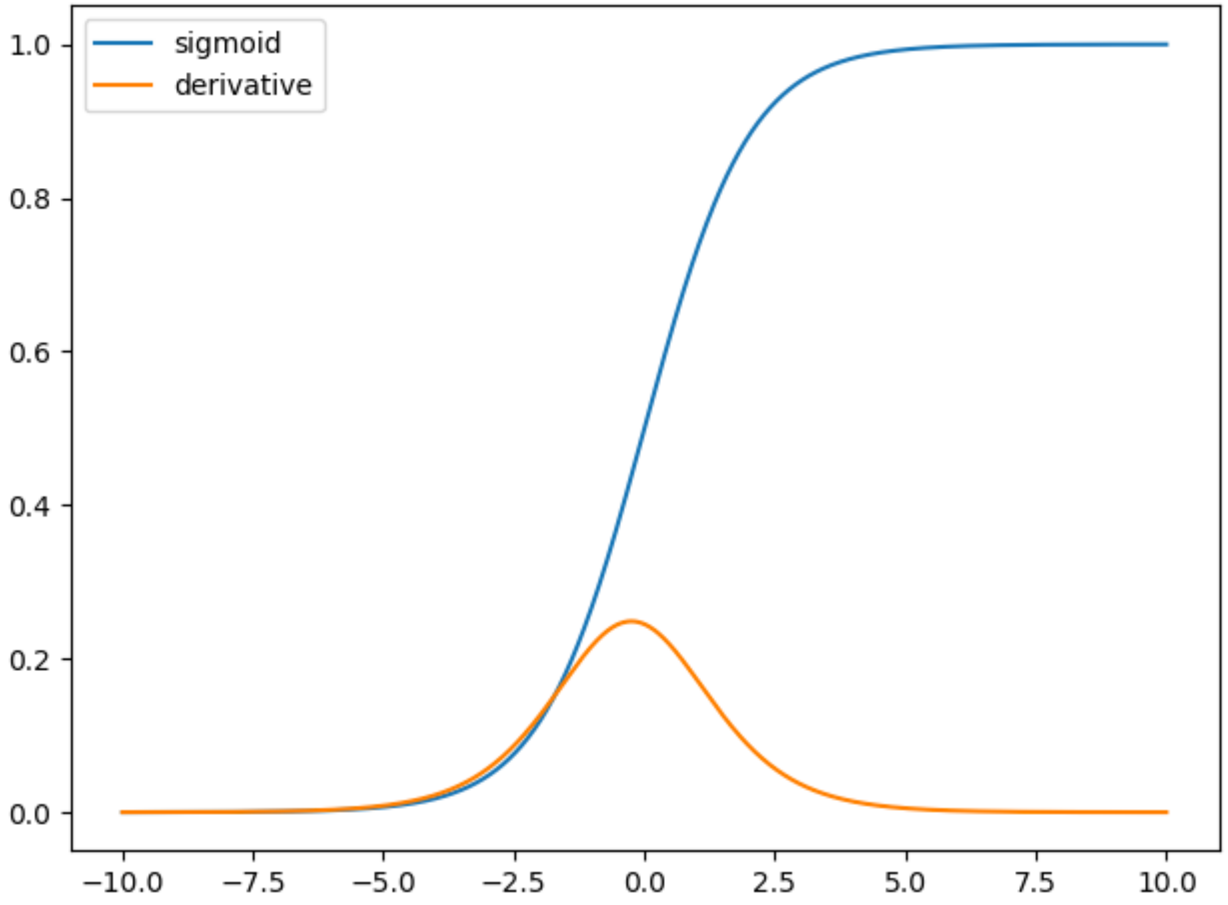


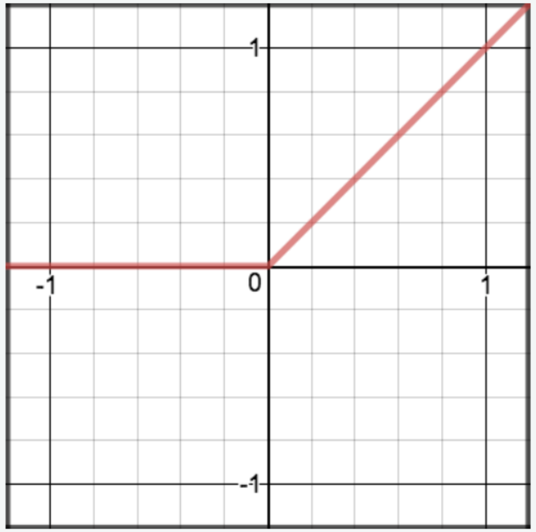
Image source: https://datascience.stackexchange.com/questions/30676/role-derivative-of-sigmoid-function-in-neural-networks

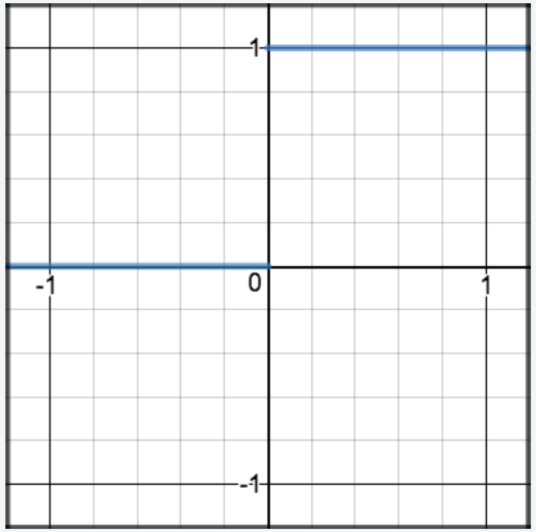
ReLU vs. Sigmoid

Pros

- Less computationally expensive than tanh and sigmoid.
- It converges faster. Linearity means that the slope doesn't plateau, thus solves the vanishing gradient problem.
- It's sparsely activated. Since ReLU is zero for all negative inputs, it's likely for any given unit to not activate at all.

Function	Derivative
$R(z) = \begin{cases} z & z > 0 \\ 0 & z <= 0 \end{cases}$	$R'(z) = \begin{cases} 1 & z > 0 \\ 0 & z < 0 \end{cases}$






```
def relu(z):
  return max(0, z)
```

```
def relu_prime(z):
  return 1 if z > 0 else 0
```

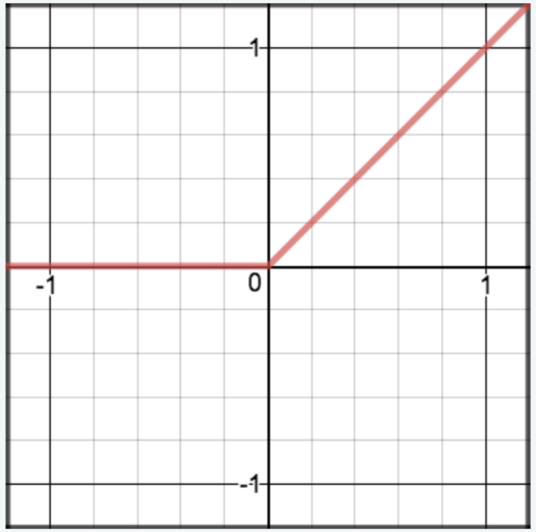

Image source: https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html

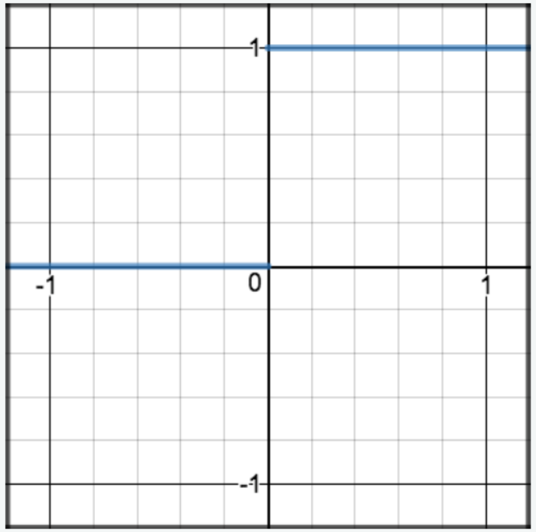
ReLU vs. Sigmoid

Cons

- The "dying ReLU" problem: it results in dead neurons.
- The range of ReLU is [0, inf). This means it can blow up the activation.

Function	Derivative
$R(z) = \begin{Bmatrix} z & z > 0 \\ 0 & z <= 0 \end{Bmatrix}$	$R'(z) = \begin{Bmatrix} 1 & z > 0 \\ 0 & z < 0 \end{Bmatrix}$






```
def relu(z):
  return max(0, z)
```

```
def relu_prime(z):
  return 1 if z > 0 else 0
```


Image source: https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html

Variants of ReLU

Leaky ReLU & Parametric ReLU (PReLU)

- It fixes the "dying ReLU" problem, as it doesn't have zero-slope parts.
- Leaky ReLU isn't always superior to plain ReLU, and should be considered only as an alternative.

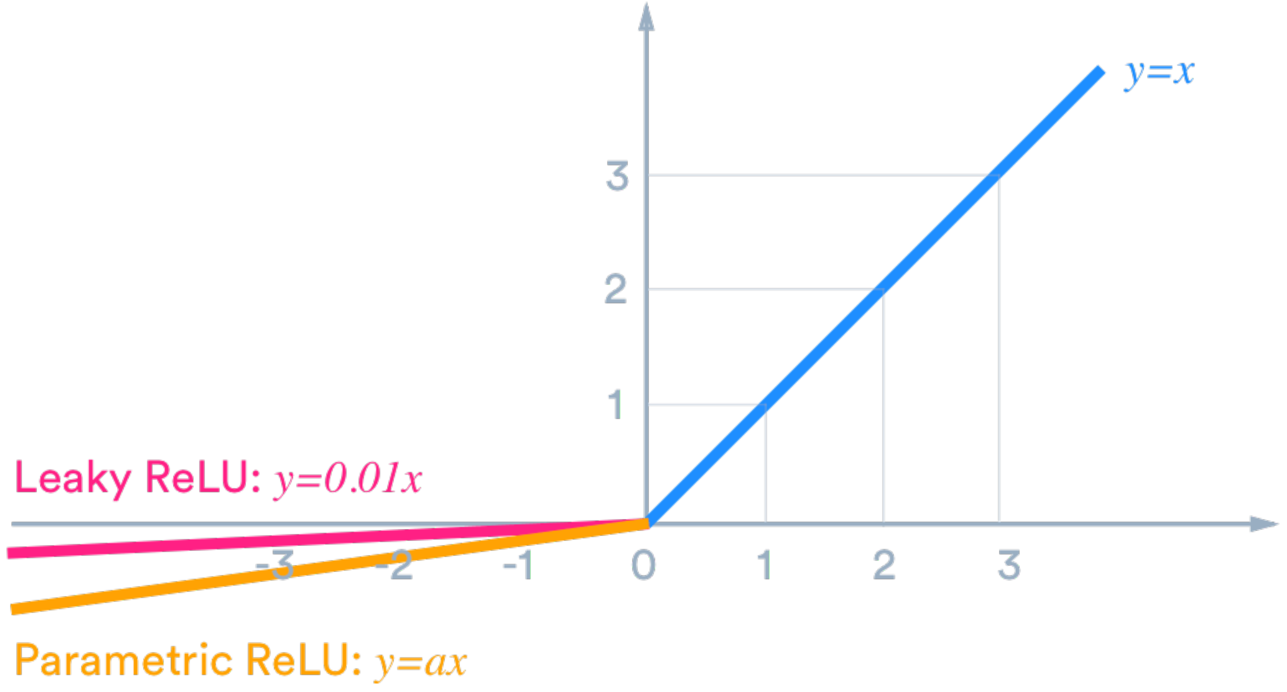


Image source: https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7

Variants of ReLU

Exponential Linear (ELU, SELU)

- Combine the good parts of ReLU and leaky ReLU:
  - it doesn't have the dying ReLU problem;
  - it saturates for large negative values, allowing them to be essentially inactive.

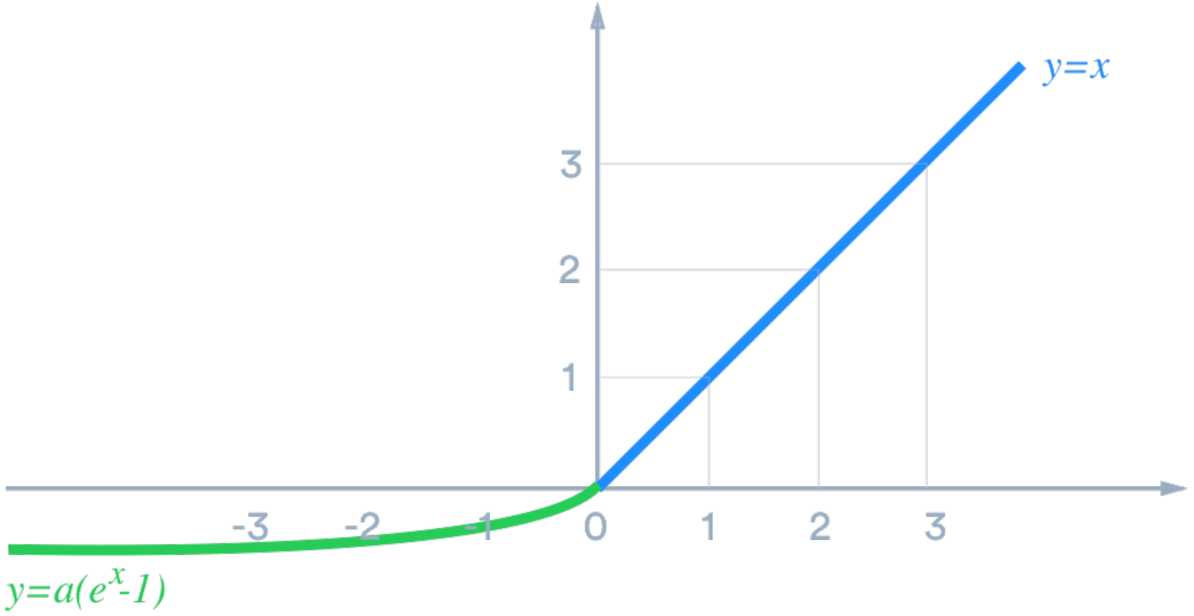


Image source: https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7

ILSVRC Winners

Visualizing and understanding convolutional networks
MD Zeiler, R Fergus - Computer Vision–ECCV 2014: 13th European …, 2014 - Springer
Large **Convolutional Network** models have recently demonstrated impressive classification performance on the ImageNet benchmark Krizhevsky et al. [18]. However there is no clear …
☆ Save 99 Cite Cited by 20521 Related articles All 23 versions

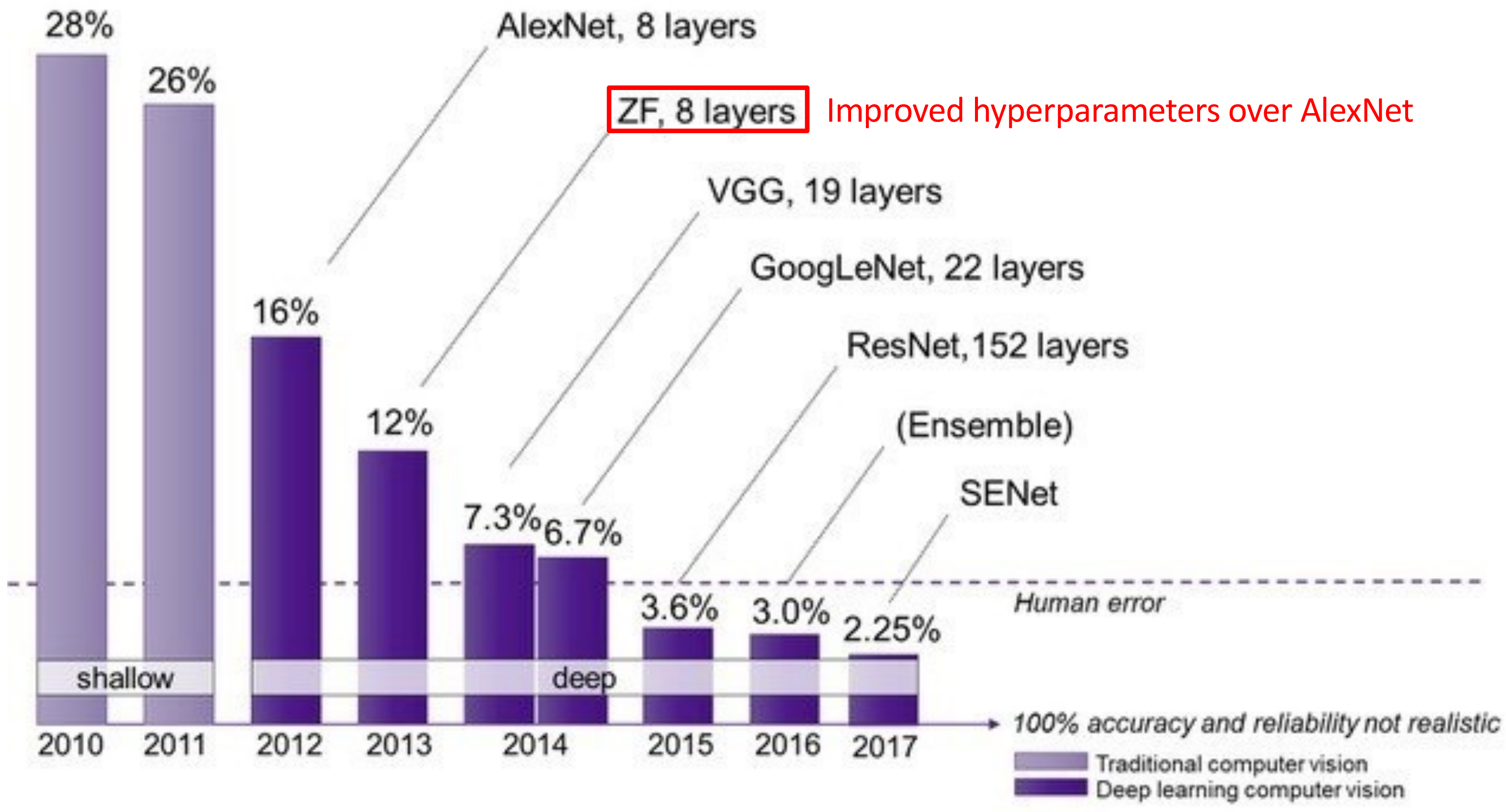
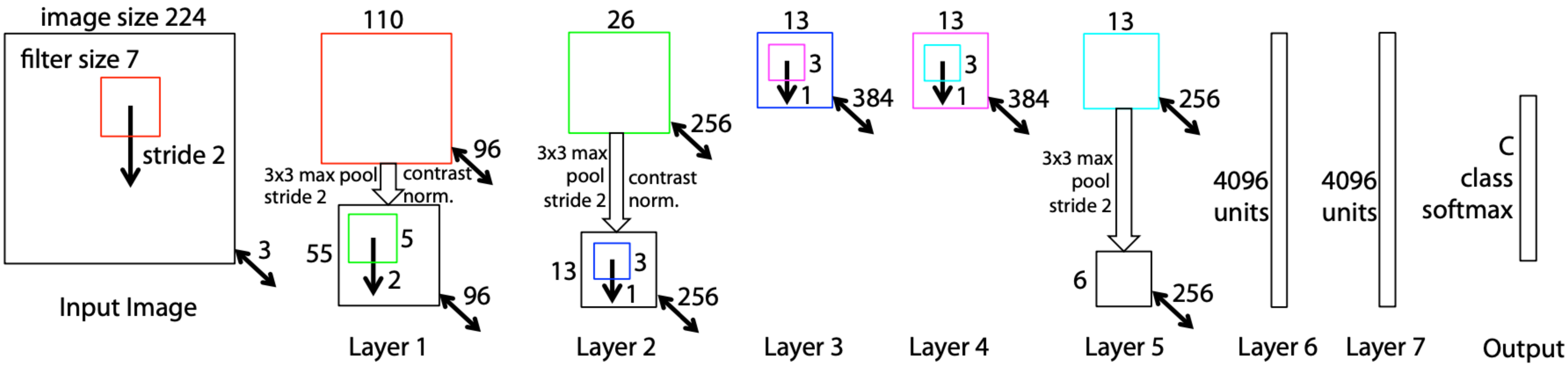


Image source: https://semiengineering.com/new-vision-technologies-for-real-world-applications/

ZFNet



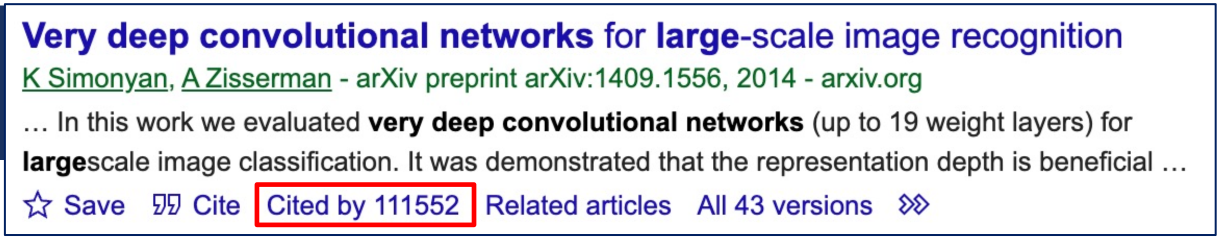
Improve AlexNet by:

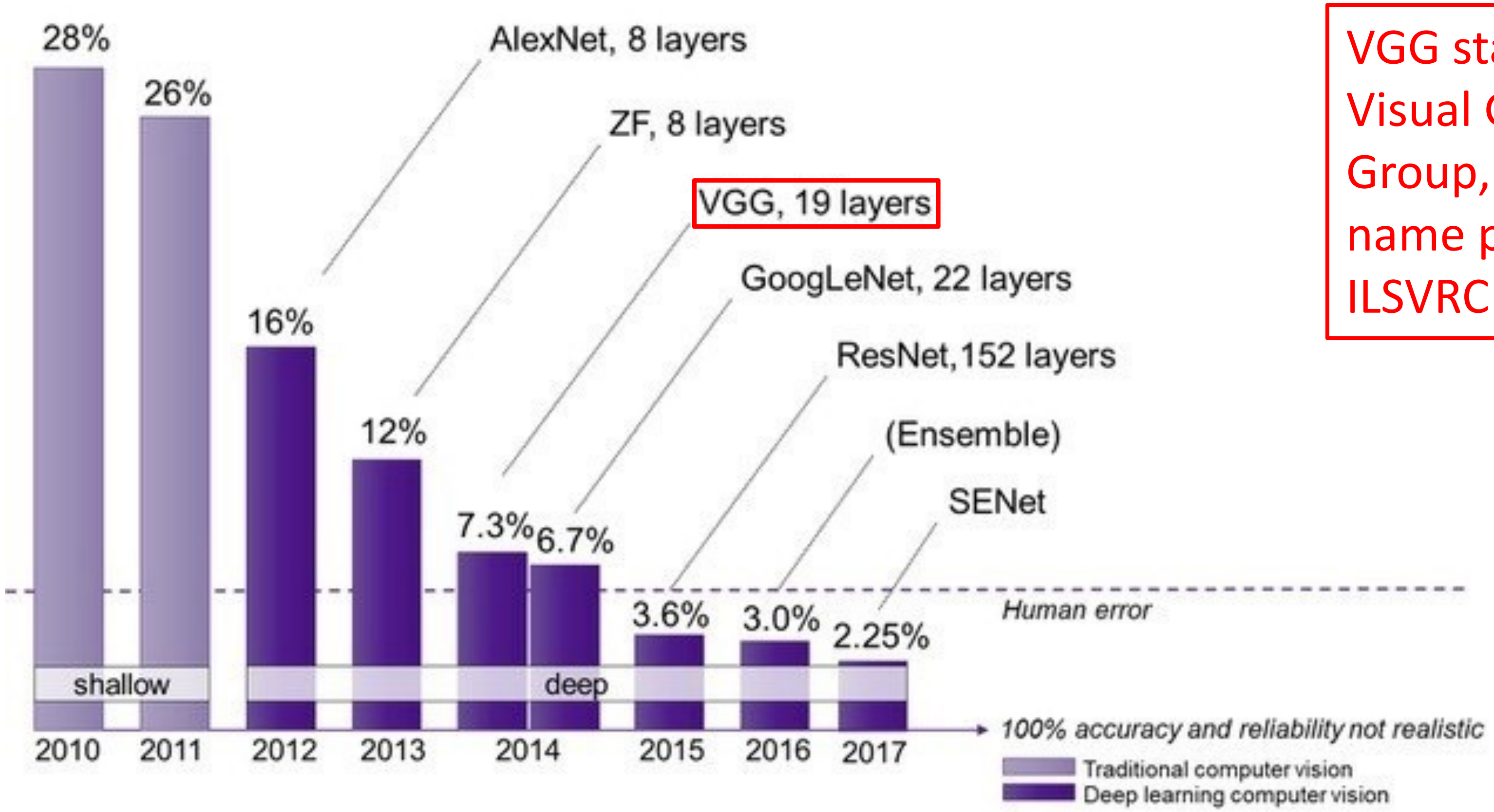
- CONV1: change from (11x11 stride 4) to (7x7 stride 2).
- CONV3,4,5: instead of 384, 384, 256 filters use 512, 1024, 512.

