Computer Vision

第九周 图像识别

庞彦
yanpang@gzhu.edu.cn

01

Object Recognition

图像识别初探究

Object Recognition

目标识别：　分类问题

Class 1:　笔记本电脑；
Class 2:　台式机电脑；
Class 3:　平板电脑。

Object Recognition

目标识别：　　分类问题

对于人类：

共同点：

都是电脑…

不同点：

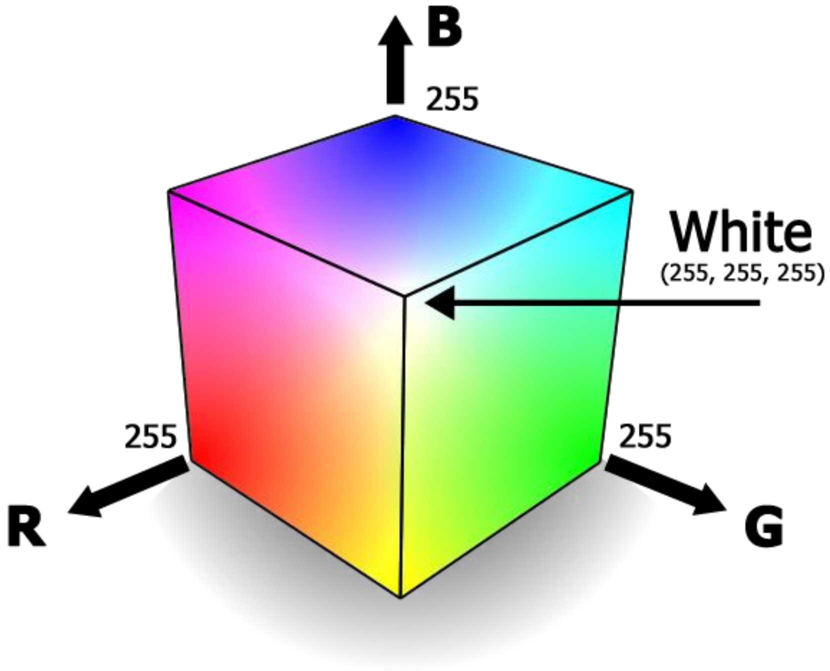
大小？形状？材质？便携性…

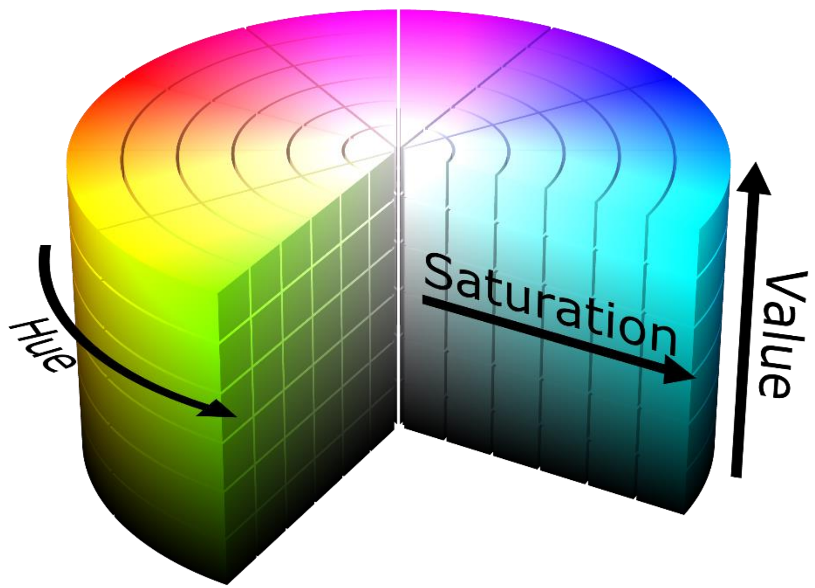
Features

计算机通过学习并挖掘目标属性的相关**特征**（Features）来对目标进行识别与探究。

Features

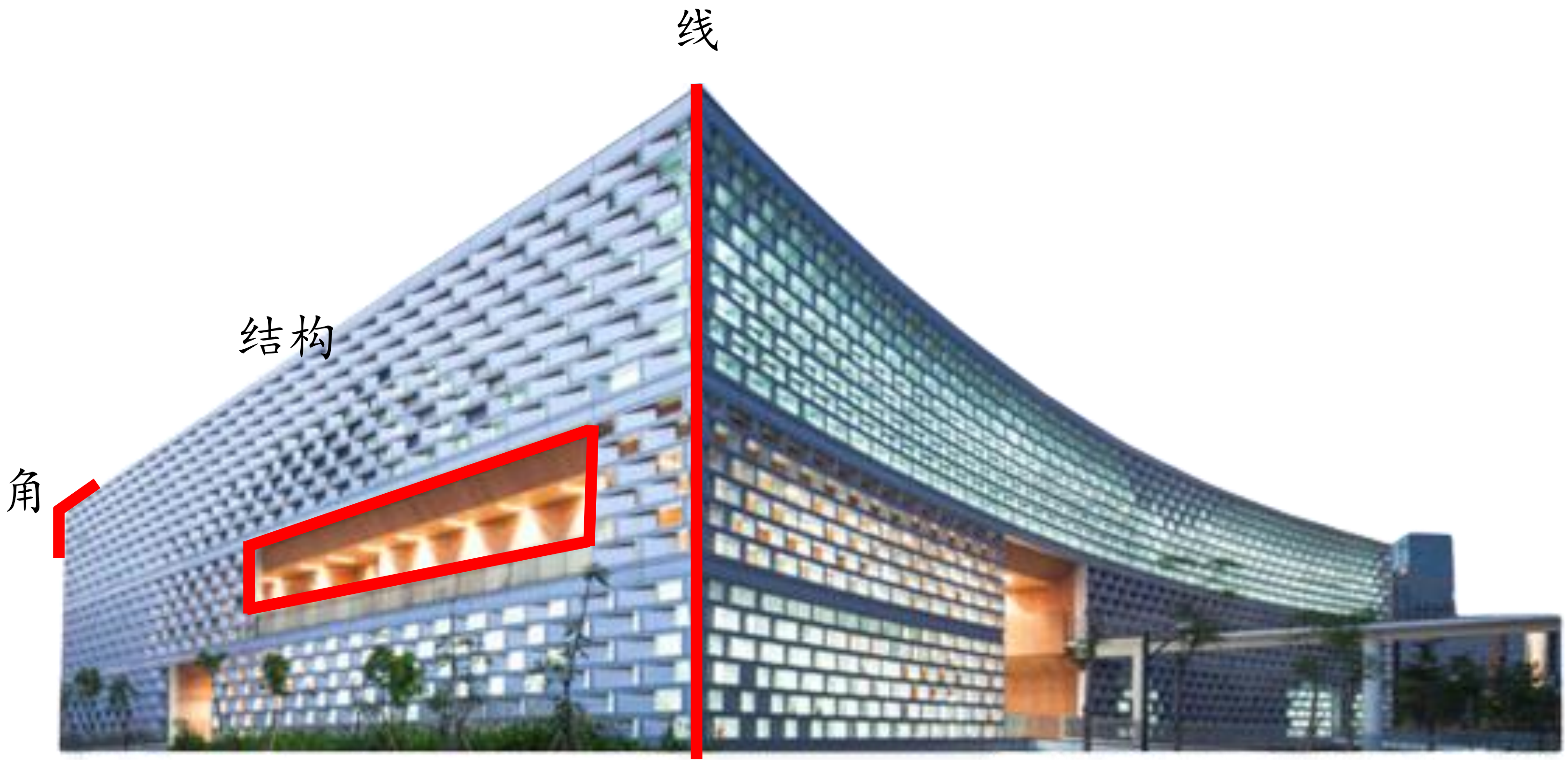
颜色特征





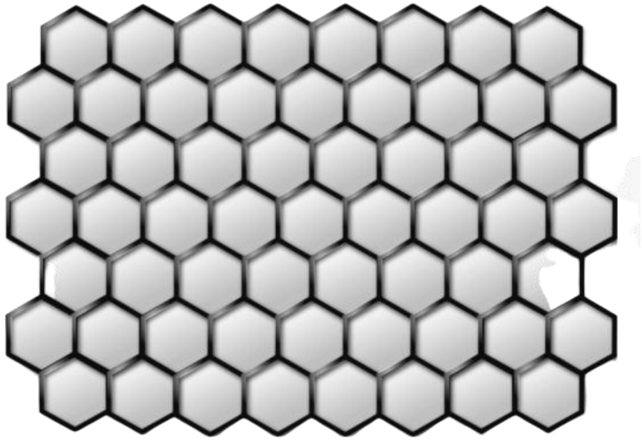
Features

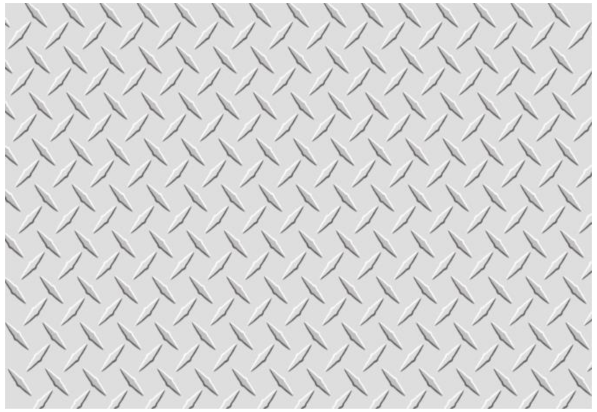
颜色特征
形状特征



Features