Image source: Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." In European conference on computer vision, pp. 818-833. Springer, Cham, 2014.

VGG

ILSVRC Winners


Very deep convolutional networks for **large**-scale image recognition
K Simonyan, A Zisserman - arXiv preprint arXiv:1409.1556, 2014 - arxiv.org
… In this work we evaluated **very deep convolutional networks** (up to 19 weight layers) for **large**scale image classification. It was demonstrated that the representation depth is beneficial …
Save Cite Cited by 111552 Related articles All 43 versions



VGG stands for Visual Geometry Group, the team name participating ILSVRC 2014.

Image source: https://semiengineering.com/new-vision-technologies-for-real-world-applications/

VGG

Main contribution: small filters, deeper networks.

- Number of layers: 8 (AlexNet & ZFNet) -> 16-19.
- Filter size: 11x11 (AlexNet), 7x7 (ZFNet) -> 3x3 everywhere.
- Fix other setting: Only stride 1, pad 1 and 2x2 MAX POOL stride 2.

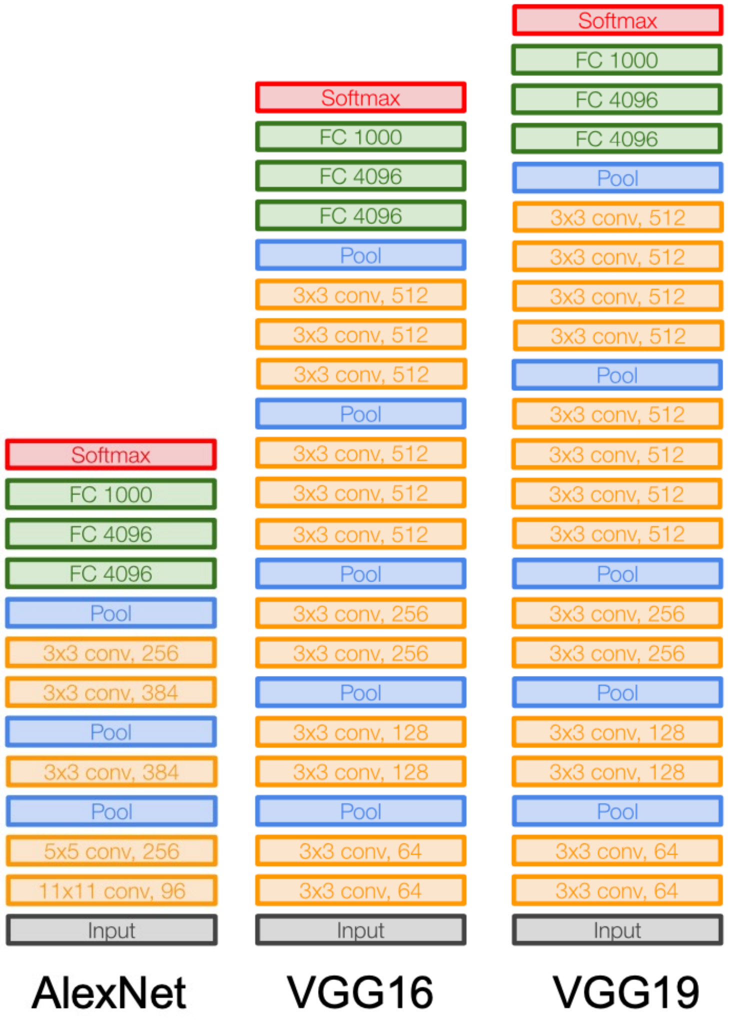


Image source: Lecture 9, cs231n.

VGG

- Stack of three 3x3 conv (stride 1) layers has same effective receptive field as one 7x7 conv layer.
- But deeper, more non-linearities, and fewer parameters.
  - $3\times(3^2C^2) = 27C^2$ vs. $7^2C^2 = 49C^2$ for C channels.

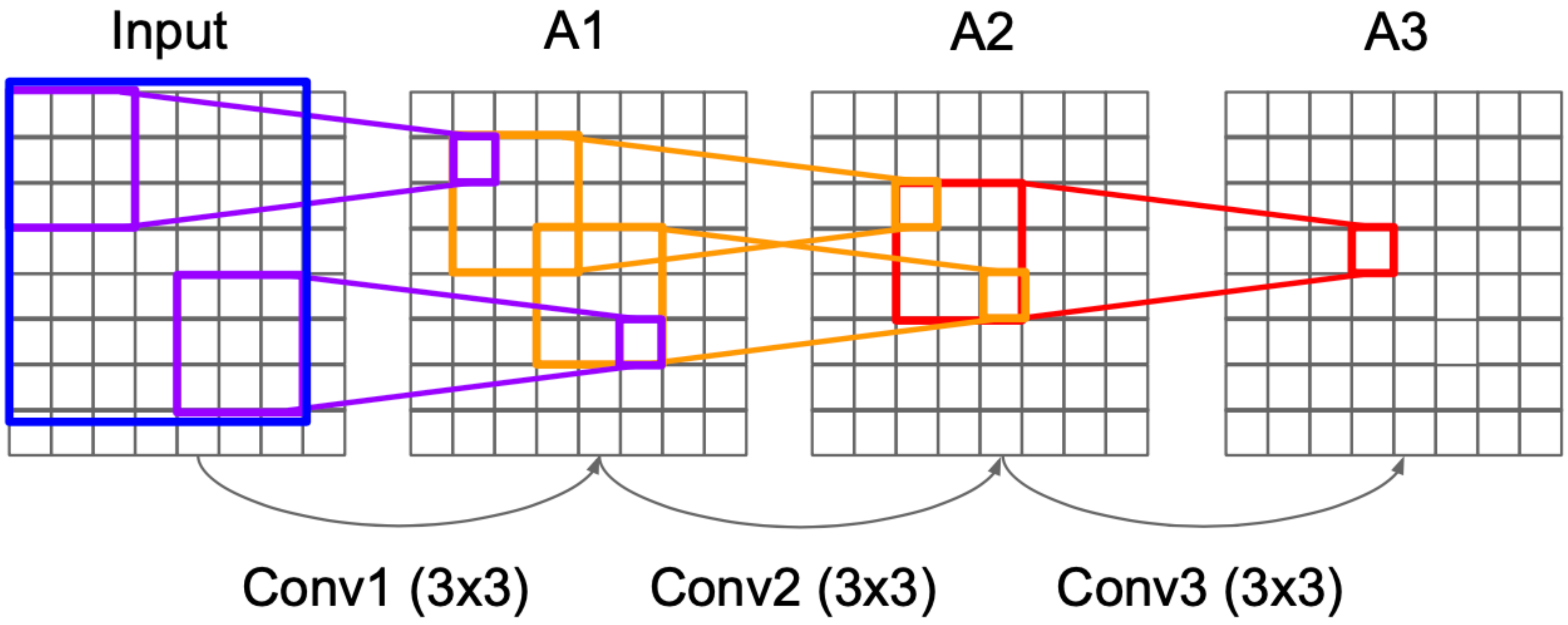


Image source: Lecture 9, cs231n.

VGG

Details:

- ILSVRC’14 2nd in classification, 1st in localization
- Similar training procedure as AlexNet
- No Local Response Normalisation (LRN)
  - “such normalisation does not improve the performance on the ILSVRC dataset, but leads to increased memory consumption and computation time”
- Use VGG16 or VGG19 (VGG19 only slightly better, more memory)
- FC7 features generalize well to other tasks.

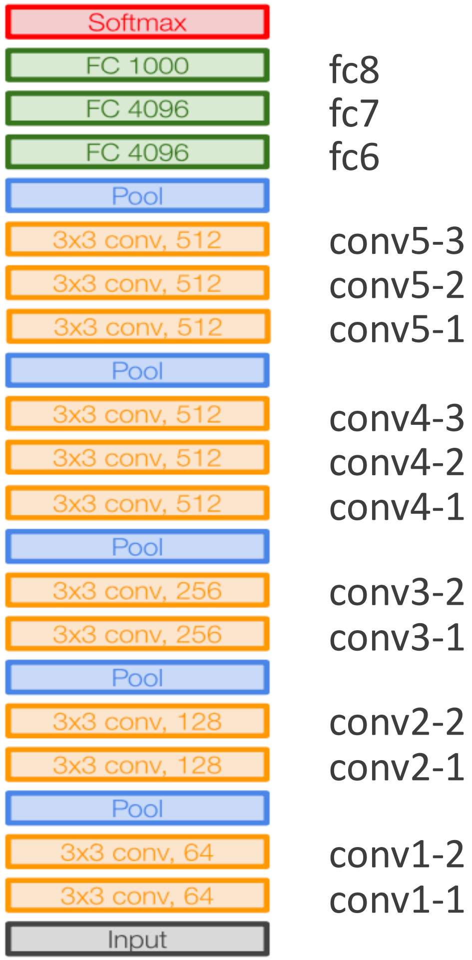
VGG

Most memory is in early CONV.

INPUT: [224x224x3]	memory: 224*224*3=150K	params: 0
CONV1-1: [224x224x64]	memory: 224*224*64=3.2M	params: (3*3*3)*64 = 1,728
CONV1-2: [224x224x64]	memory: 224*224*64=3.2M	params: (3*3*64)*64 = 36,864
POOL1: [112x112x64]	memory: 112*112*64=800K	params: 0
CONV2-1: [112x112x128]	memory: 112*112*128=1.6M	params: (3*3*64)*128 = 73,728
CONV2-2: [112x112x128]	memory: 112*112*128=1.6M	params: (3*3*128)*128 = 147,456
POOL2: [56x56x128]	memory: 56*56*128=400K	params: 0
CONV3-1: [56x56x256]	memory: 56*56*256=800K	params: (3*3*128)*256 = 294,912
CONV3-2: [56x56x256]	memory: 56*56*256=800K	params: (3*3*256)*256 = 589,824
POOL2: [28x28x256]	memory: 28*28*256=200K	params: 0
CONV4-1: [28x28x512]	memory: 28*28*512=400K	params: (3*3*256)*512 = 1,179,648
CONV4-2: [28x28x512]	memory: 28*28*512=400K	params: (3*3*512)*512 = 2,359,296
CONV4-3: [28x28x512]	memory: 28*28*512=400K	params: (3*3*512)*512 = 2,359,296
POOL3: [14x14x512]	memory: 14*14*512=100K	params: 0
CONV5-1: [14x14x512]	memory: 14*14*512=100K	params: (3*3*512)*512 = 2,359,296
CONV5-2: [14x14x512]	memory: 14*14*512=100K	params: (3*3*512)*512 = 2,359,296
CONV5-3: [14x14x512]	memory: 14*14*512=100K	params: (3*3*512)*512 = 2,359,296
POOL4: [7x7x512]	memory: 7*7*512=25K	params: 0
FC6: [1x1x4096]	memory: 4096	params: 7*7*512*4096 = 102,760,448
FC7: [1x1x4096]	memory: 4096	params: 4096*4096 = 16,777,216
FC8: [1x1x1000]	memory: 1000	params: 4096*1000 = 4,096,000

TOTAL memory: 24M * 4 bytes ≈ 96MB / image (for a forward pass)
TOTAL params: 138M parameters

Most params are in late FC.



VGG16

Image source: Lecture 9, cs231n.

GOOGLENET

ILSVRC Winners

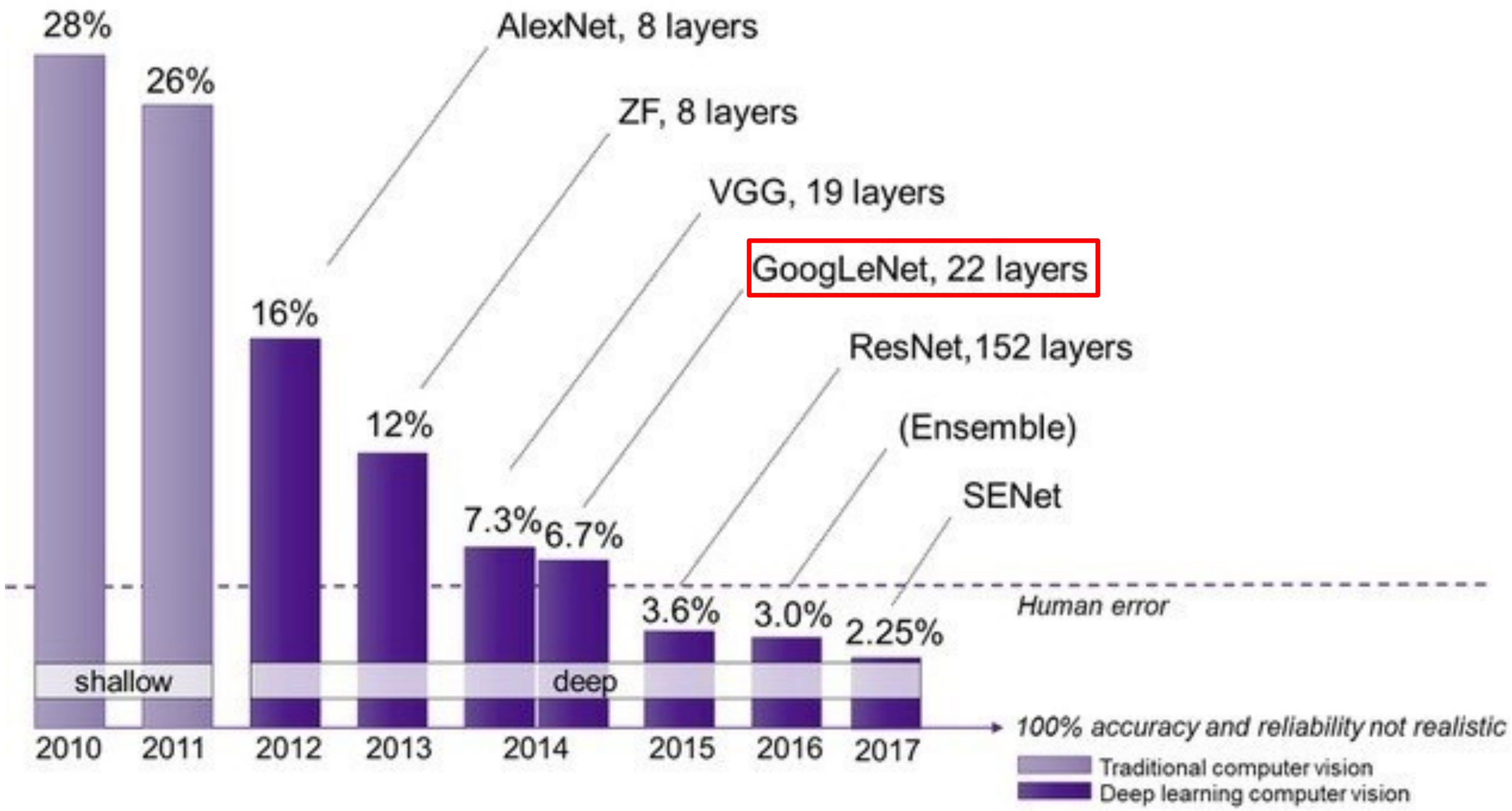


Image source: https://semiengineering.com/new-vision-technologies-for-real-world-applications/

Motivation

- The most straightforward way of improving the performance of deep neural networks is by increasing their size.
  - Depth: the number of network levels.
  - Width: the number of units at each level.
- However, this simple solution comes with two major drawbacks:
  - Bigger size typically means a larger number of parameters, which makes the enlarged network more prone to overfitting.
  - Dramatically increase the use of computational resources.
  - Hard to train.

Sparsity

- Solution: introduce sparsity and replace the fully connected layers by the sparse ones.
- However, filter-level sparsity doesn't work, because our current hardware by utilizing computations on dense matrices.
- Architecture-level sparsity: clustering sparse matrices into relatively dense submatrices tends to give competitive performance for sparse matrix multiplication.

$$\begin{bmatrix} 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 4 & 1 \\ 3 & 2 \end{bmatrix} \Leftrightarrow \frac{\begin{bmatrix} 2 & 3 \\ 0 & 2 \end{bmatrix} \otimes \begin{bmatrix} 4 & 1 \\ 3 & 2 \end{bmatrix}}{\begin{bmatrix} 1 & 2 & 0 \\ 2 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 4 & 1 \\ 3 & 2 \end{bmatrix}}$$

Image source: https://zhuanlan.zhihu.com/p/32702031

Inception

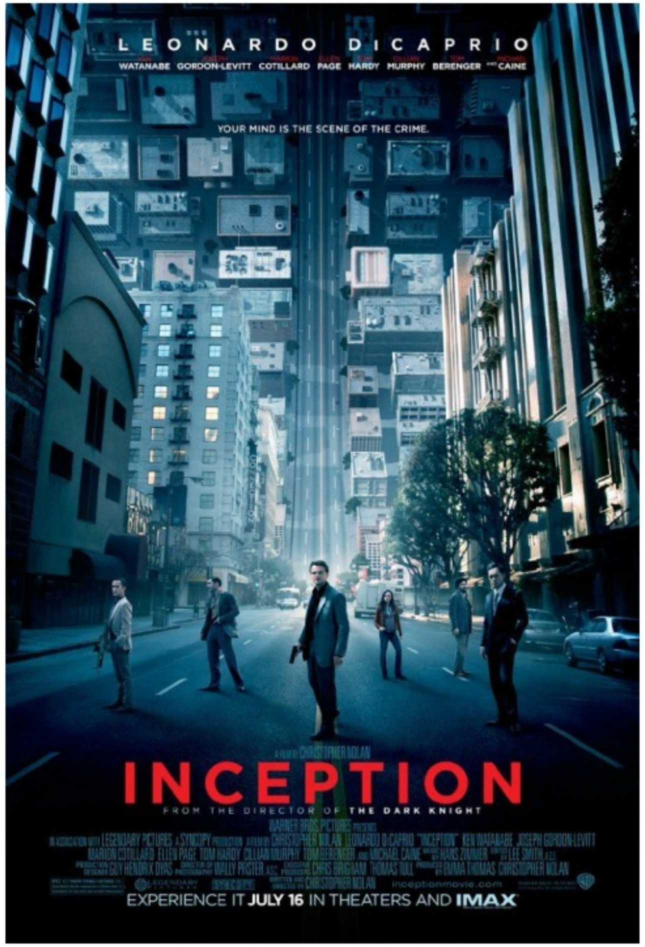
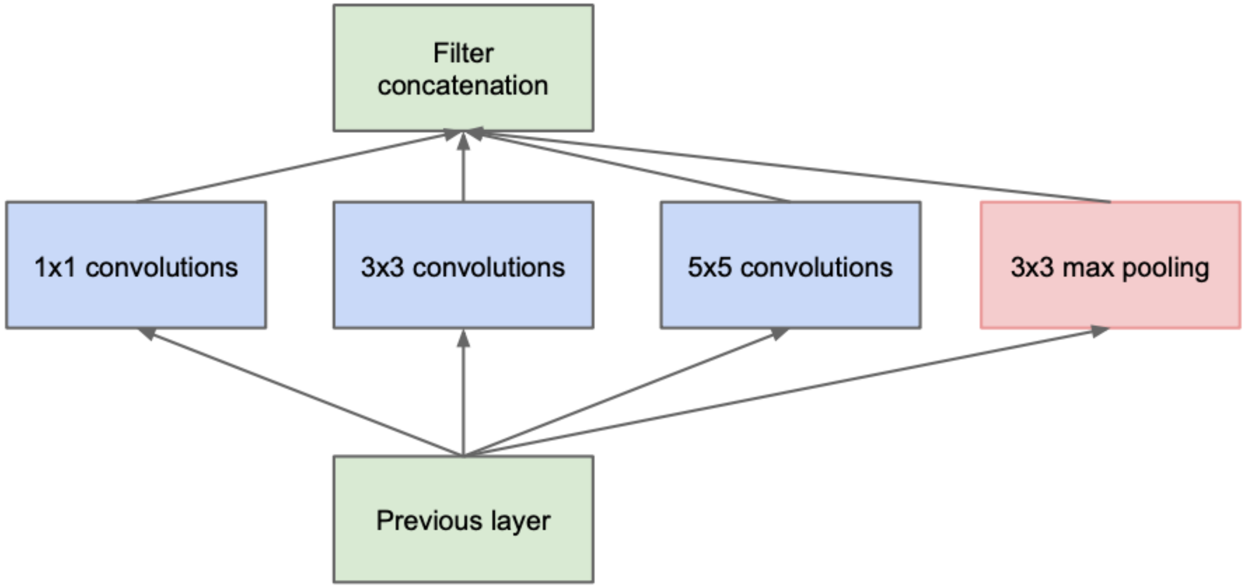


Image source: https://m.media-amazon.com/images/M/MV5BMjAxMzY3NjcxNF5BMl5BanBnXkFtZTcwNTI5OTM0Mw@@._V1_SY1000_CR0,0,675,1 https://knowyourmeme.com/memes/we-need-to-go-deeper

Inception Module

- Apply parallel filter operations on the input from previous layer:
  - Multiple receptive field sizes for convolution (1x1, 3x3, 5x5).
  - Pooling operation (3x3).