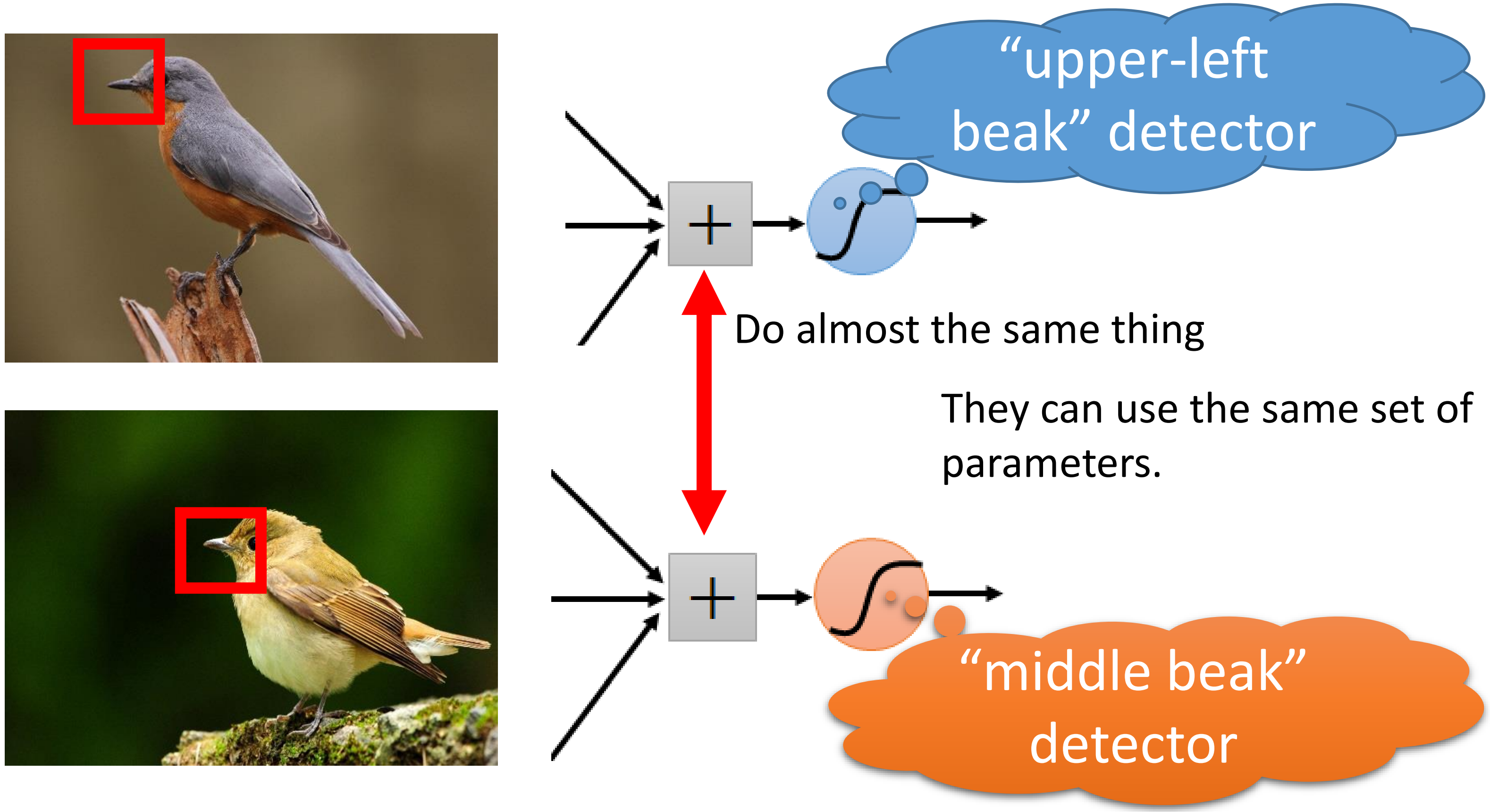
颜色特征
形状特征
纹理特征
… …





Why CNN for Image

- The same patterns appear in different regions.



Why CNN for Image

✓**Subsampling** the pixels will not change the object;

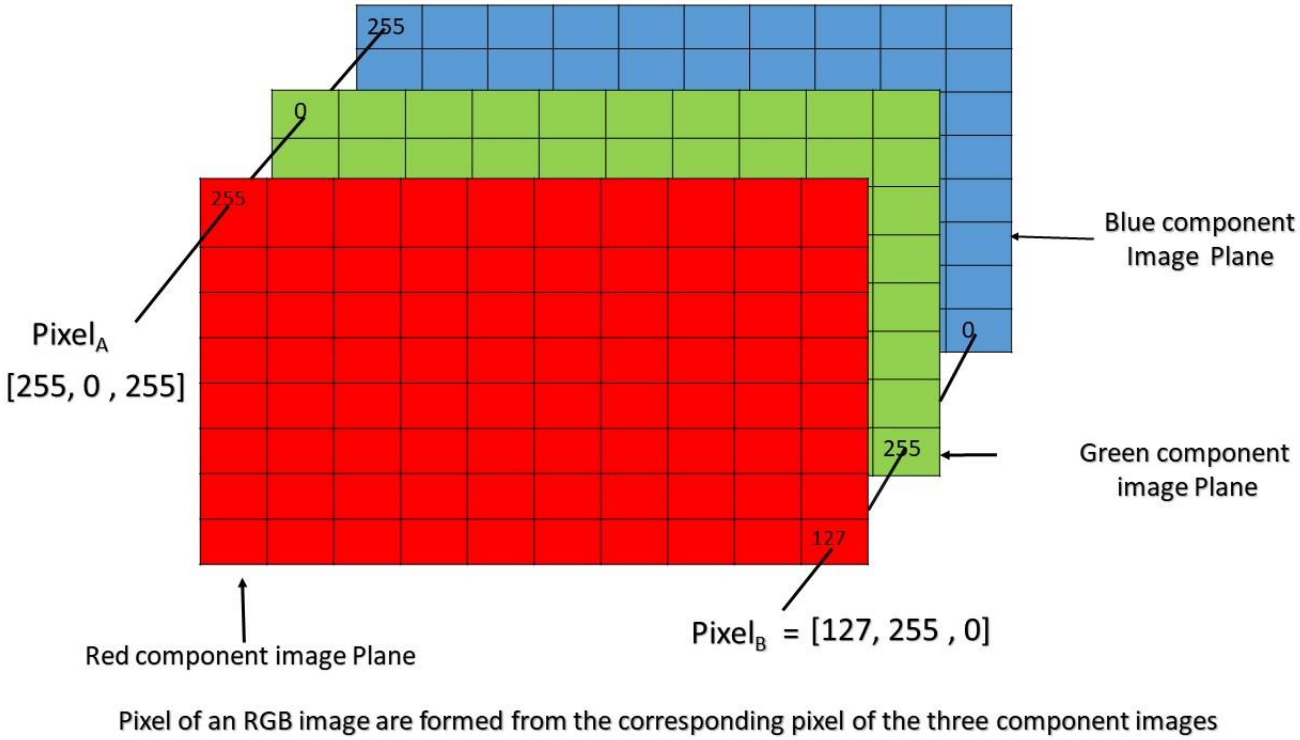
✓It just make the image smaller.