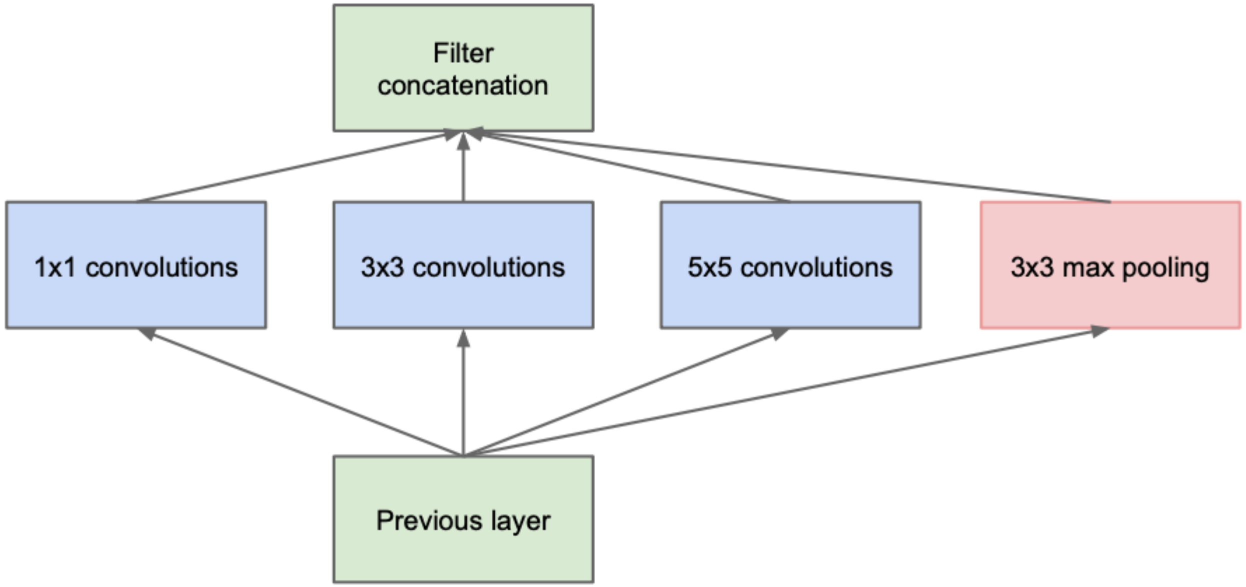


Inception module, naïve version

Image source: Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9. 2015.

Inception Module

- Approximation of an optimal local sparse structure.
- Process visual/spatial information at various scales and then aggregate.
- This is a bit optimistic, computationally.
  - 5x5 convolutions are especially expensive.



Inception module, naïve version

Image source: Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9. 2015.

Inception Module

- Parameters:

  [1x1 conv, 128] 128x1x1x256
  [3x3 conv, 192] 192x3x3x256
  [5x5 conv, 96] 96x5x5x256
  Total: 108k

- Total depth after concat can only grow at every layer!

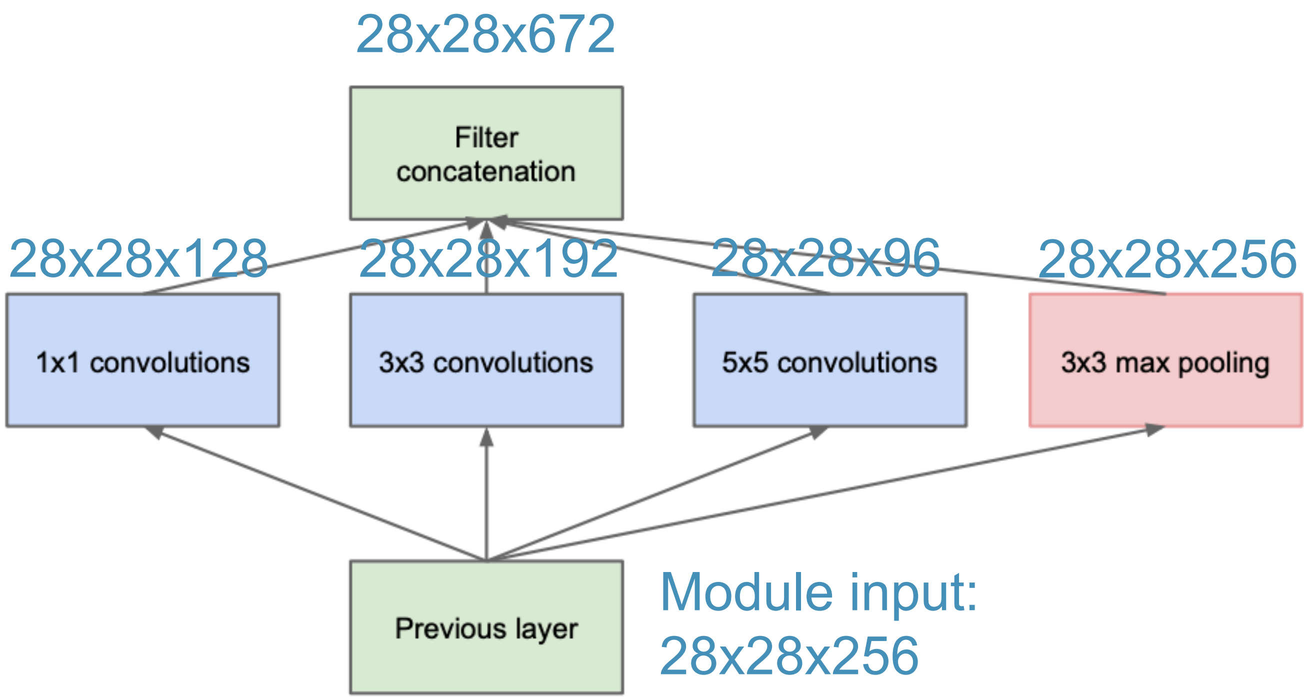
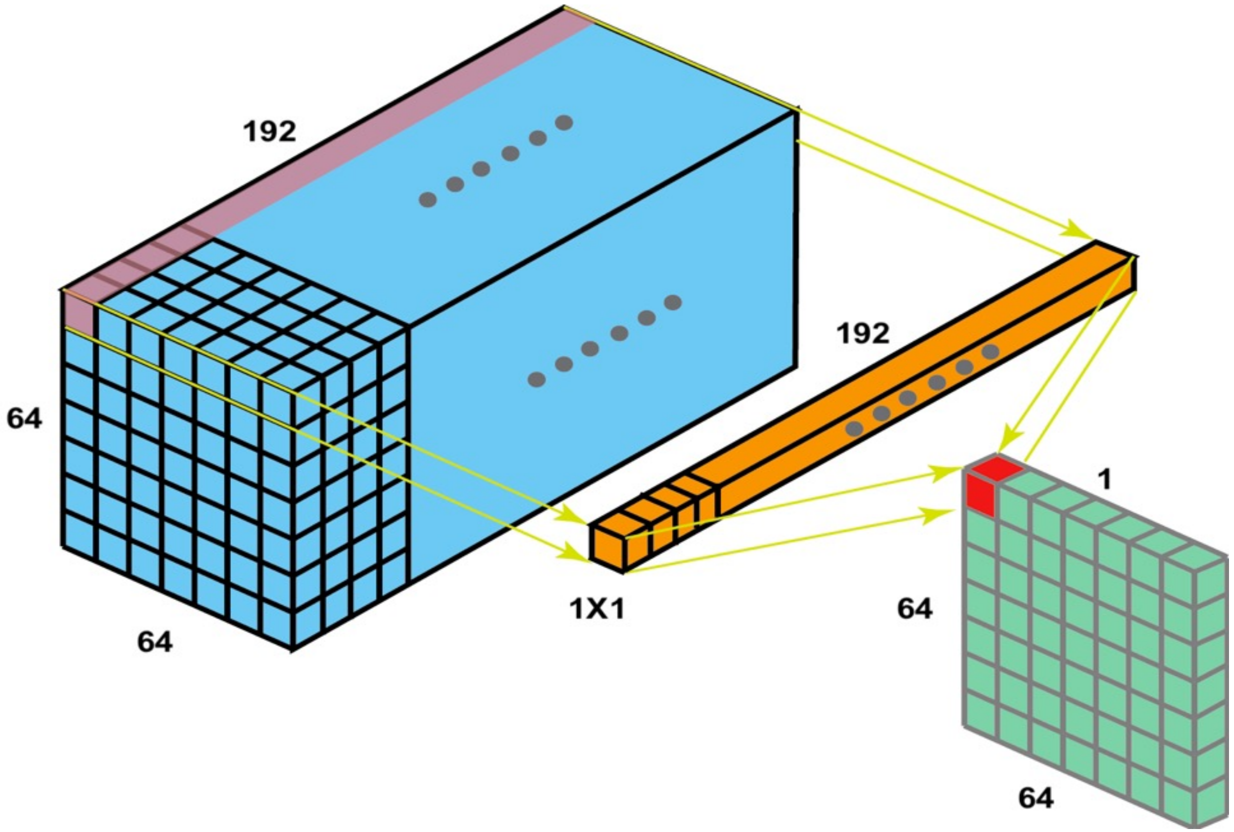


Image source: Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9. 2015.

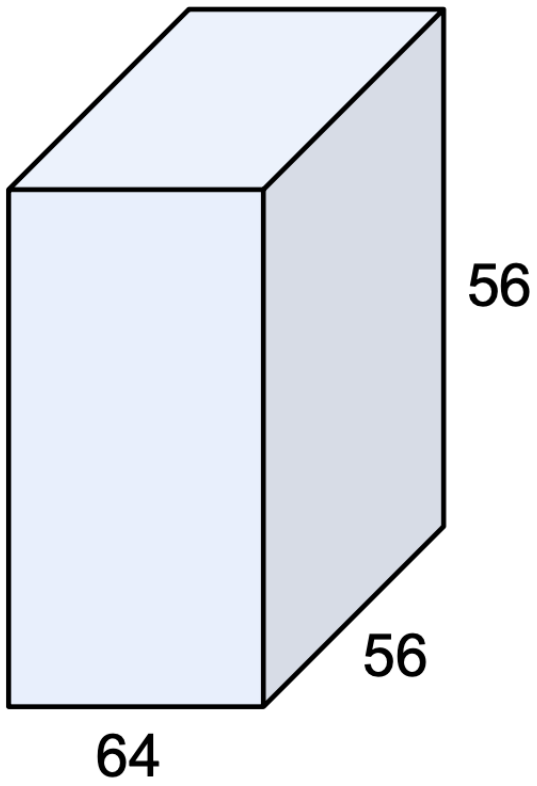
1x1 Convolution Filter

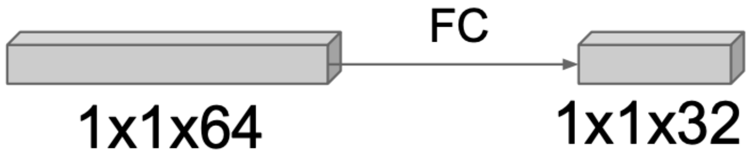


1X1 convolution was used to reduce/augment the number of channels while introducing non-linearity

Image source: https://medium.com/analytics-vidhya/talented-mr-1x1-comprehensive-look-at-1x1-convolution-in-deep-learning-f6b355825578

1x1 Convolution Filter





- Preserves spatial dimensions, reduces depth.
- Combination of feature maps.
- Number of parameters for 1x1 convolution filters: 64x32=2048

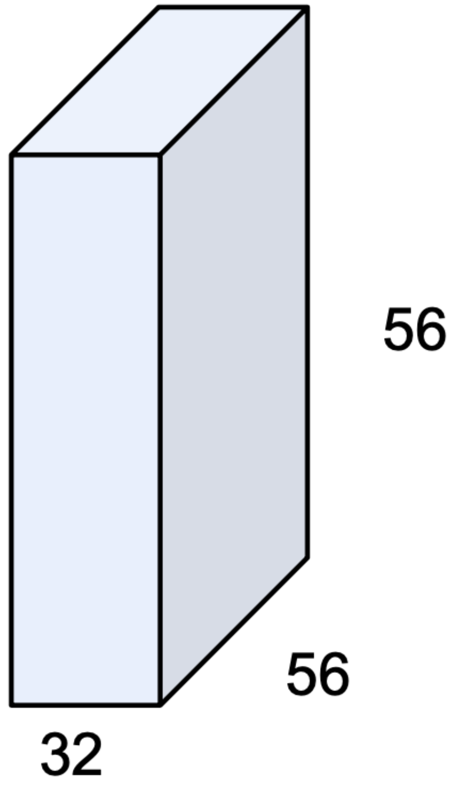


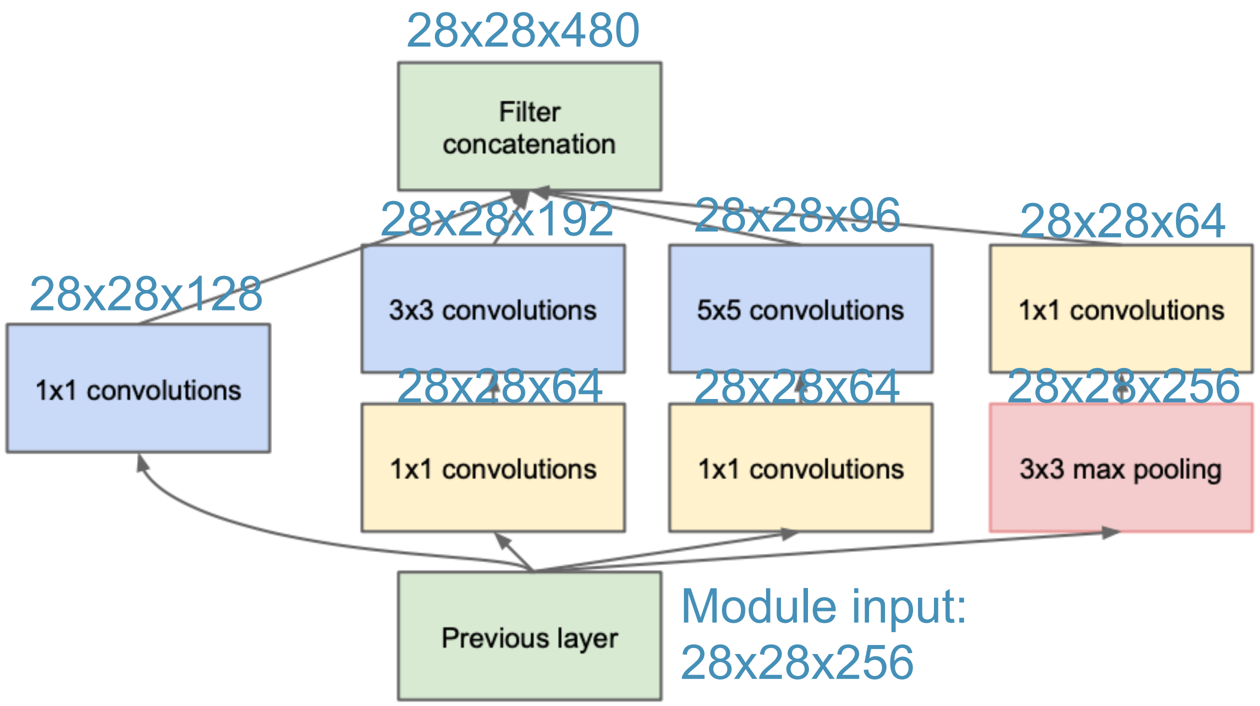
Image source: Lecture 9, cs231n.

Inception Module

- Using same parallel layers as naïve example, and adding "1x1 conv, 64 filter" bottlenecks:
- Parameters:

  [1x1 conv, 64] 64x1x1x256 Increased
  [1x1 conv, 64] 64x1x1x256
  [1x1 conv, 128] 128x1x1x256
  [3x3 conv, 192] 192x3x3x64 Decreased
  [5x5 conv, 96] 96x5x5x64 from 256
  [1x1 conv, 64] 64x1x1x256 Increased

- **Total: 33k,** decreased from 108k for naïve version.



Inception module with dimensionality reduction

Image source: Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9. 2015.

GoogLeNet

Use Inception as the basic module in GoogLeNet

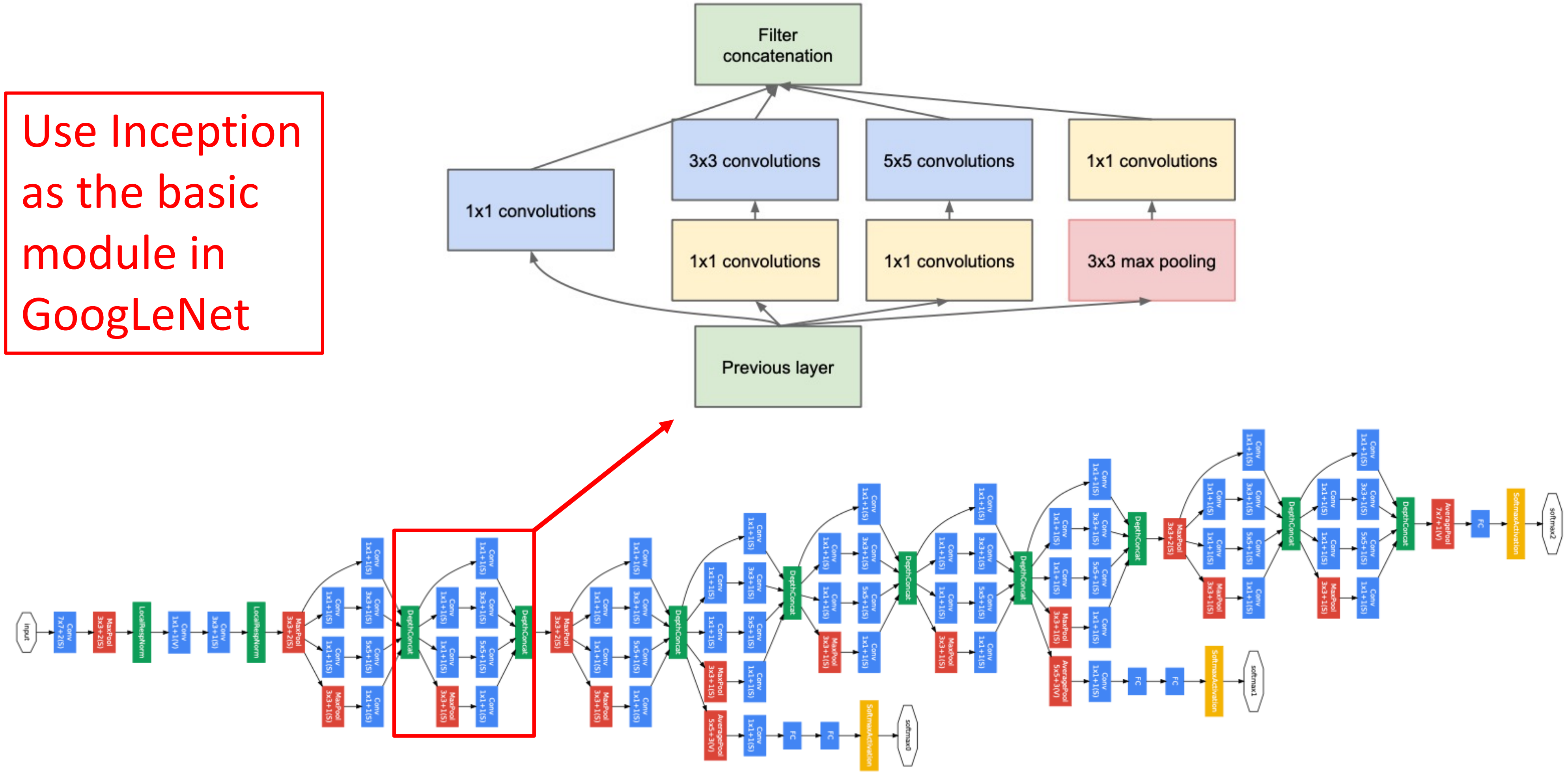


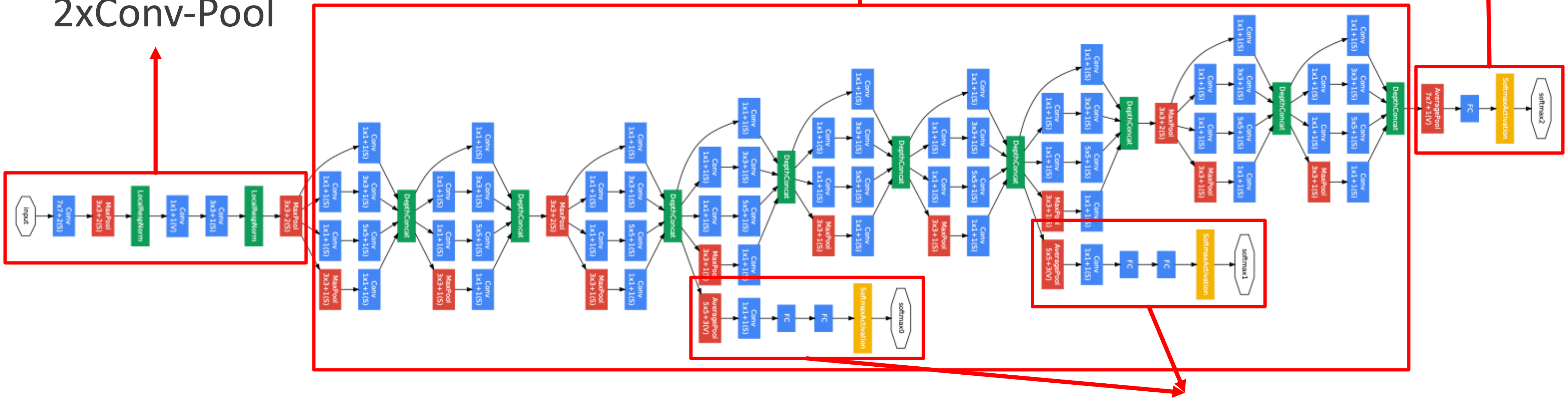
Image source: Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9. 2015.

GoogLeNet

Stacked Inception modules

Classifie
r output

Conv-Pool-
2xConv-Pool



Auxiliary classifier output to combat vanishing gradient problem

Image source: Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9. 2015.

GoogLeNet

- Instead of multiple expensive FC layers, GoogLeNet uses a global average pooling (GAP) layer to spatially average across each feature map, before final FC layer.

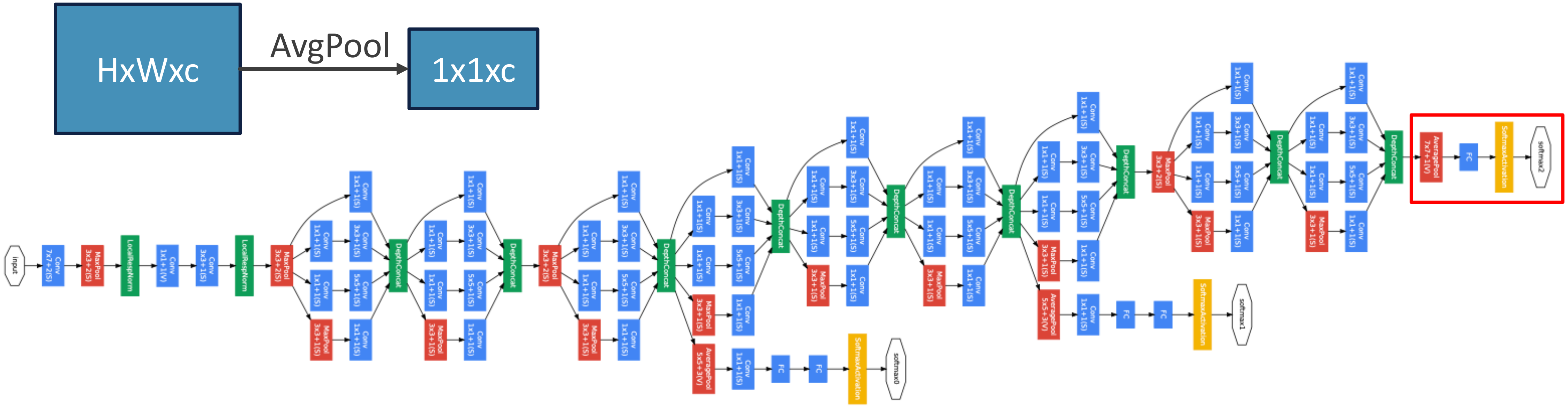
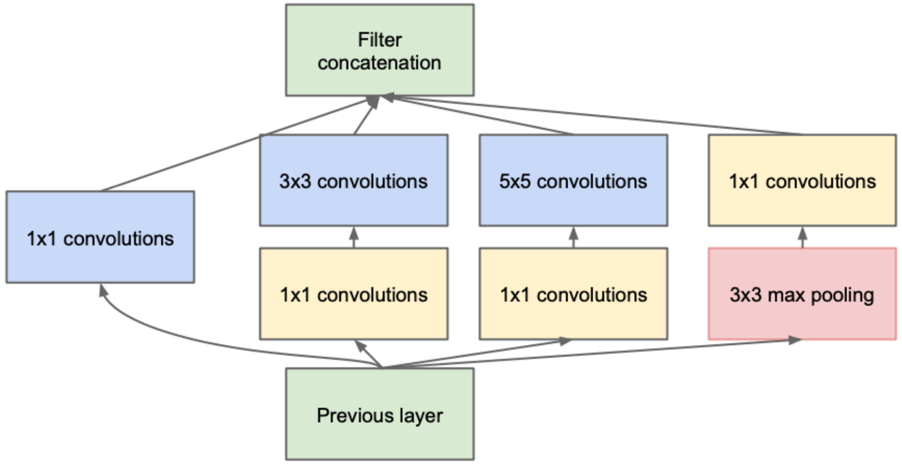


Image source: Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9. 2015.

GoogLeNet

- Deeper networks, with computational efficiency
  - 22 layers
  - Efficient “Inception” module
  - Avoids expensive FC layers
  - 12x less params than AlexNet
  - 27x less params than VGG-16
  - ILSVRC’14 classification winner (6.7% top 5 error)



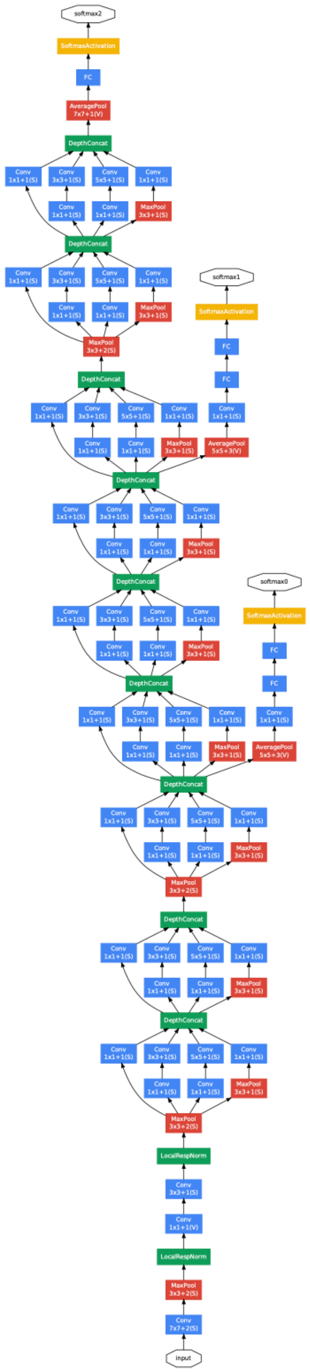
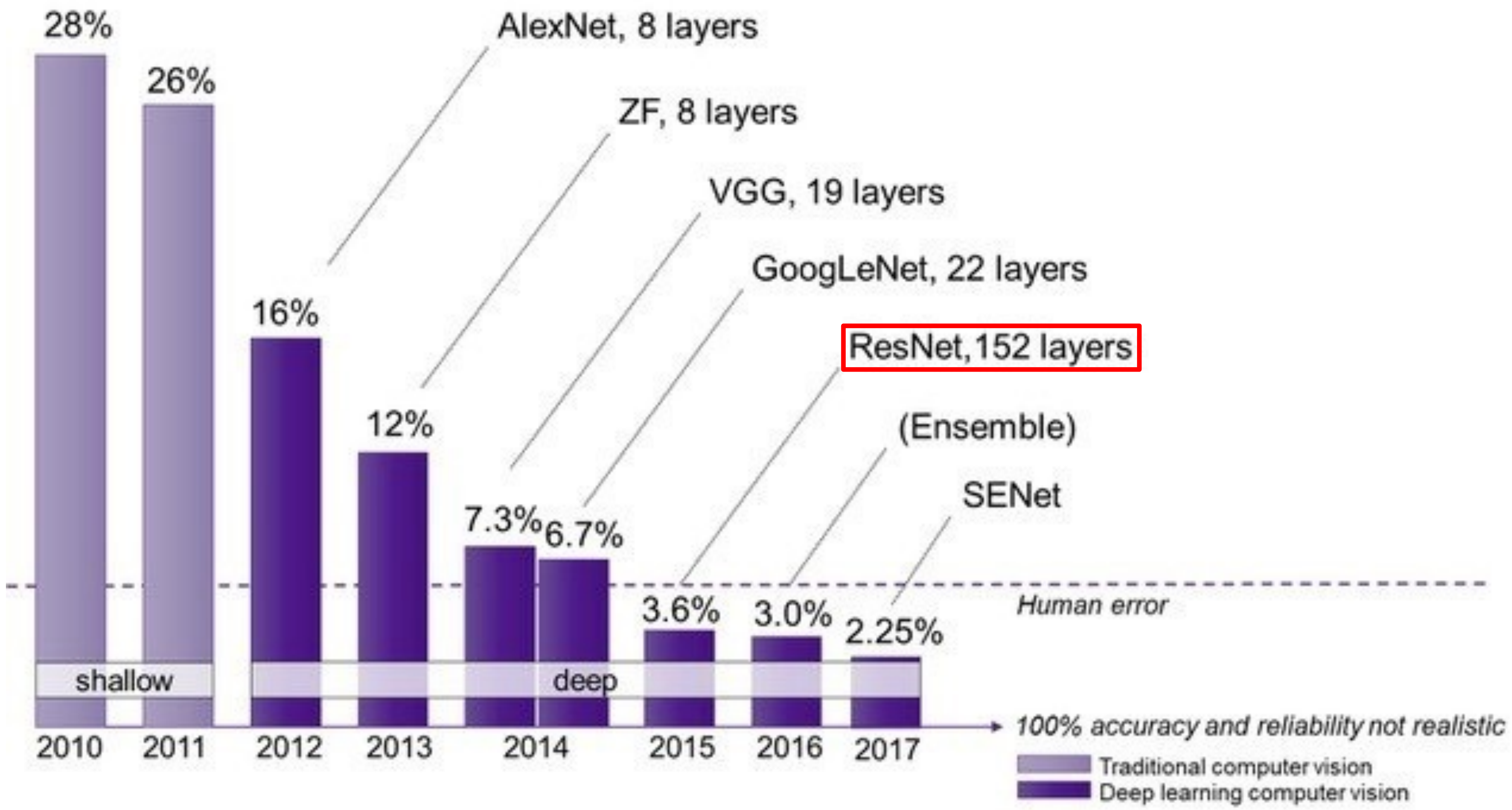


Image source: Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9. 2015.

RESNET

ILSVRC Winners

Deep residual learning for image recognition
K He, X Zhang, S Ren, J Sun - … and **pattern recognition**, 2016 - openaccess.thecvf.com
… **Deeper** neural **networks** are more difficult to train. We present a **residual learning** framework to ease the training of **networks** that are substantially **deeper** than those used previously. …
☆ Save Cite Cited by 185269 Related articles All 76 versions



Kaiming He 何恺明

Research Scientist

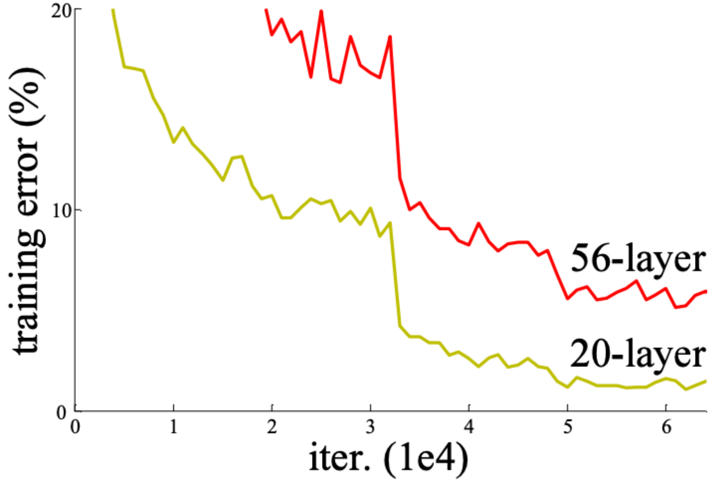
Facebook AI Research (FAIR), Menlo Park, CA

Join MIT as a faculty member in 2024

Image source: https://semiengineering.com/new-vision-technologies-for-real-world-applications/ https://kaiminghe.github.io/

The Deeper The Better?

- Now, it seems that we can conclude: "the deeper the better".
- Is learning better networks as easy as stacking more layers?
- A phenomenon has been observed: with the network depth increasing, accuracy becomes worse.
- And the most unexpected thing is that it is not caused by overfitting!
  - Training error increases as network gets deeper.



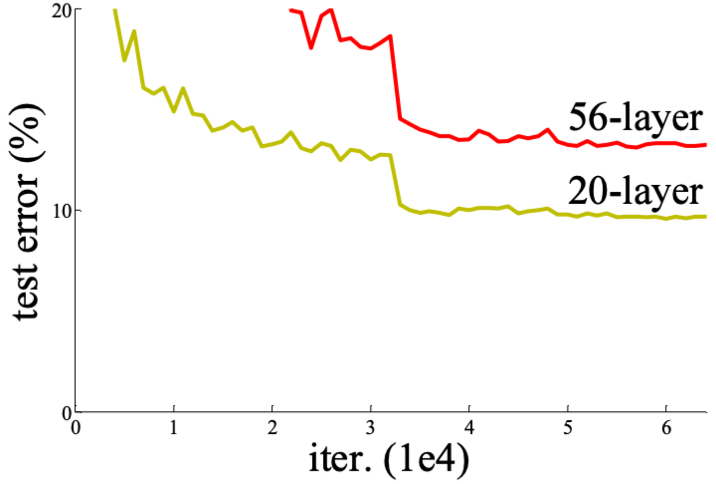


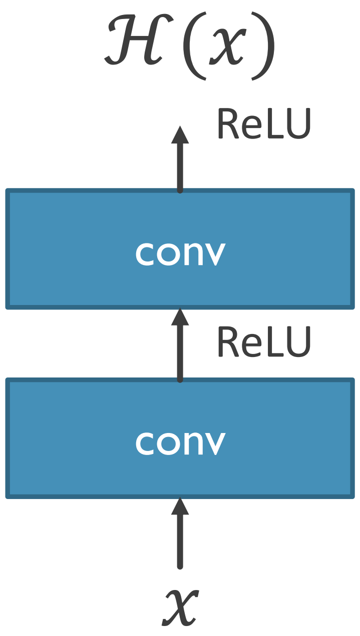
Image source: He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

The Deeper The Better?

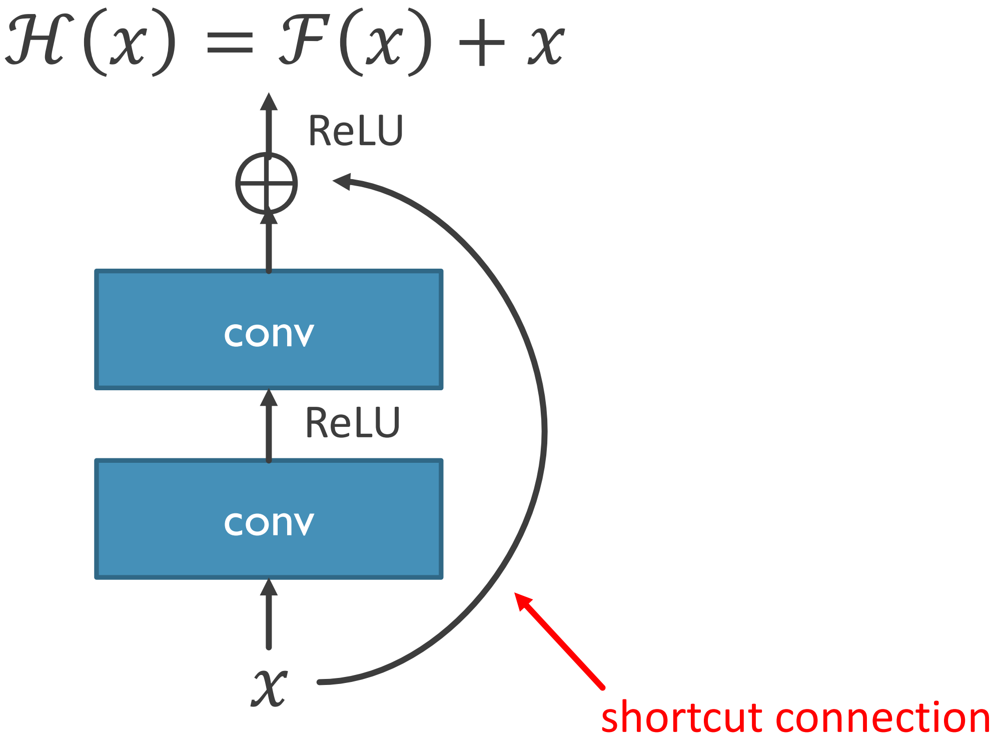
- Hypothesis: the problem is essentially an optimization problem, and deeper models are harder to optimize.
- Why?
  - Consider a shallow model and its deep counterpart.
  - The deep counterpart is constructed by: the additional layers are identity mapping, and the other layers are copied from the learned shallow model.
  - By this construction, they should produce identical results. Thus, a deeper model should produce no higher training error than its shallower counterpart.
- However, experiments show that we are unable to find solutions that are comparably good or better than the constructed solution.

ResNet Building Block

- Fit the residual $\mathcal{F}(x) = \mathcal{H}(x) - x$ instead of $\mathcal{H}(x)$ directly.



Plain layers



Residual block

Rationale behind ResNet

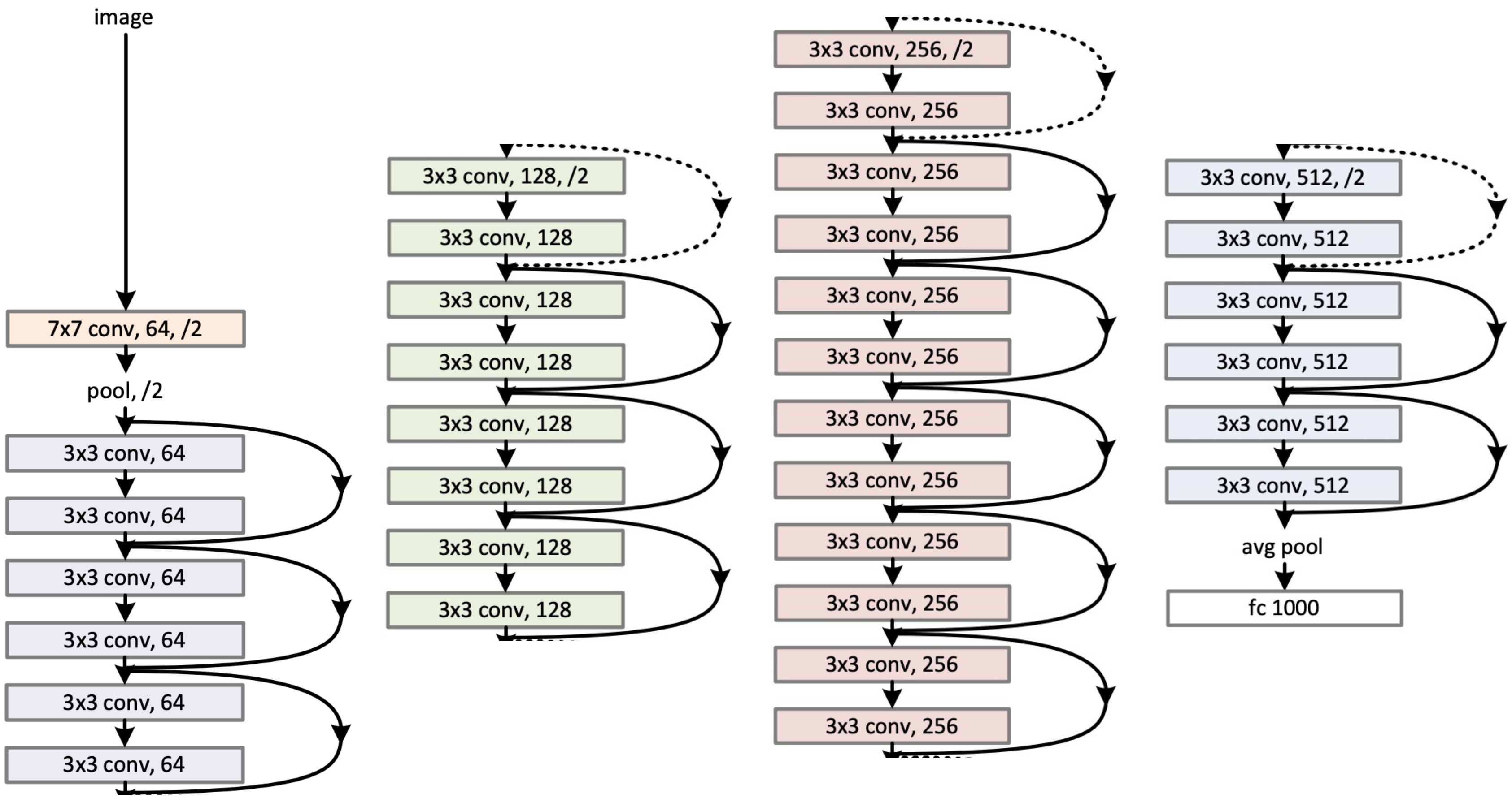
Why does fitting the residual help?

- Generally, if a few stacked layers are able to learn $\mathcal{H}(x)$, they are also able to learn $\mathcal{F}(x) = \mathcal{H}(x) - x$.
- However, it is easy to make $\mathcal{F}(x) = 0$, but difficult to make $\mathcal{H}(x) = x$.
- Declared in the original paper by He et al.:

> *"The degradation problem suggests that the solvers might have difficulties in approximating identity mappings by multiple nonlinear layers. With the residual learning reformulation, if identity mappings are optimal, the solvers may simply drive the weights of the multiple nonlinear layers toward zero to approach identity mappings."*

ResNet Architectures

- Solid lines are the identical shortcut connections.
- Dotted lines are the shortcut connections with increased dimension.