parrot

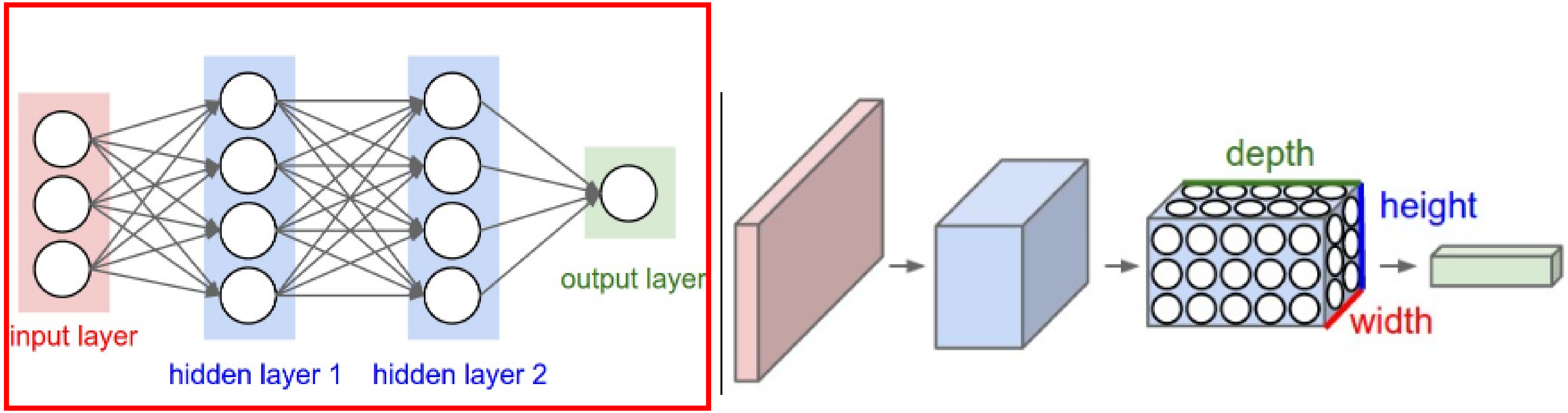
parrot

Less parameters for the network to process the image

Color Image： 3 channels



Convolutional Neural Networks （CNNs）



Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

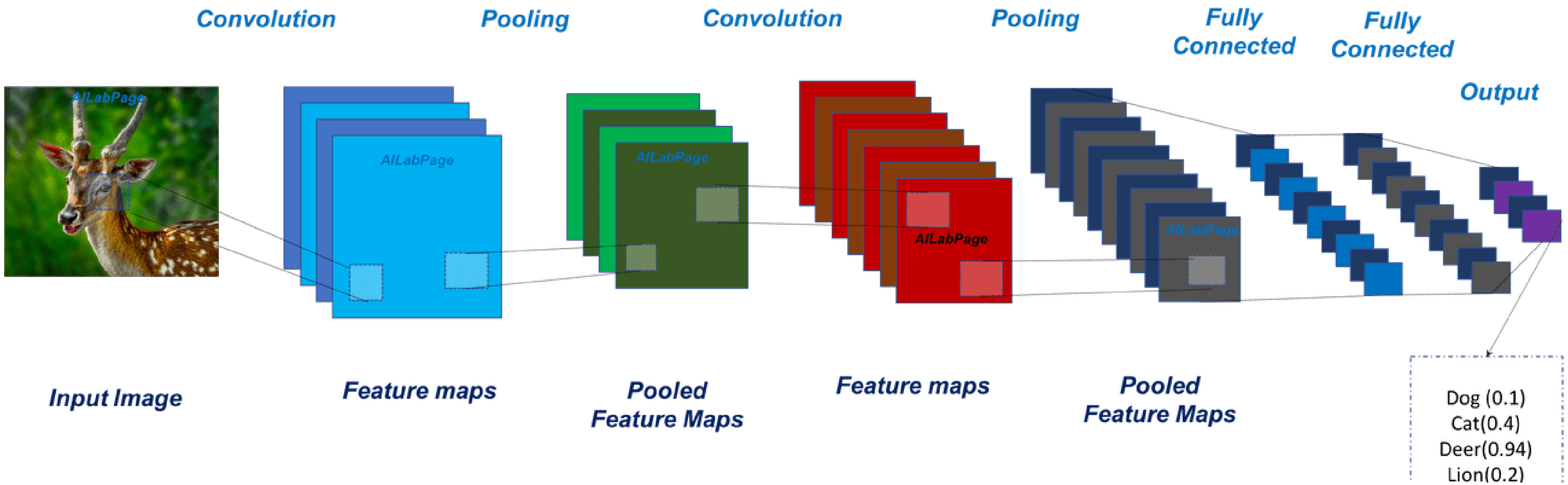
Convolutional Neural Networks（CNNs）

卷积层	Convolutional Layer
池化层	Pooling Layer
全连接层	Fully-Connected Layer

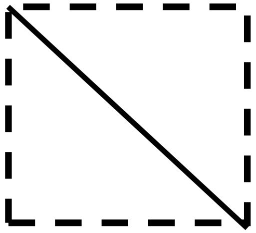
We will stack these layers to form a full ConvNet architecture.

[INPUT - CONV - POOL – FC - OUTPUT]

Convolutional Neural Networks （CNNs）



Convolution Layer: The Kernel

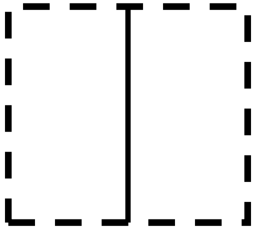


1	-1	-1
-1	1	-1
-1	-1	1

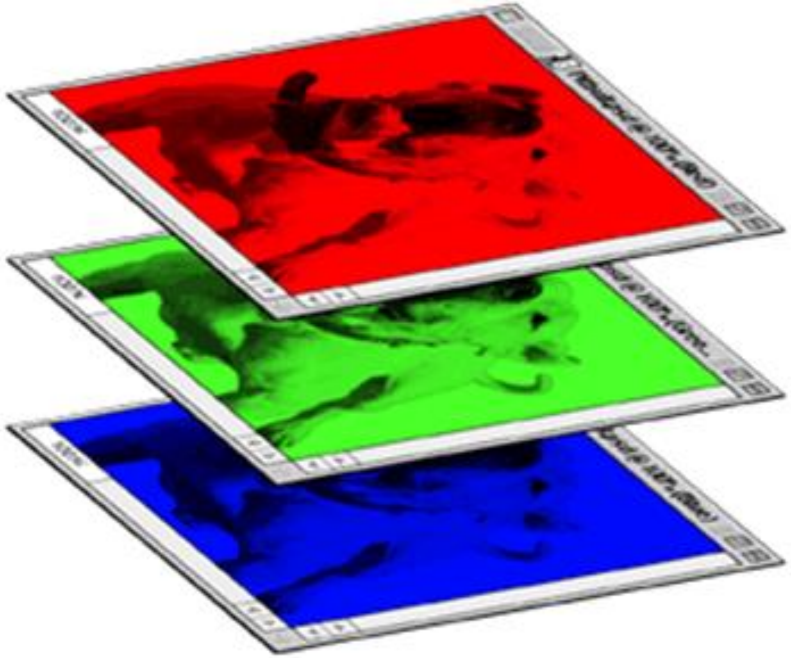
Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

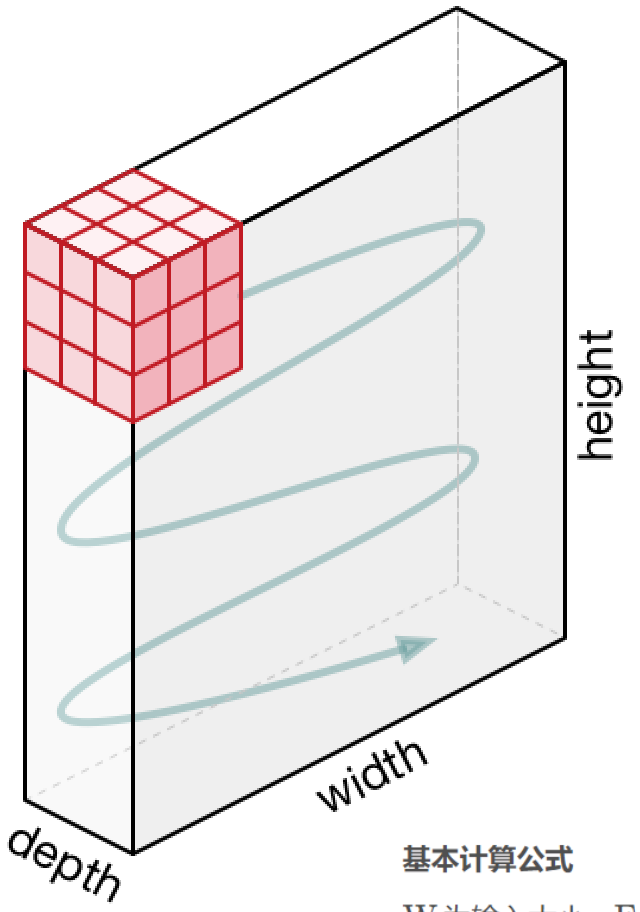


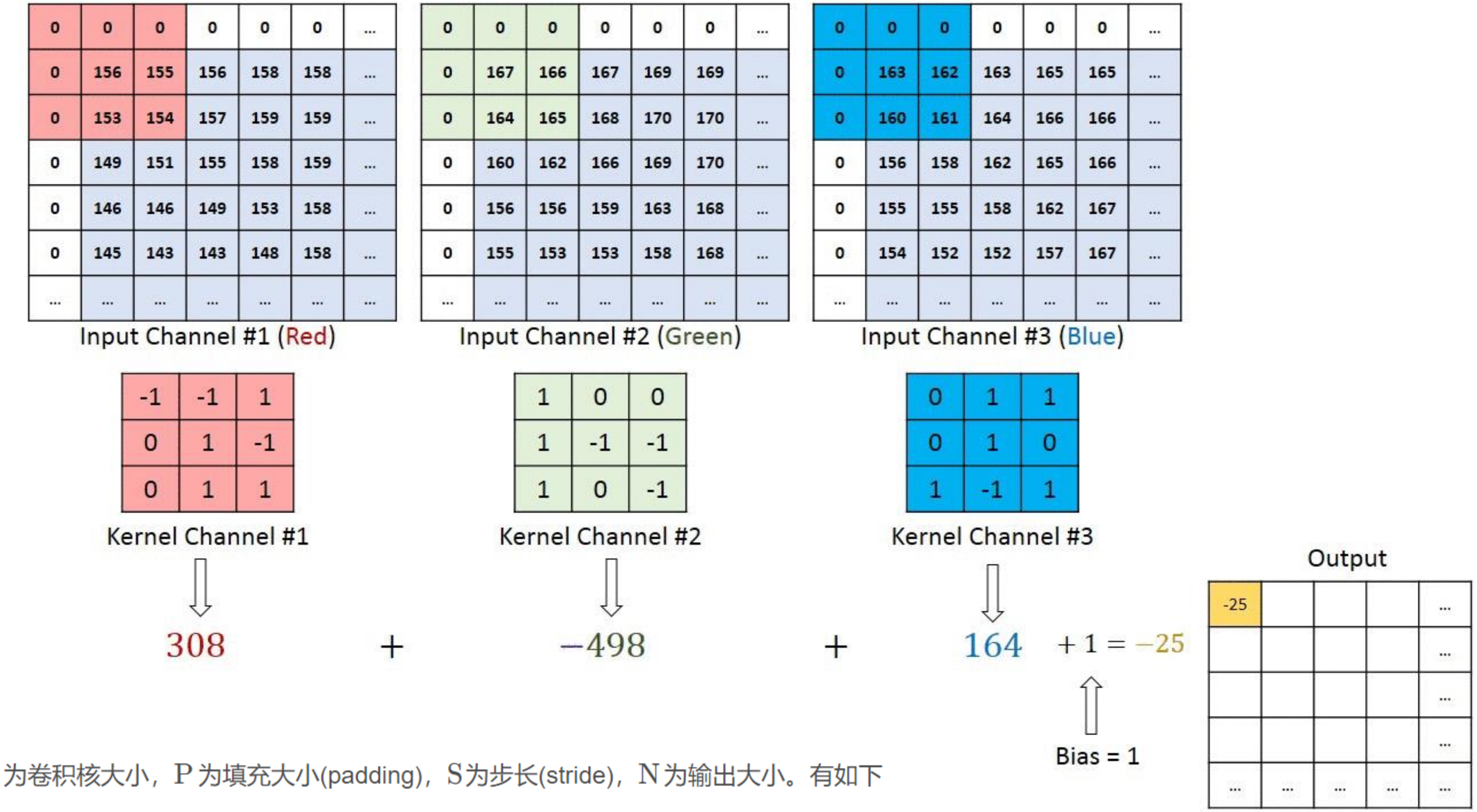
Colorful image



1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Convolution Layer: The Kernel