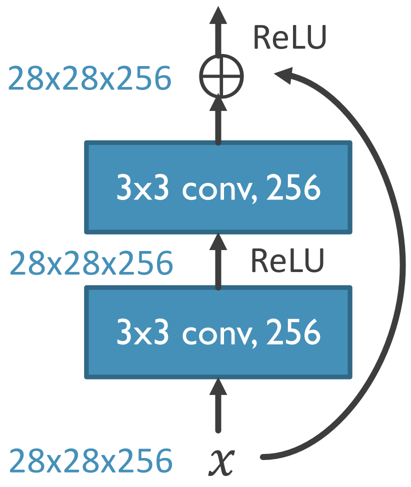
Architecture of ResNet-34

Image source: He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

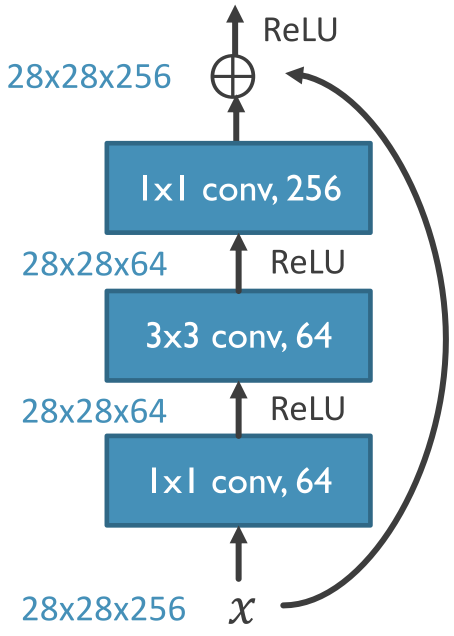
Deeper Bottleneck Architectures

- Deeper bottleneck architectures are adopted in ResNet-50, -101, and -152.
- Deeper non-bottleneck ResNets also gain accuracy from increased depth, but are not as economical as the bottleneck ResNets.
- The usage of bottleneck designs is mainly due to practical considerations.

Total param:
2x(256x3x3x256)
=1,179,648



Total param:
256x1x1x64
+64x3x3x64
+64x1x1x256
=69,632



ResNet Architectures

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

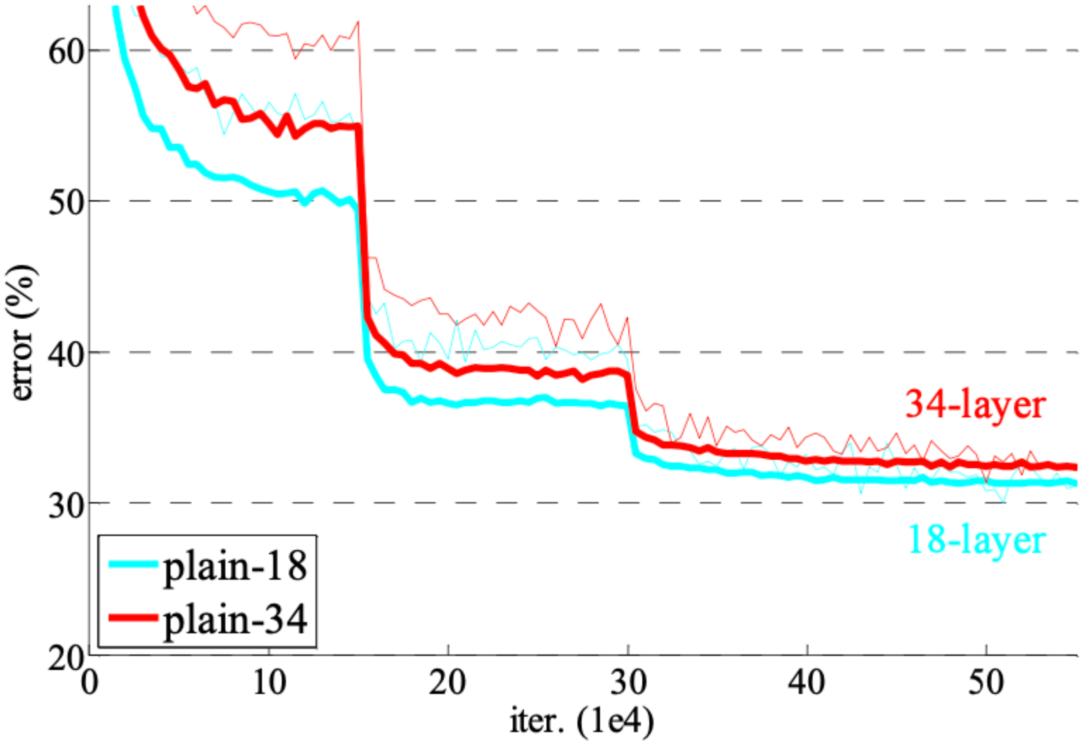
Image source: He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

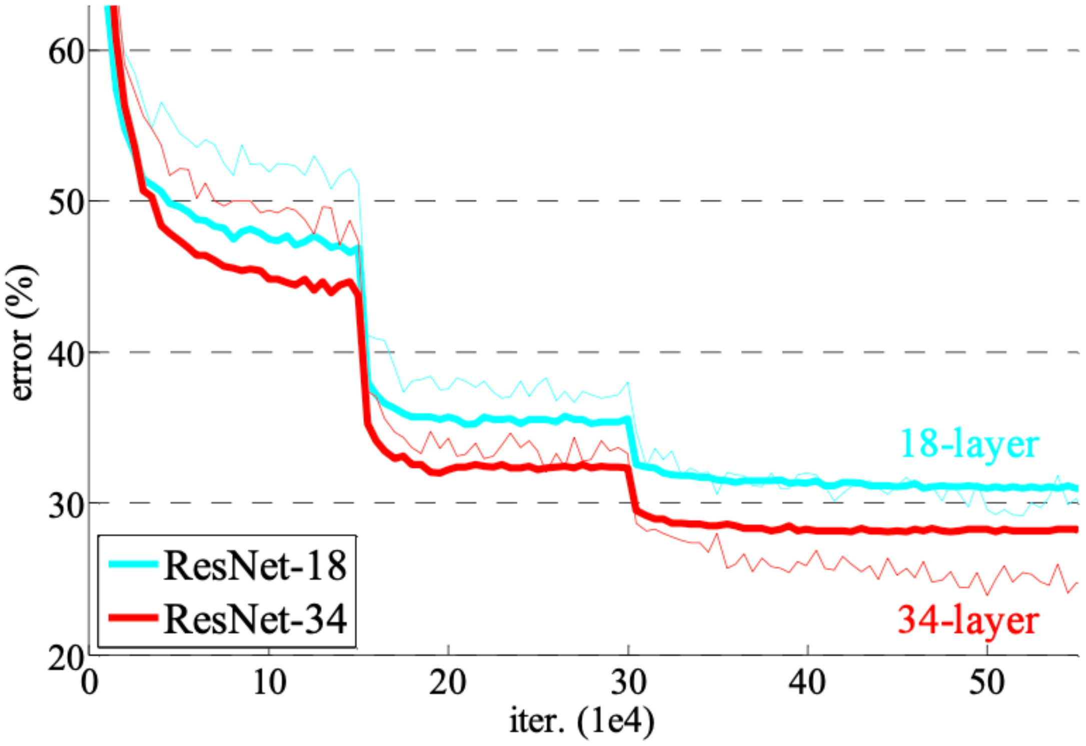
ResNet Training

Training ResNet in practice:

- Batch Normalization after every CONV layer.
- Kaiming initialization.
- SGD + Momentum (0.9).
- Learning rate: 0.1, divided by 10 when validation error plateaus.
- Mini-batch size 256.
- Weight decay of 1e-5.
- No dropout used.

ResNet Performance



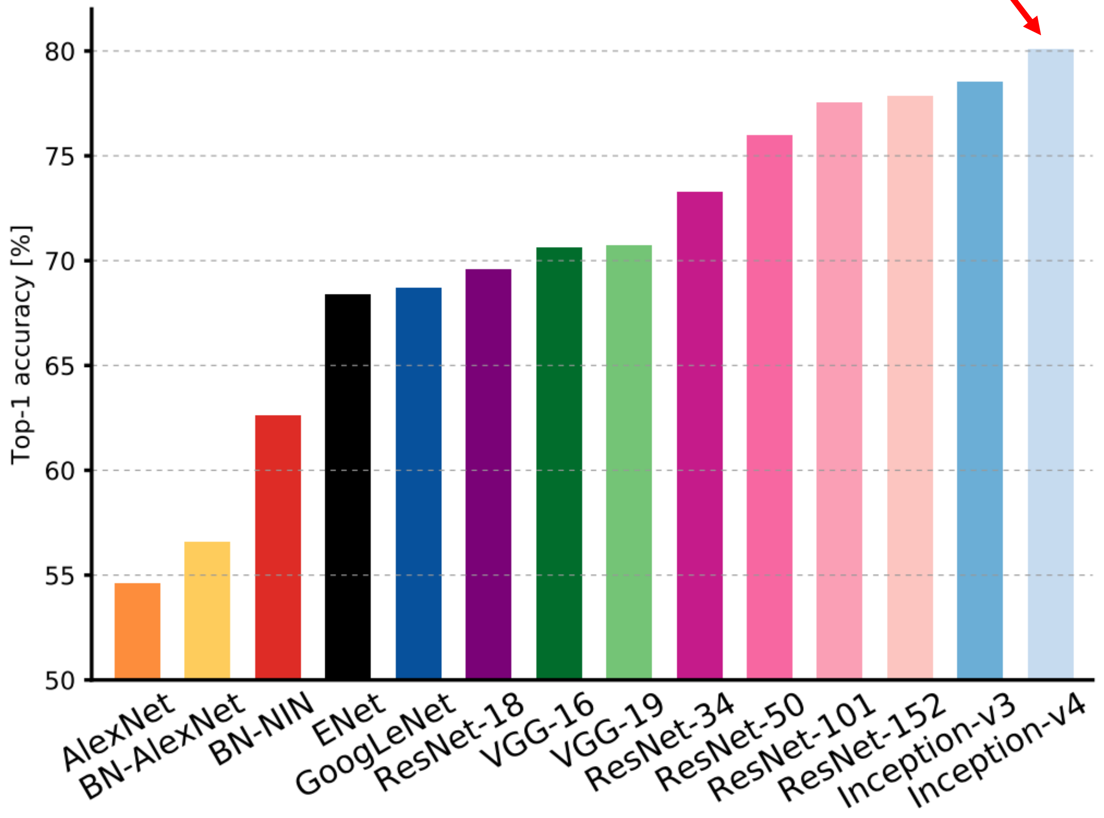


thin curves: training error bold curves: validation error

Image source: He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

ILSVRC Model Comparison

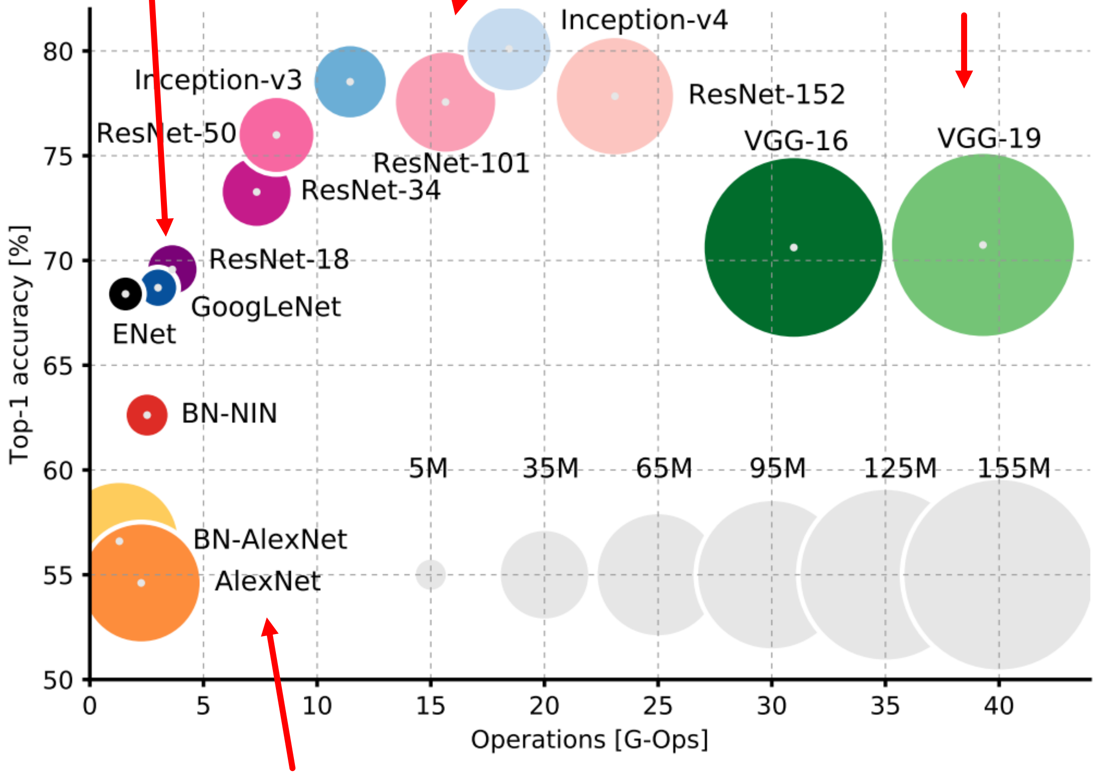
Inception-v4: Resnet + Inception



GoogLeNet: most efficient

ResNet: efficient and accurate

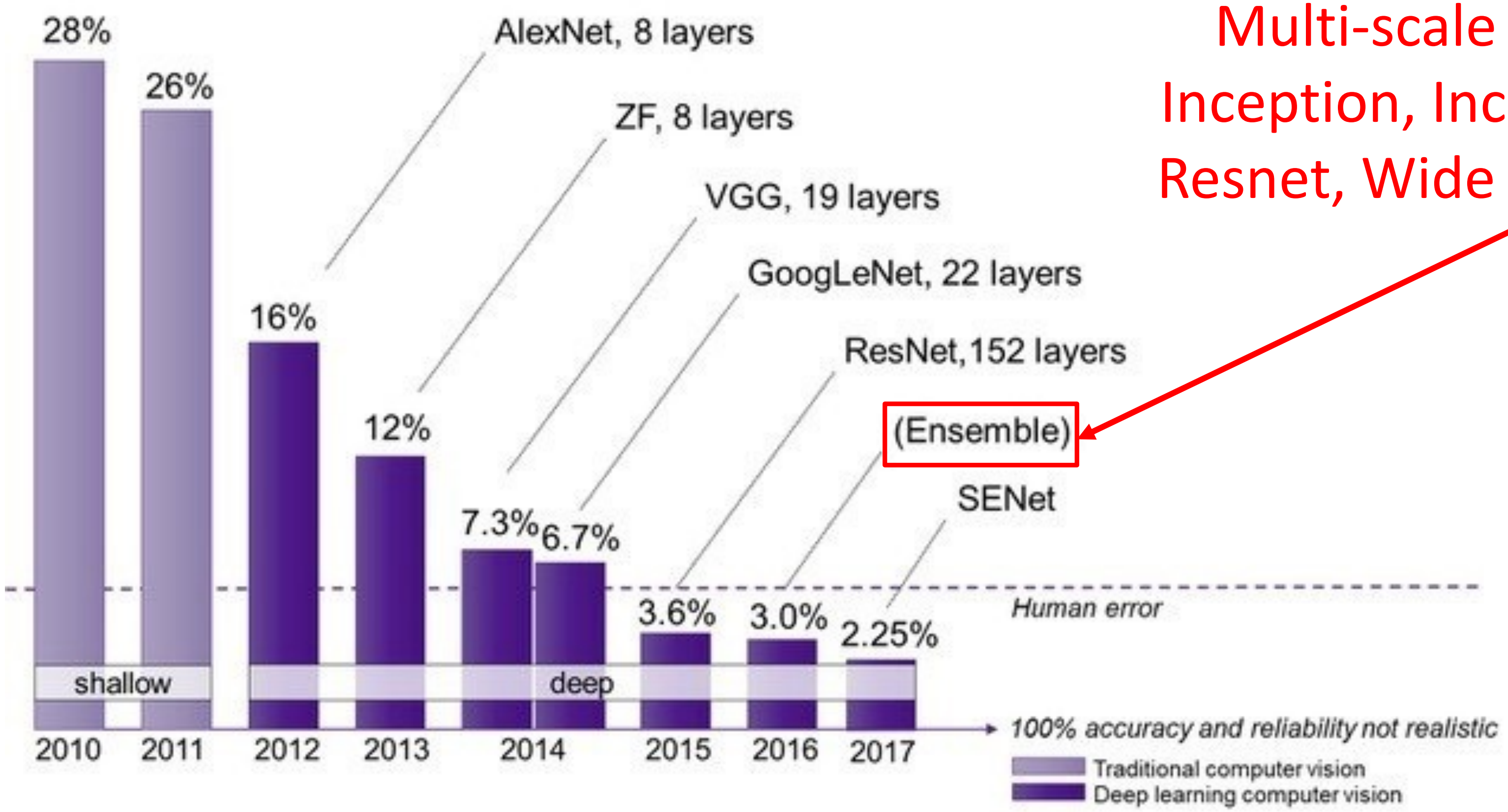
VGG: most parameters, most operations



AlexNet: Father of all models

Image source: Canziani, Alfredo, Adam Paszke, and Eugenio Culurciello. "An analysis of deep neural network models for practical applications." arXiv preprint arXiv:1605.07678 (2016).

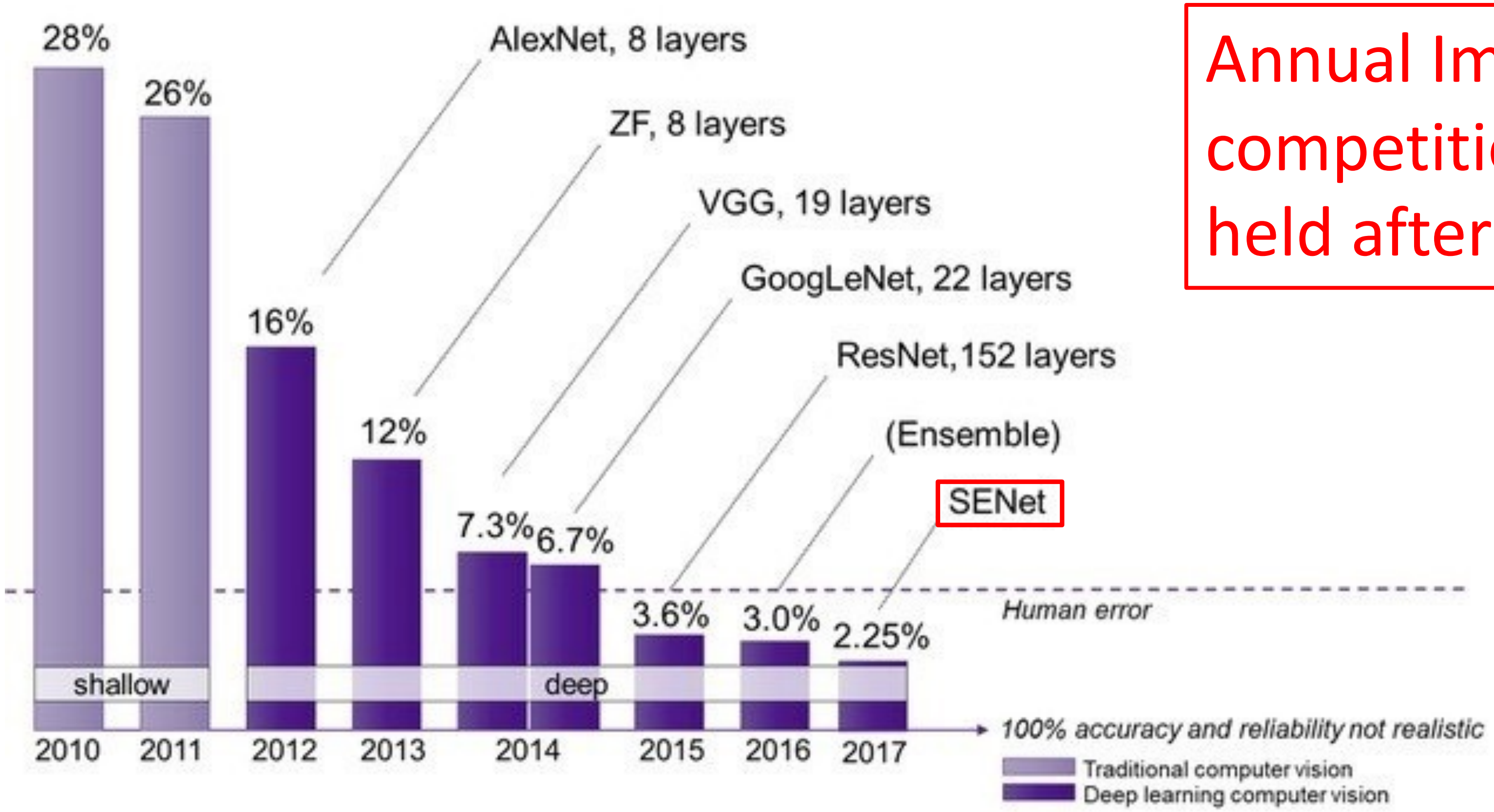
ILSVRC



Multi-scale ensemble of Inception, Inception-Resnet, Resnet, Wide Resnet models

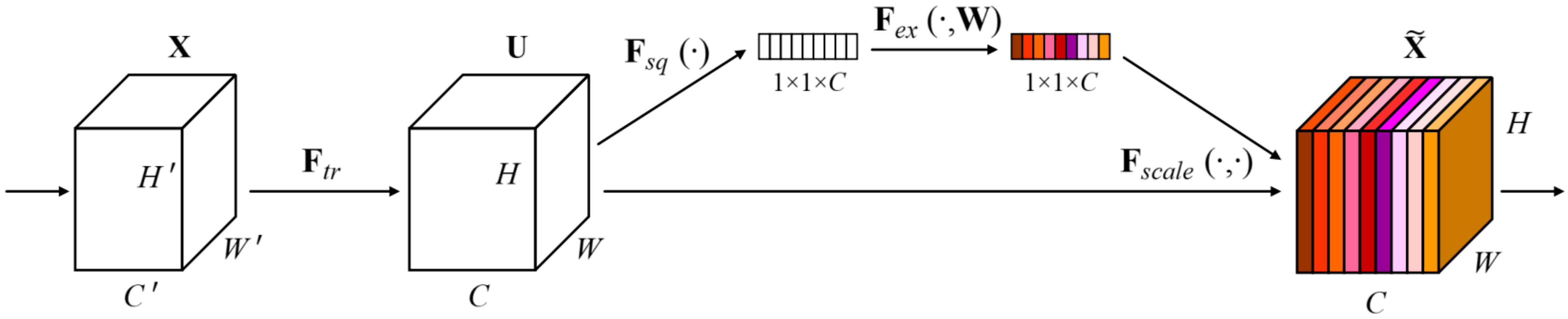
ILSVRC

Squeeze-and-excitation networks
J Hu, L Shen, G Sun - … of the IEEE conference on computer …, 2018 - openaccess.thecvf.com
Convolutional neural **networks** are built upon the convolution operation, which extracts informative features by fusing spatial and channel-wise information together within local …
☆ Save 99 Cite Cited by 25199 Related articles All 25 versions