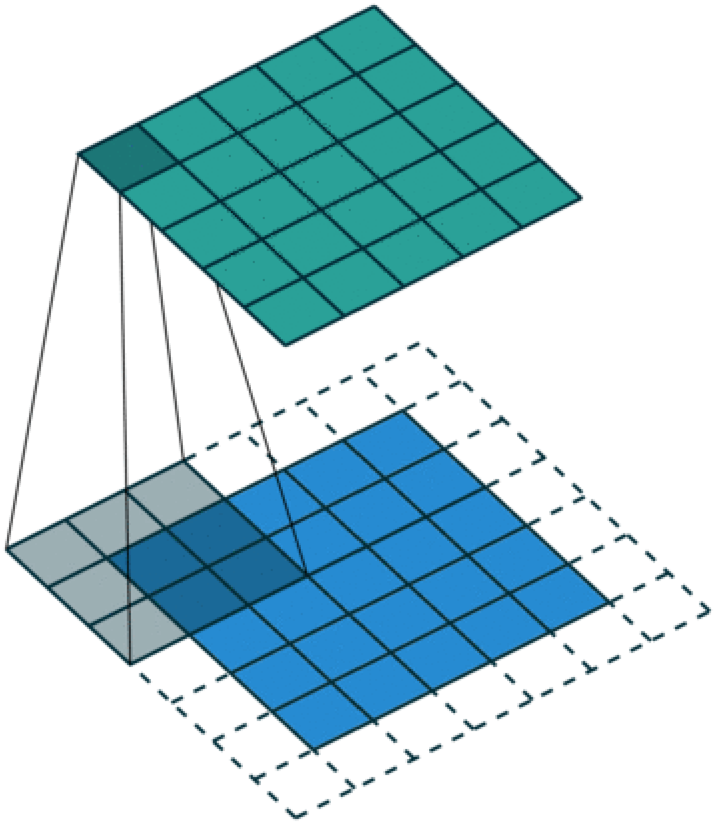


基本计算公式

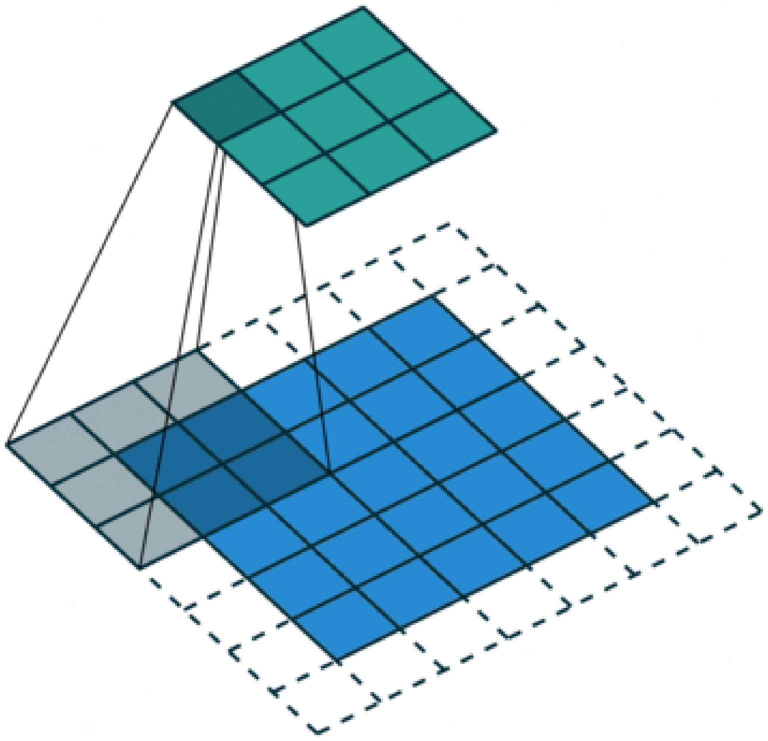
W 为输入大小，F 为卷积核大小，P 为填充大小(padding)，S为步长(stride)，N为输出大小。有如下计算公式：