Annual ImageNet competition no longer held after 2017

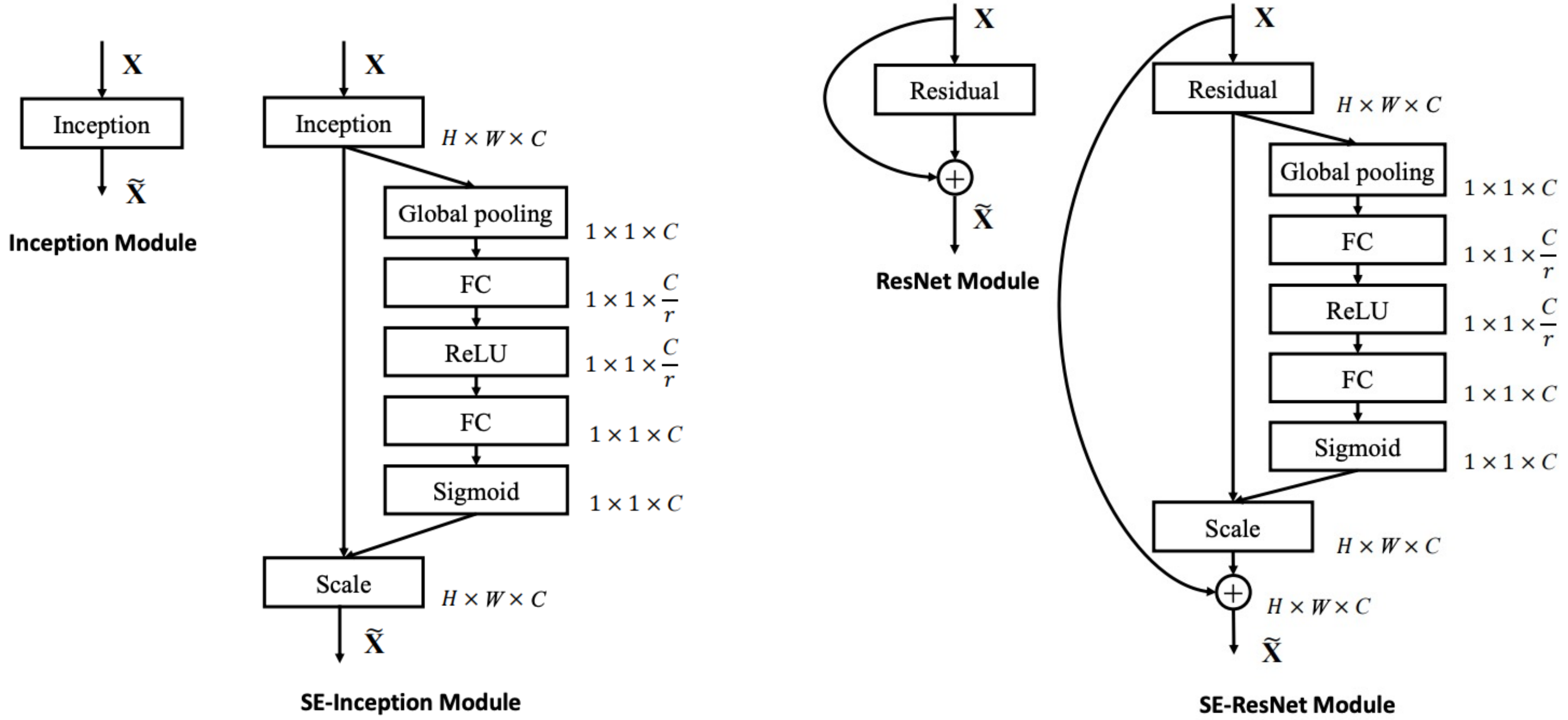
SENet



- Motivation: Explicitly model channel-interdependencies within modules.
- The SE module has three steps:
  - Squeeze: global average pooling for each channel.
  - Excitation: Explicitly model channel association by channel-wise weights.
  - Scale: Reweight feature maps.

Image source: Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132-7141. 2018.

SENet



- SE module can embed with Inception or ResNet.
- Besides, a bottleneck architecture is adopted at excitation step for more nonlinearity and reduce model parameters.

Image source: Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132-7141. 2018.

CBAM

Cbam: Convolutional block attention module
S Woo, J Park, JY Lee… - Proceedings of the …, 2018 - openaccess.thecvf.com
We propose Convolutional Block Attention Module (**CBAM**), a simple and effective attention module that can be integrated with any feed-forward convolutional neural networks. Given …
☆ Save 99 Cite Cited by 14090 Related articles All 13 versions »

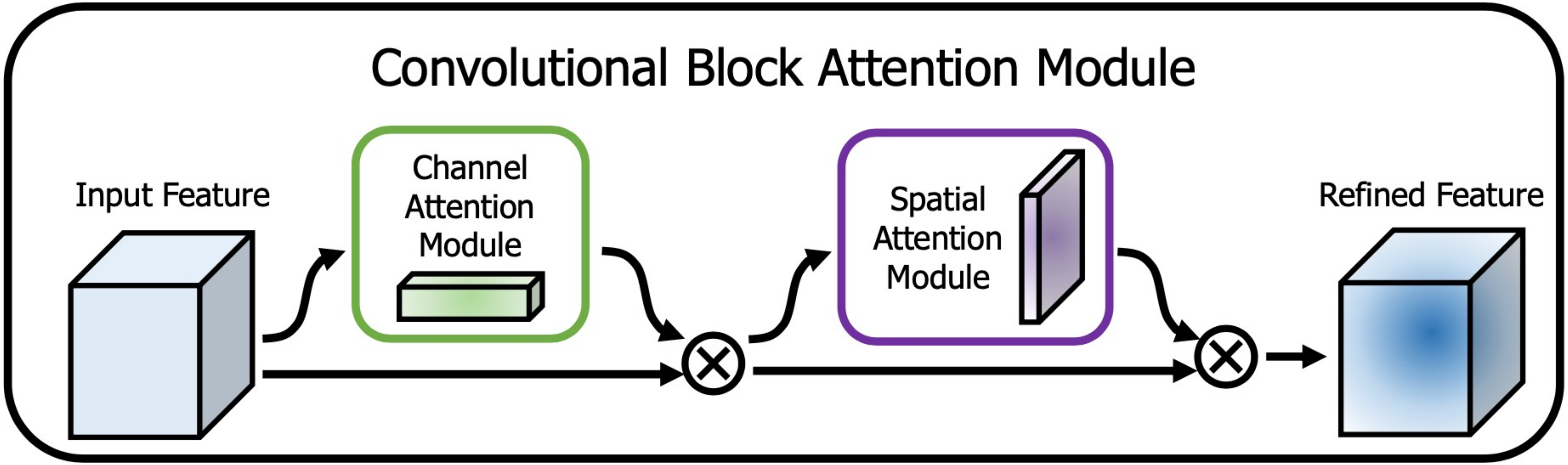


Image source: Woo, Sanghyun, Jongchan Park, Joon-Young Lee, and In So Kweon. "Cbam: Convolutional block attention module." In Proceedings of the European conference on computer vision (ECCV), pp. 3-19. 2018.

CBAM

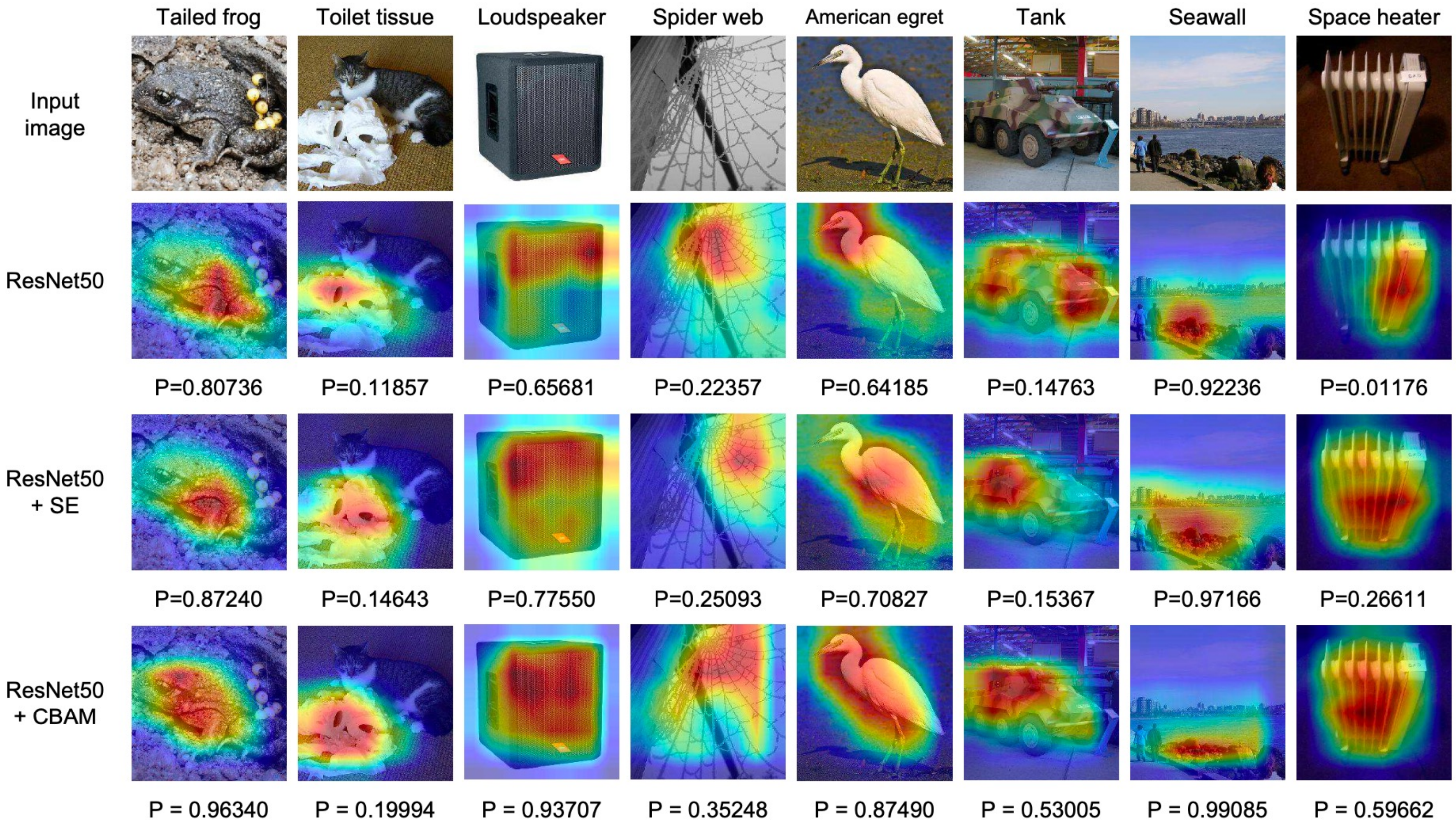
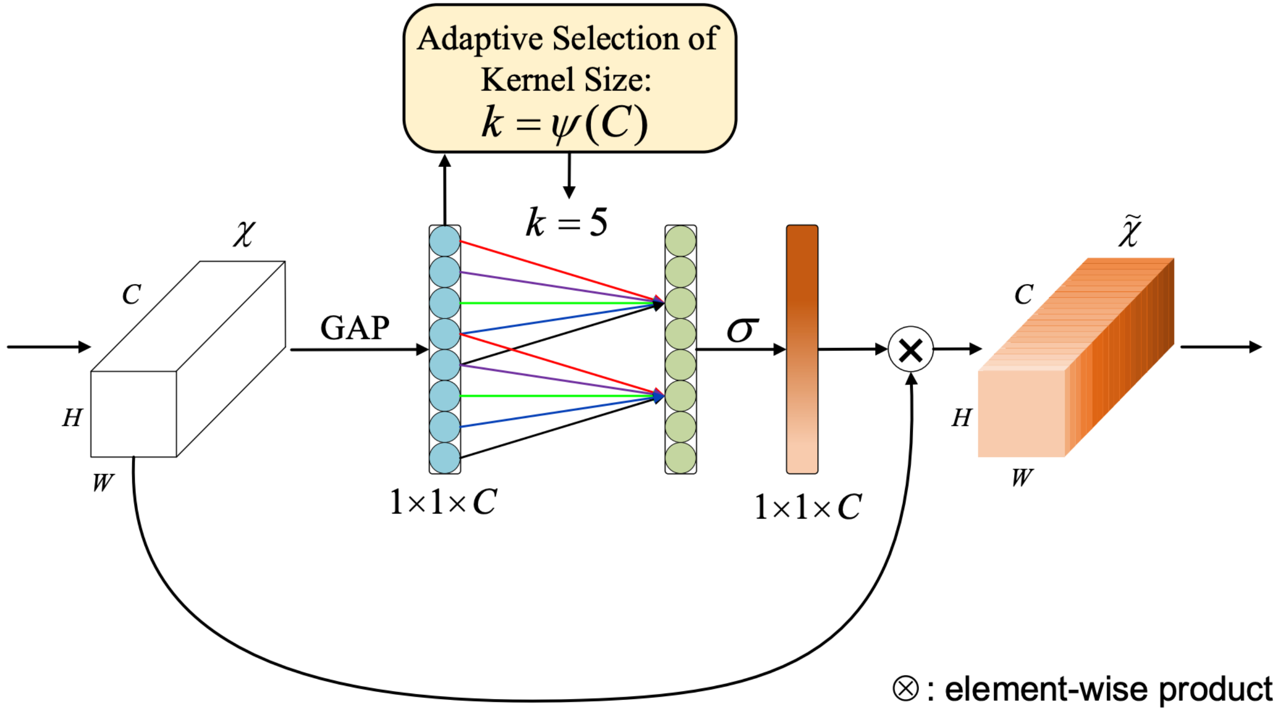
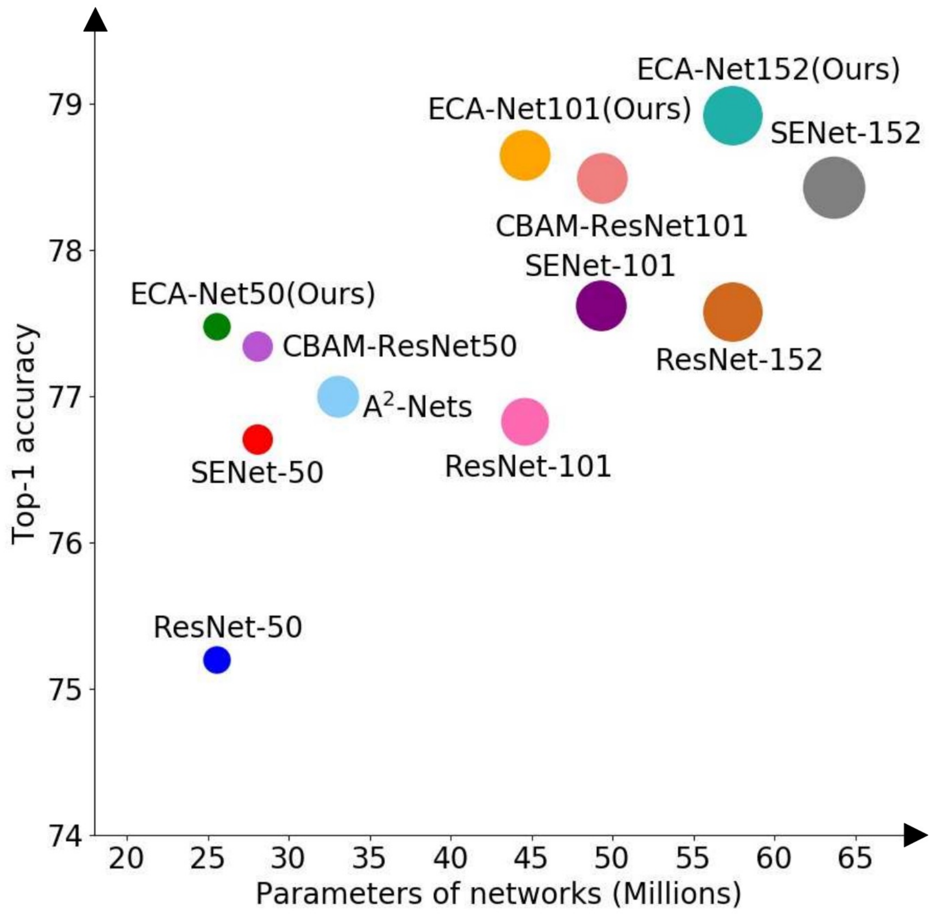


Image source: Woo, Sanghyun, Jongchan Park, Joon-Young Lee, and In So Kweon. "Cbam: Convolutional block attention module." In Proceedings of the European conference on computer vision (ECCV), pp. 3-19. 2018.

ECA-Net

ECA-Net: Efficient channel attention for deep convolutional neural networks
Q Wang, B Wu, P Zhu, P Li, W Zuo… - Proceedings of the …, 2020 - openaccess.thecvf.com
Recently, channel attention mechanism has demonstrated to offer great potential in improving the performance of deep convolutional neural networks (CNNs). However, most …
Save Cite Cited by 3213 Related articles All 13 versions





Reference: Wang, Qilong, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. "ECA-Net: Efficient channel attention for deep convolutional neural networks." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 11534-11542. 2020.

FOLLOWING ADVANCES

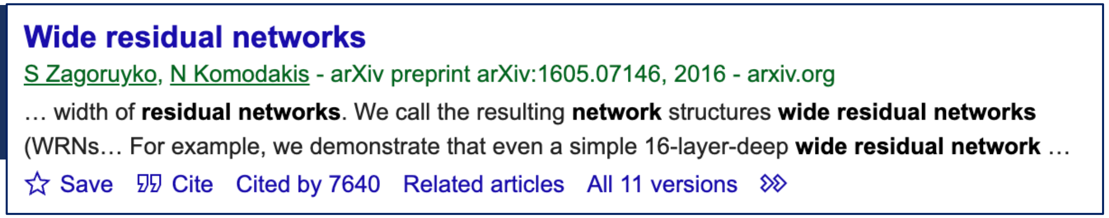
Directions

- Improve accuracy.
- Improve efficiency.
- Model architecture searching.

Directions

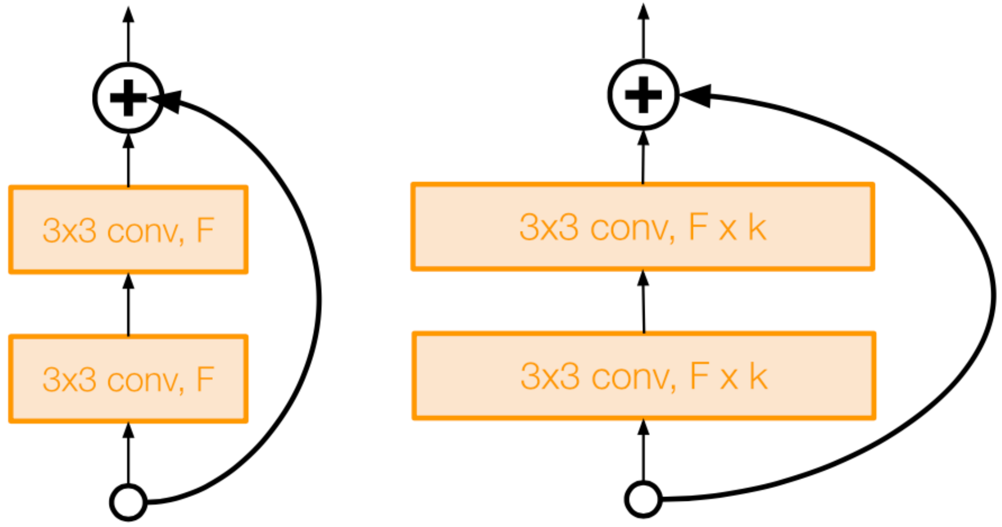
- Improve accuracy.
  - Wide ResNet.
  - ResNeXt.
  - DenseNet.
- Improve efficiency.
- Model architecture searching.

Wide ResNet



Wide residual networks
S Zagoruyko, N Komodakis - arXiv preprint arXiv:1605.07146, 2016 - arxiv.org
… width of **residual networks**. We call the resulting **network** structures **wide residual networks** (WRNs… For example, we demonstrate that even a simple 16-layer-deep **wide residual network** …
Save Cite Cited by 7640 Related articles All 11 versions

- Argues that residuals are the important factor, not depth.
- Use wider residual blocks (Fxk filters instead of F filters in each layer).
- 50-layer wide ResNet outperforms 152-layer original ResNet.
- Increasing width instead of depth more computationally efficient.