$$N = \frac{(W - F + 2P)}{S} + 1$$

Convolution Layer: Padding and Stride

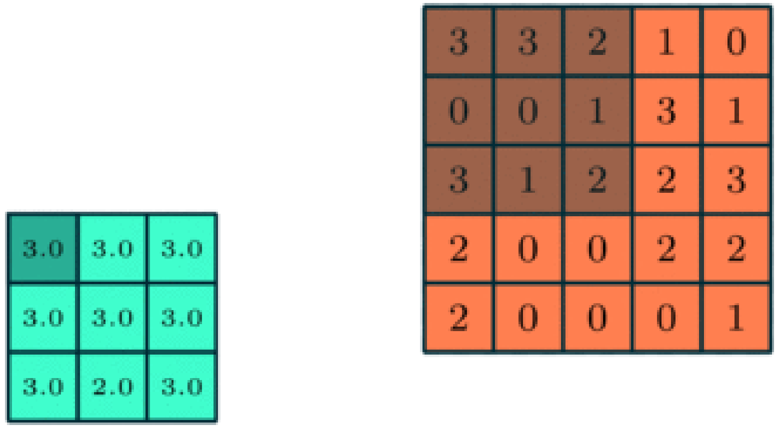


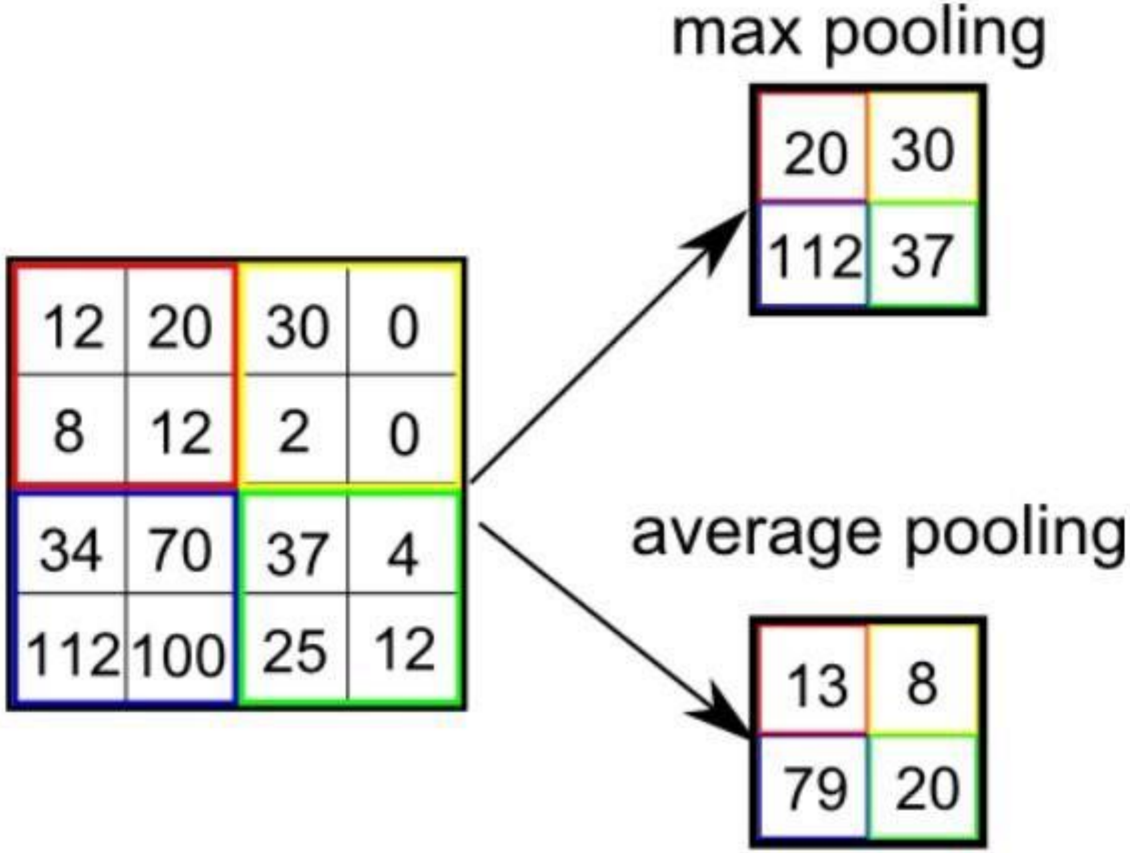
padding = 1
stride = 1



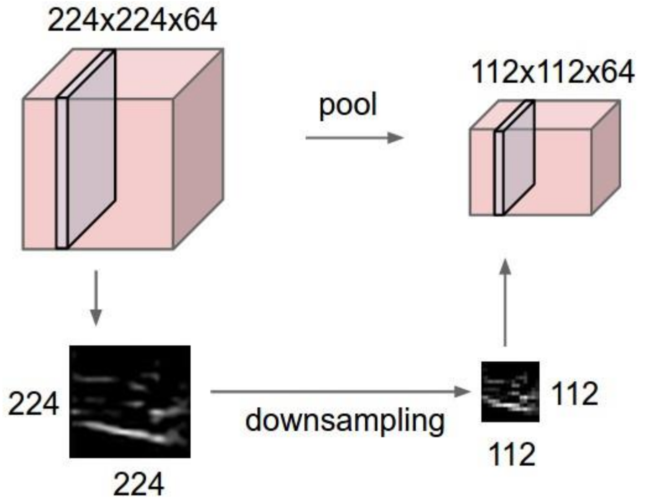
padding = 1
stride = 2

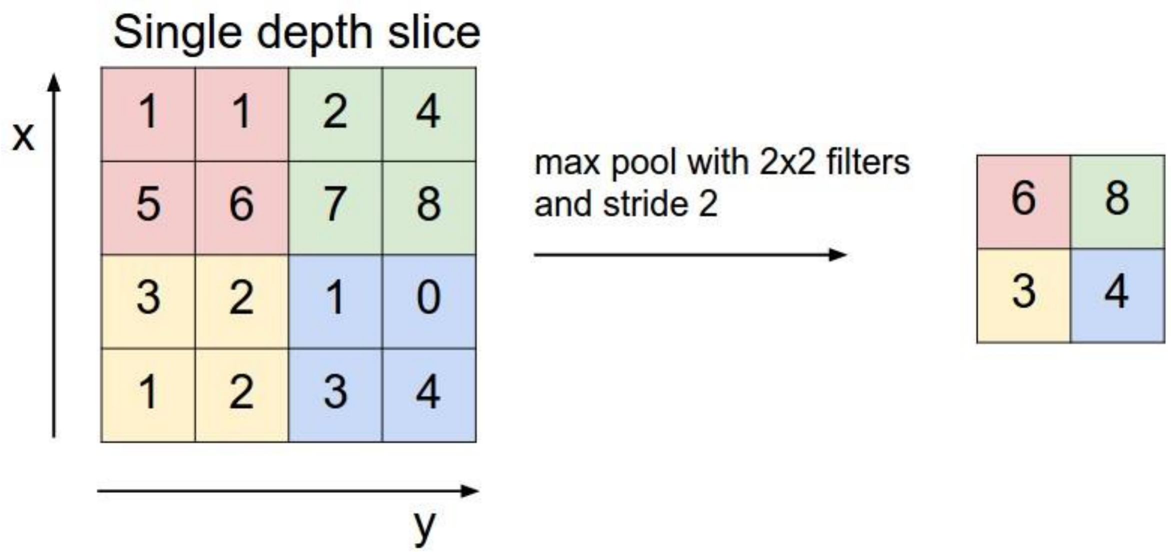
Pooling Layer



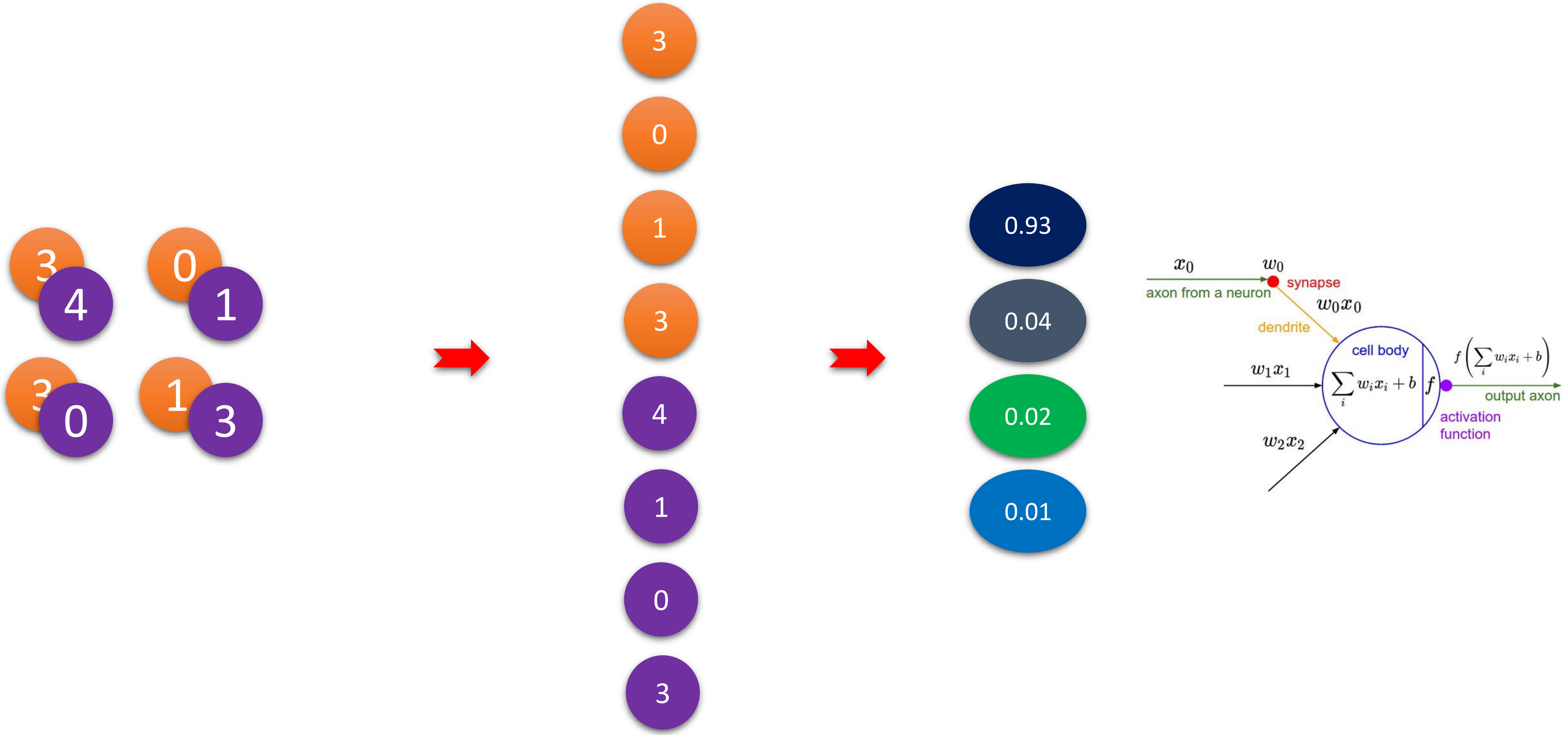


Pooling Layer