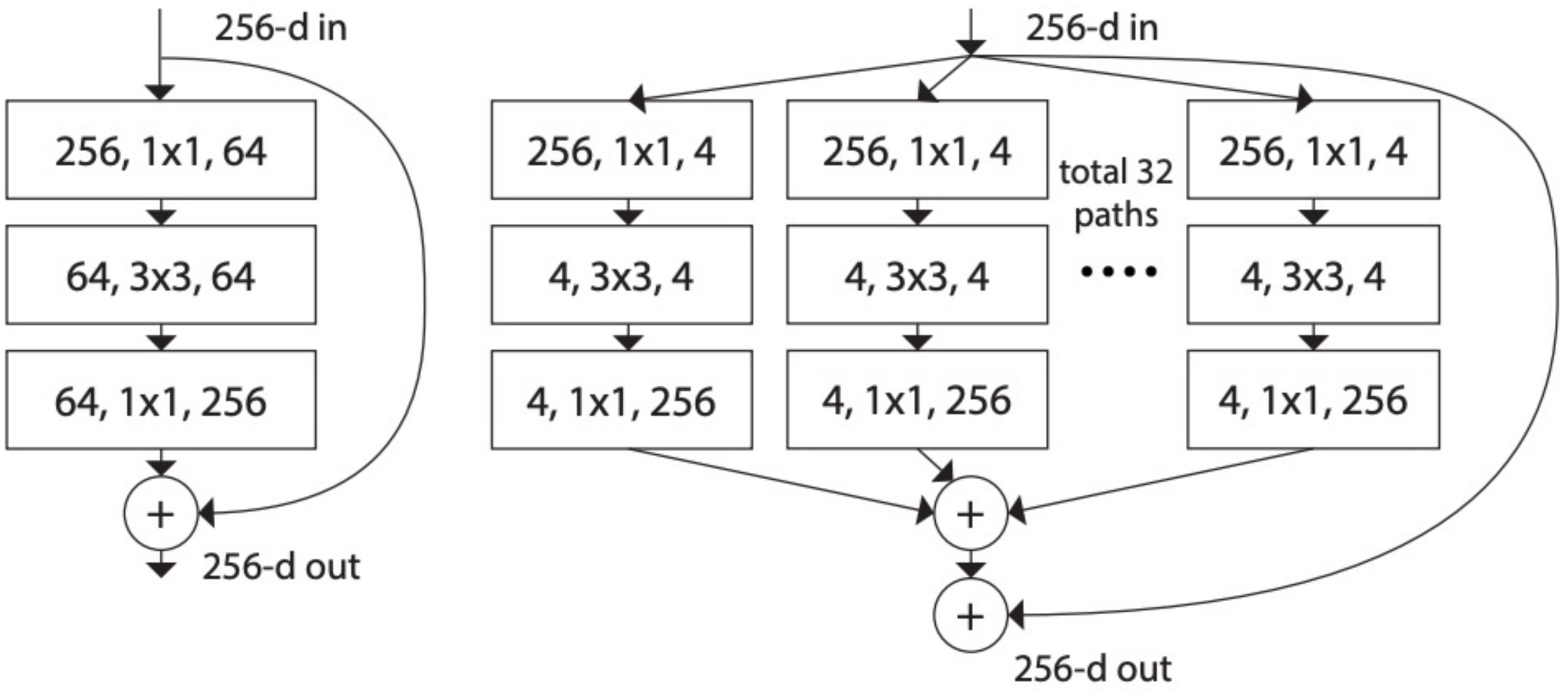
Basic residual block

Wide residual block

Image source: Lecture 9, cs231n.

ResNeXt

Aggregated residual transformations for deep neural networks
S Xie, R Girshick, P Dollár, Z Tu… - Proceedings of the IEEE …, 2017 - openaccess.thecvf.com
… are **aggregated** by summation. We pursuit a simple realization of this idea — the transformations to be **aggregated** are … We empirically demonstrate that our **aggregated** transformations …
Save Cite Cited by 10779 Related articles All 13 versions



- Increases width of residual block through multiple parallel pathways.
- Parallel pathways similar in spirit to Inception module.

Image source: Xie, Saining, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. "Aggregated residual transformations for deep neural networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1492-1500. 2017.

DenseNet

Densely connected convolutional networks
G Huang, Z Liu, L Van Der Maaten… - Proceedings of the …, 2017 - openaccess.thecvf.com
… part of **convolutional networks** is down-… **network** into multiple **densely connected dense** blocks; see Figure 2. We refer to layers between blocks as transition layers, which do **convolution** …

- Residual is good!
- Deep residual has some problems: A great amount of redundancy in deep (residual) networks.
  - Not all layers may be needed.
- Wide residual has some problems: Large number of parameters.
- Dense residual is the solution!

DenseNet

- Each layer is connected to every other layer in feedforward fashion.
  - ResNet: L layers have L shortcut connections.
  - DenseNet: L layers have $L(L+1)/2$ shortcut connections.

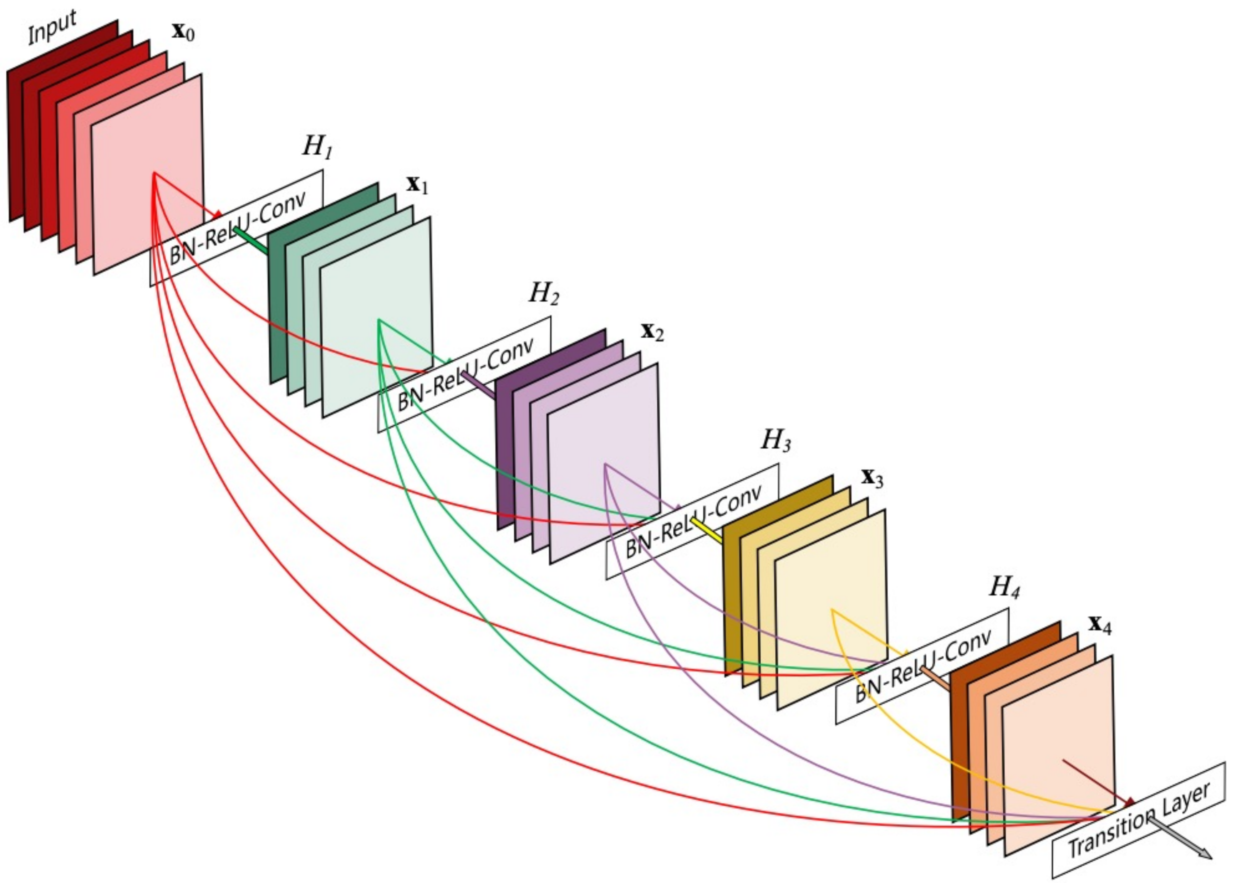


Image source: Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. “Densely connected convolutional networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700-4708. 2017.

DenseNet

- Advantages:
  - Require fewer parameters. Avoid redundant feature-maps.
  - Encourages feature reuse throughout the network.
  - Improved flow of information and gradients. Easy to train.
- Experiments showed that shallow 50-layer network can outperform deeper 152 layer ResNet.

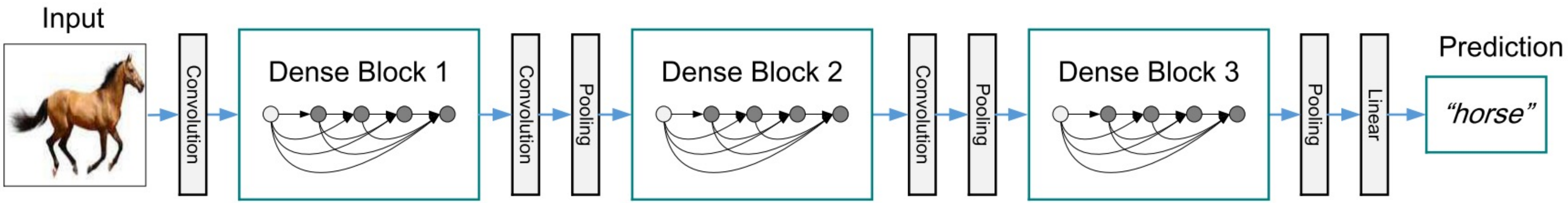


Image source: Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. "Densely connected convolutional networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700-4708. 2017.

DenseNet

- Experiment explicitly designed for the effectiveness of feature reuse.
- Features extracted by very early layers are, indeed, directly used by deep layers throughout the same dense block.

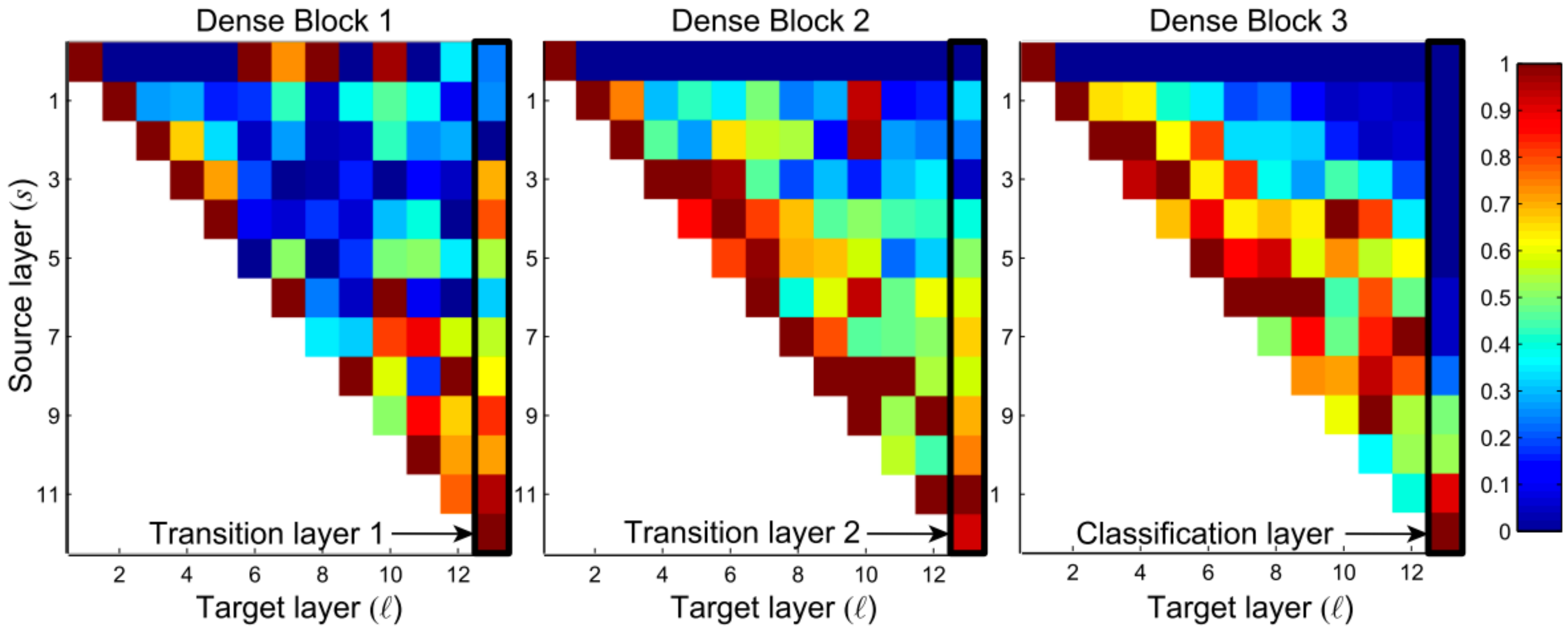


Image source: Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. "Densely connected convolutional networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700-4708. 2017.

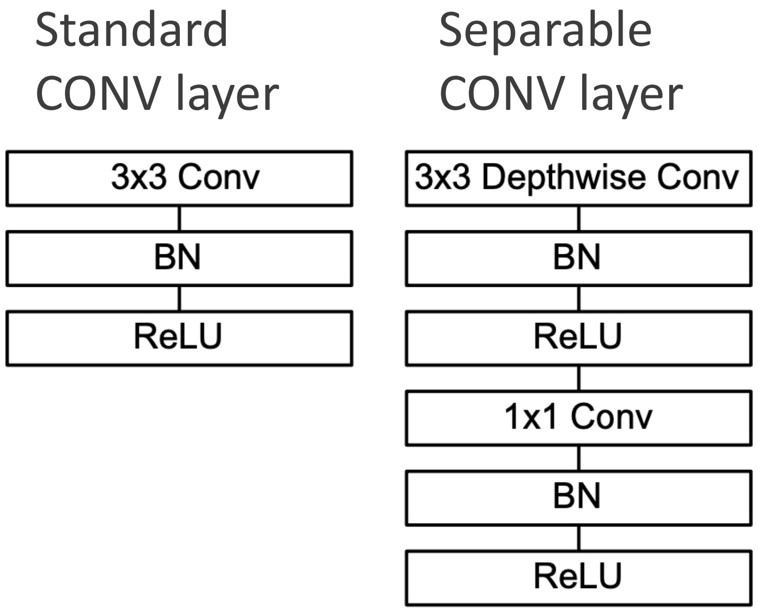
Directions

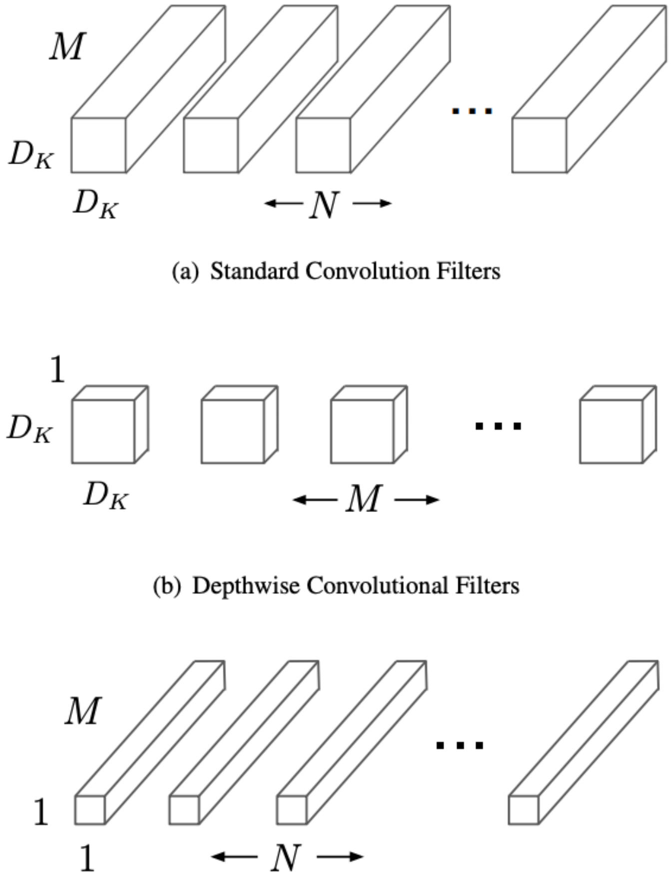
- Improve accuracy.
- Improve efficiency.
  - MobileNet.
  - ShuffleNet.
- Model architecture searching.

MobileNet

Mobilenets: Efficient convolutional neural networks for mobile vision applications
AG Howard, M Zhu, B Chen, D Kalenichenko… - arXiv preprint arXiv …, 2017 - arxiv.org
… models called **MobileNets** for mobile and embedded vision applications. **MobileNets** are …
We then demonstrate the effectiveness of **MobileNets** across a wide range of applications …
☆ Save 99 Cite Cited by 21791 Related articles All 10 versions

- Group convolutions: decompose standard convolution filters into depthwise filters and pointwise filters.
- Much more efficient, with little loss in accuracy.





(a) Standard Convolution Filters

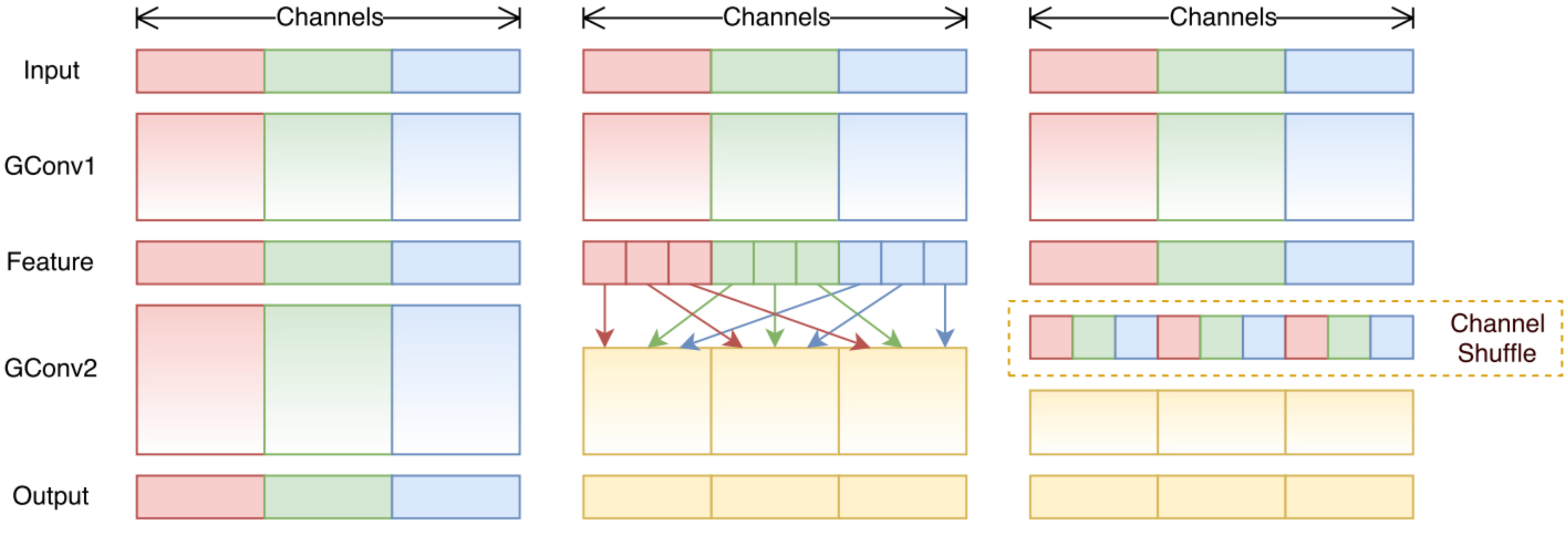
(b) Depthwise Convolutional Filters

(c) 1 × 1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Image source: Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).

ShuffleNet

Shufflenet: An extremely efficient convolutional neural network for mobile devices
X Zhang, X Zhou, M Lin, J Sun - Proceedings of the IEEE …, 2018 - openaccess.thecvf.com
… called **ShuffleNet**. Compared with popular structures like [31… complexity budget, our **ShuffleNet** allows more feature map … MobileNet [12], **ShuffleNet** achieves superior performance by a …



(a) (b) (c)

- A side effect brought by group convolutions: frequently using costly dense 1×1 convolutions.
- A novel channel shuffle operation to help the information flowing across feature channels.

Image source: Zhang, Xiangyu, Xinyu Zhou, Mengxiao Lin, and Jian Sun. "Shufflenet: An extremely efficient convolutional neural network for mobile devices." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6848-6856. 2018.

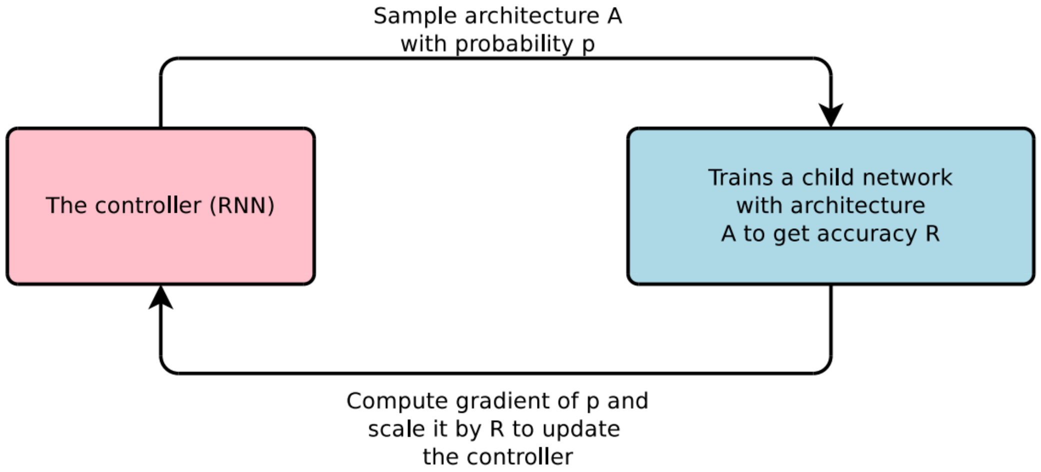
Directions

- Improve accuracy.
- Improve efficiency.
- Model architecture searching.
  - NAS.
  - EfficientNet.

NAS

Neural architecture **search** with **reinforcement learning**
B Zoph, QV Le - arXiv preprint arXiv:1611.01578, 2016 - arxiv.org
… recurrent **neural network**. Let's suppose we would like to predict feedforward **neural networks** … our controller recurrent **neural network** samples a simple convolutional **network**. It predicts …
☆ Save 99 Cite Cited by 5604 Related articles All 15 versions

- Train another network by reinforcement learning to learn to design good network architecture.



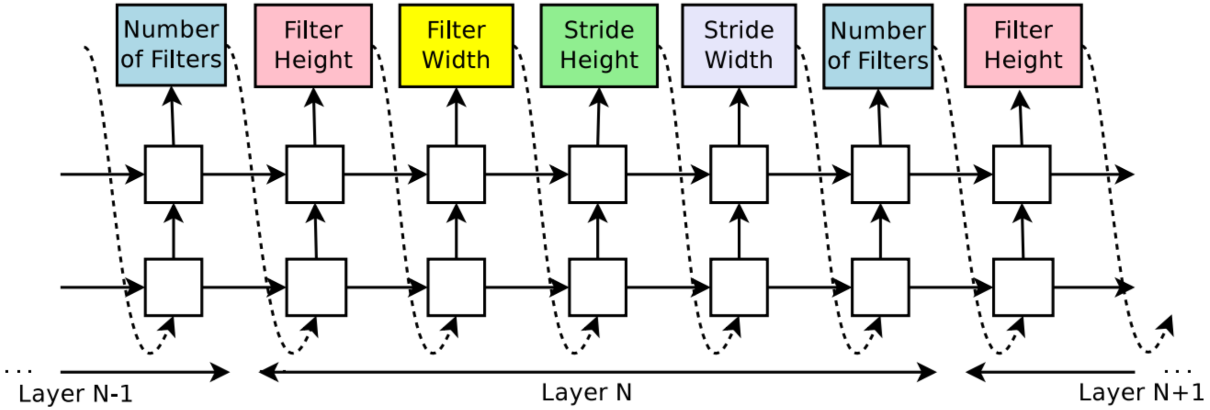


Image source: Zoph, Barret, and Quoc V. Le. "Neural architecture search with reinforcement learning." arXiv preprint arXiv:1611.01578 (2016).

EfficientNet

Efficientnet: Rethinking model scaling for convolutional neural networks
M Tan, Q Le - International conference on machine learning, 2019 - proceedings.mlr.press
Convolutional Neural Networks (ConvNets) are commonly developed at a fixed resource budget, and then scaled up for better accuracy if more resources are given. In this paper, we …
☆ Save 99 Cite Cited by 15618 Related articles All 12 versions »

- We can increase the model complexity with a variety of methods.
- Search for optimal set of compound scaling factors.
- Scale up using smart heuristic rules.

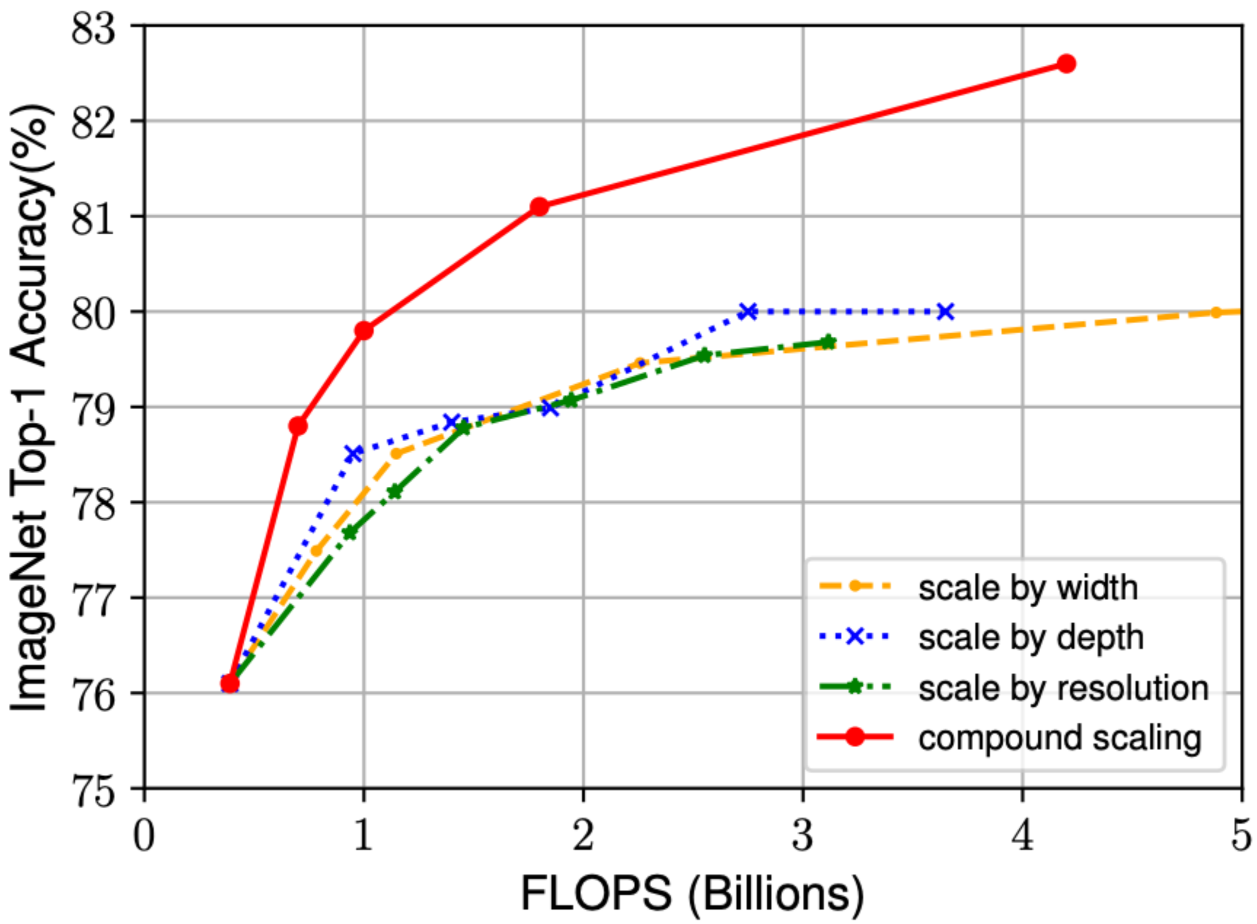


Image source: Tan, Mingxing, and Quoc V. Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." arXiv preprint arXiv:1905.11946 (2019).

EfficientNet

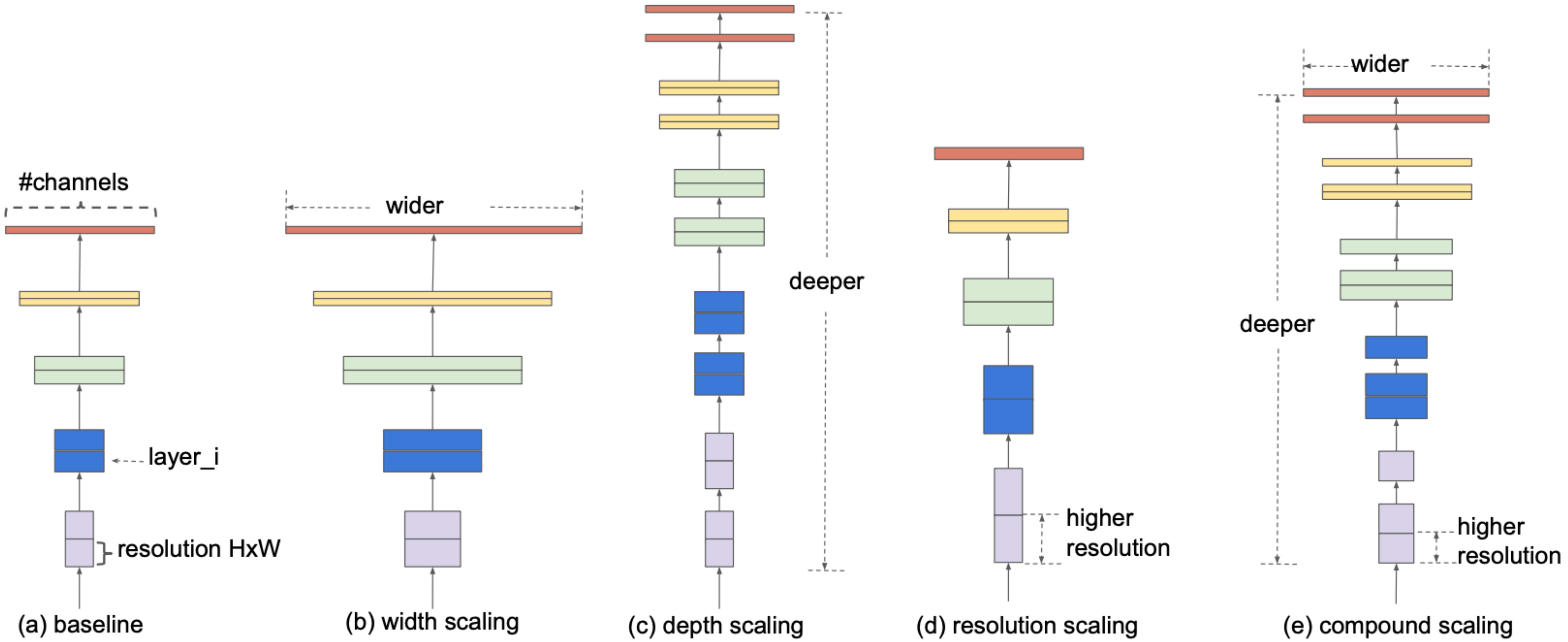


Image source: Tan, Mingxing, and Quoc V. Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." arXiv preprint arXiv:1905.11946 (2019).

PRE-TRAINED MODELS

Transfer Learning

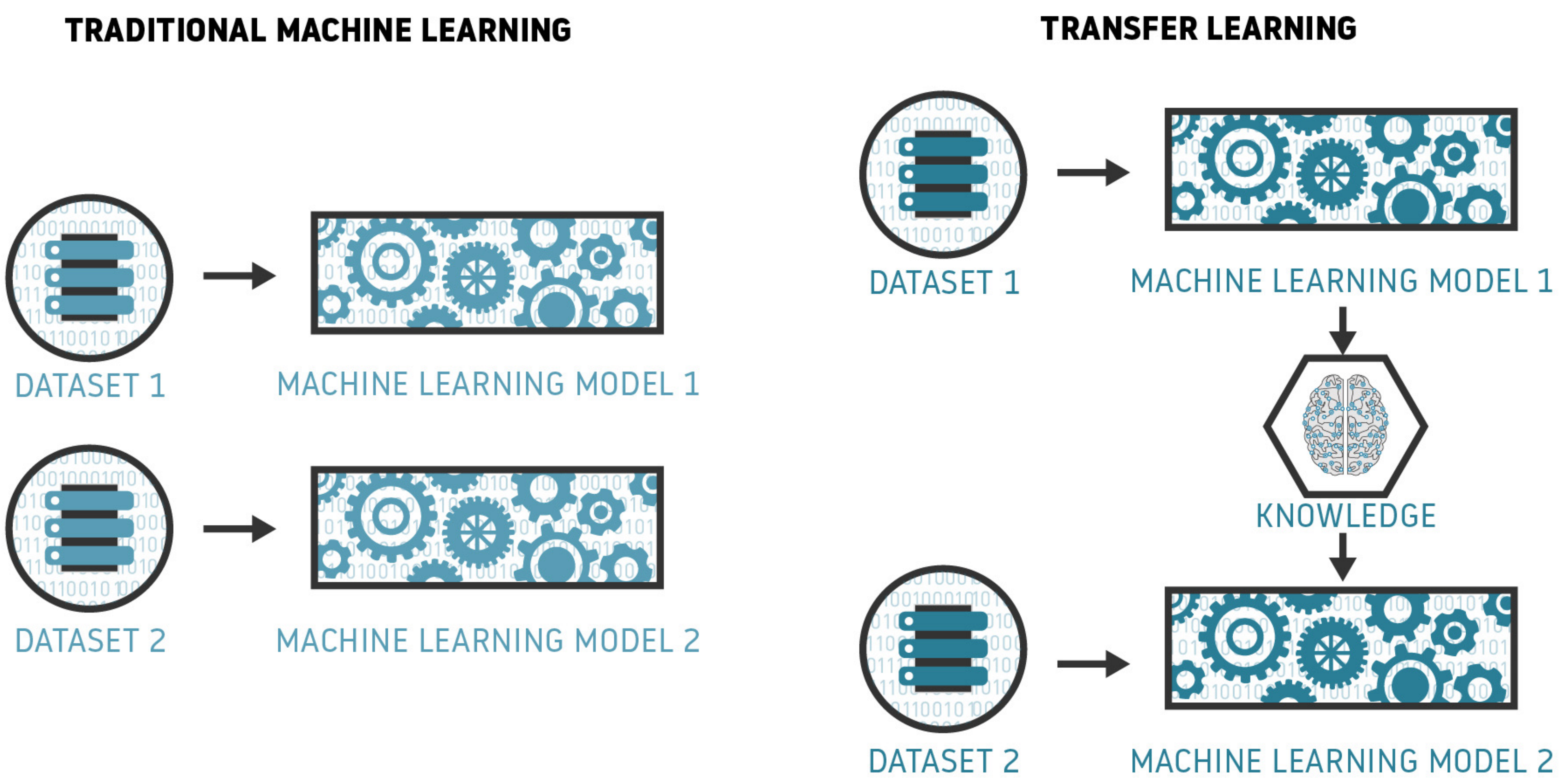


Image source: https://datascience.aero/transfer-learning-aviation/

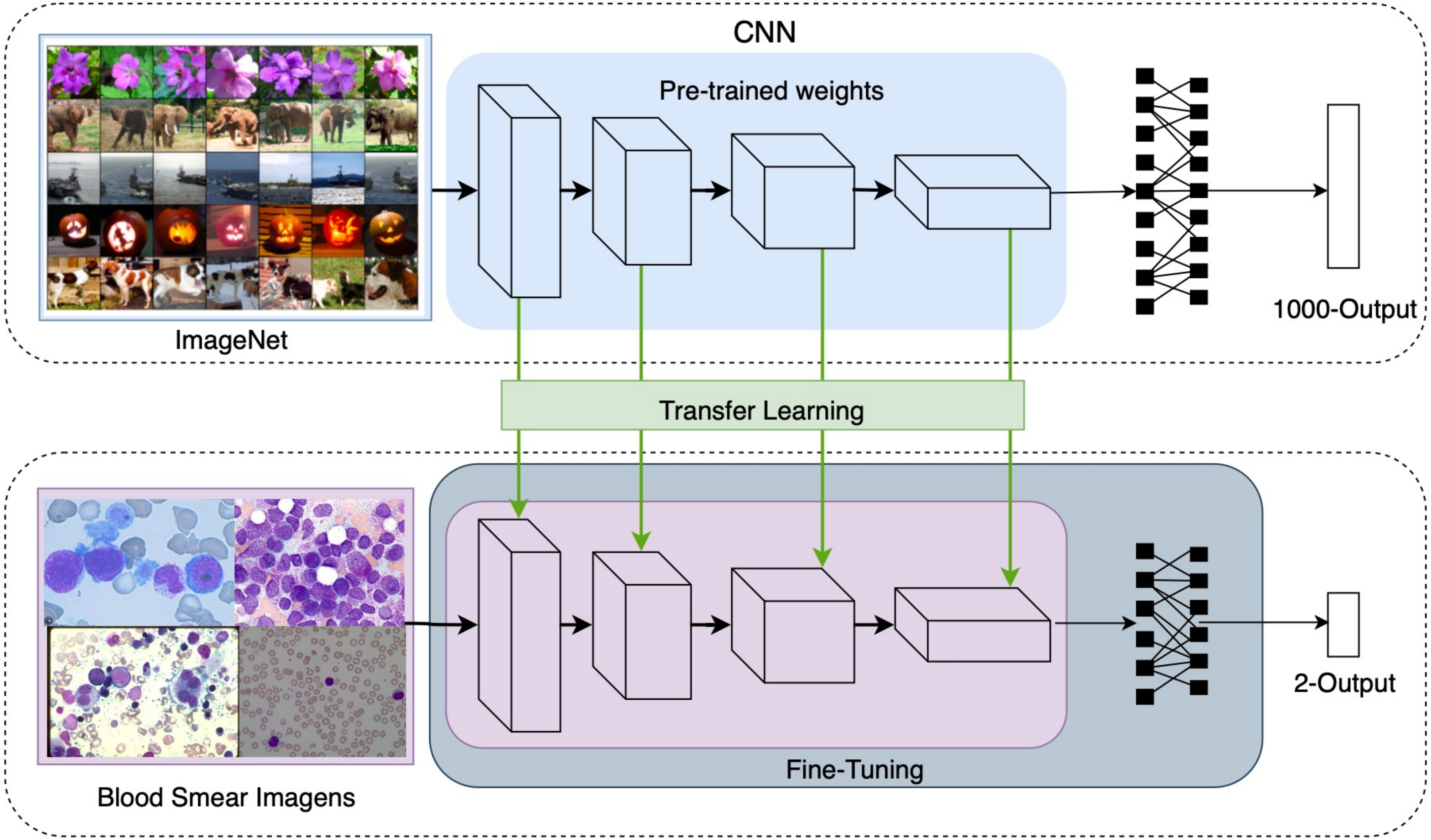
Pre-Trained Models

- In deep learning, transfer learning is usually expressed through the use of pre-trained models.
- A pre-trained model is a model that was trained on a large benchmark dataset to solve a problem similar to the one that we want to solve.
- Accordingly, due to the computational cost of training such models, it is common practice to import and use models from published literature.
  - E.g. VGG, GoogLeNet, ResNet, MobileNet…

Fine-Tuning

Source domain

Target domain



Fine-Tuning Strategies

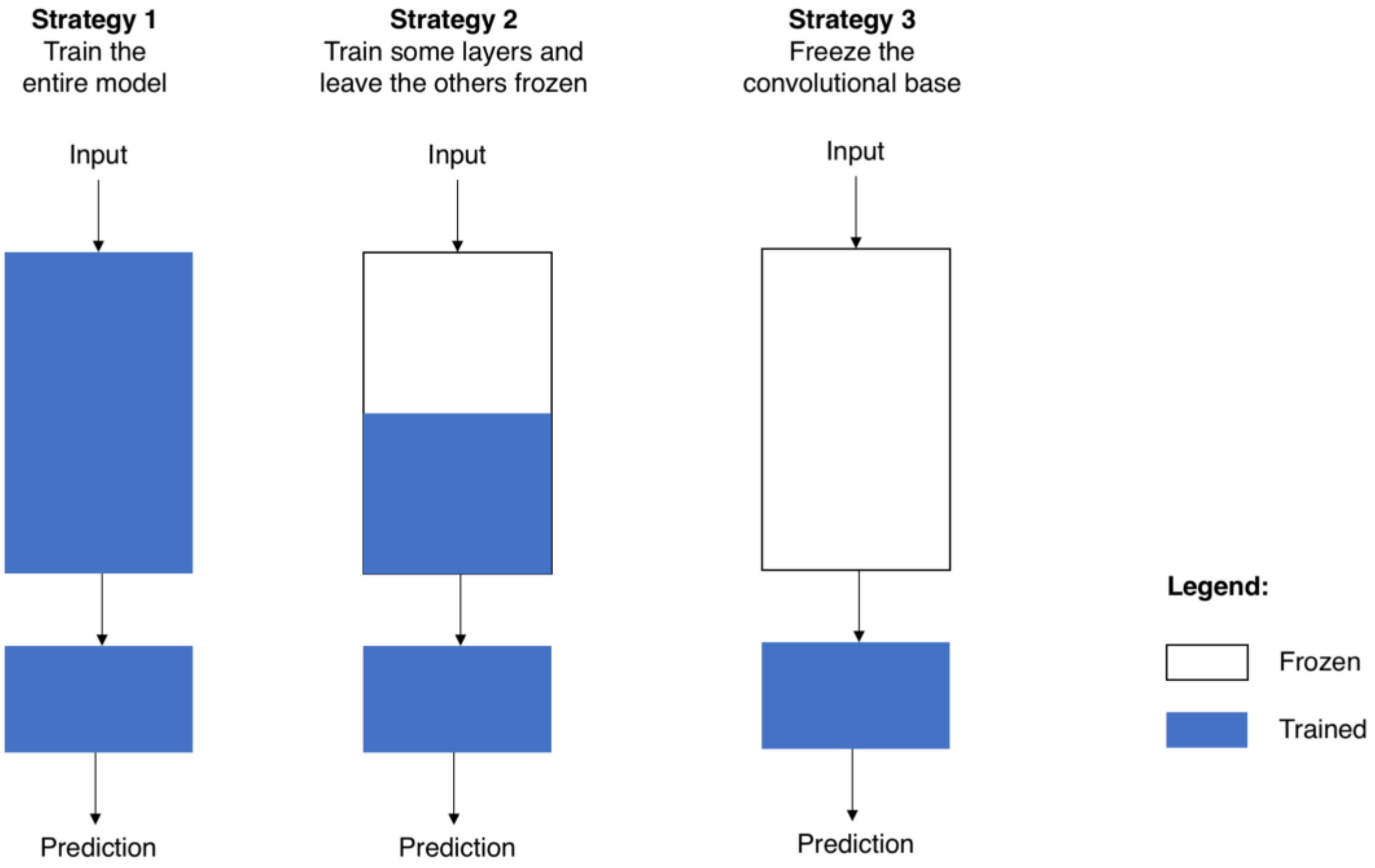


Image source: https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751

Fine-Tuning Strategies

- Choose fine-tuning strategy based on your target dataset.

Data similarity / Data amount	Similar	Different
Little	Finetune linear classifier on top layer	You're in trouble... Try data augmentation / collect more data
Large	Finetune a few layers	Finetune a larger number of layers

Conclusion

After this lecture, you should know:

- How are CNN models evolved from LeNet to DenseNet.
- Why are deeper networks perform better than shallower networks?
- Why do convolution filters with small size perform better than the ones with large size?
- What is the usage of 1x1 convolution filter?
- Why does residual information help learning?

Suggested Reading

- AlexNet paper
- VGG paper
- GoogLeNet paper
- ResNet paper
- SENet paper
- DenseNet paper

Assignment 2

- Assignment 2 is released. The deadline is 18:00, 28th October.

Thank you!

- Any question?
- Don’t hesitate to send email to me for asking questions and discussion. ☺