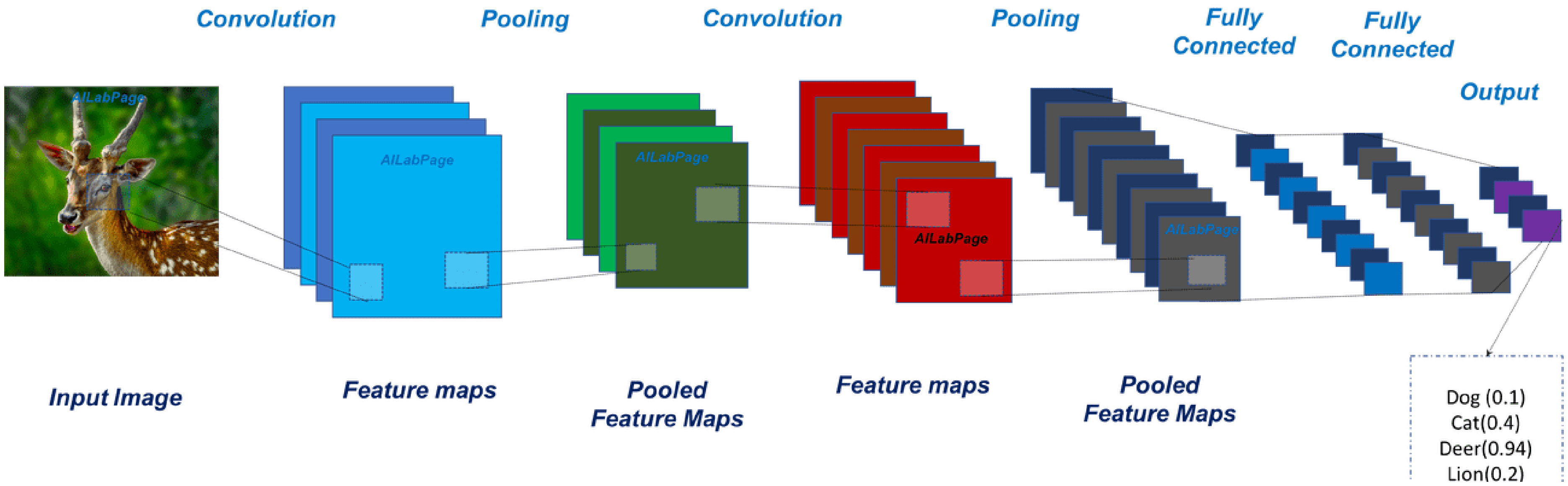




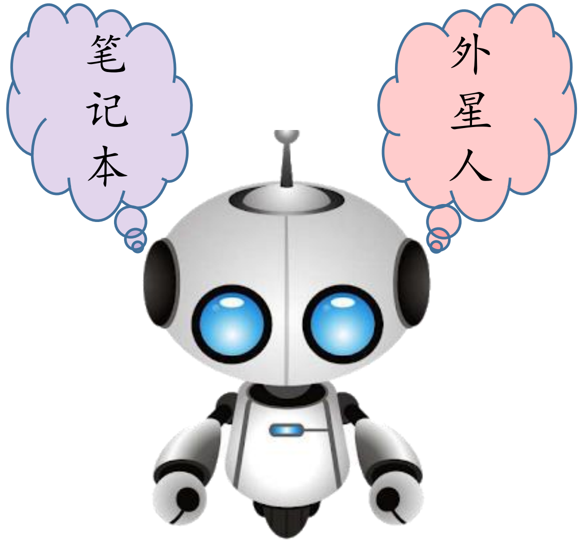
Flatten


3
4
0
1
3
0
1
3
3
0
1
3
4
1
0
3
0.93
0.04
0.02
0.01
x_0
w_0
synapse
axon from a neuron
$w_0 x_0$
dendrite
cell body
$w_1 x_1$
$\sum_i w_i x_i + b$
f
$f\left(\sum_i w_i x_i + b\right)$
output axon
activation
function
$w_2 x_2$

Review: Convolutional Neural Networks （CNNs）

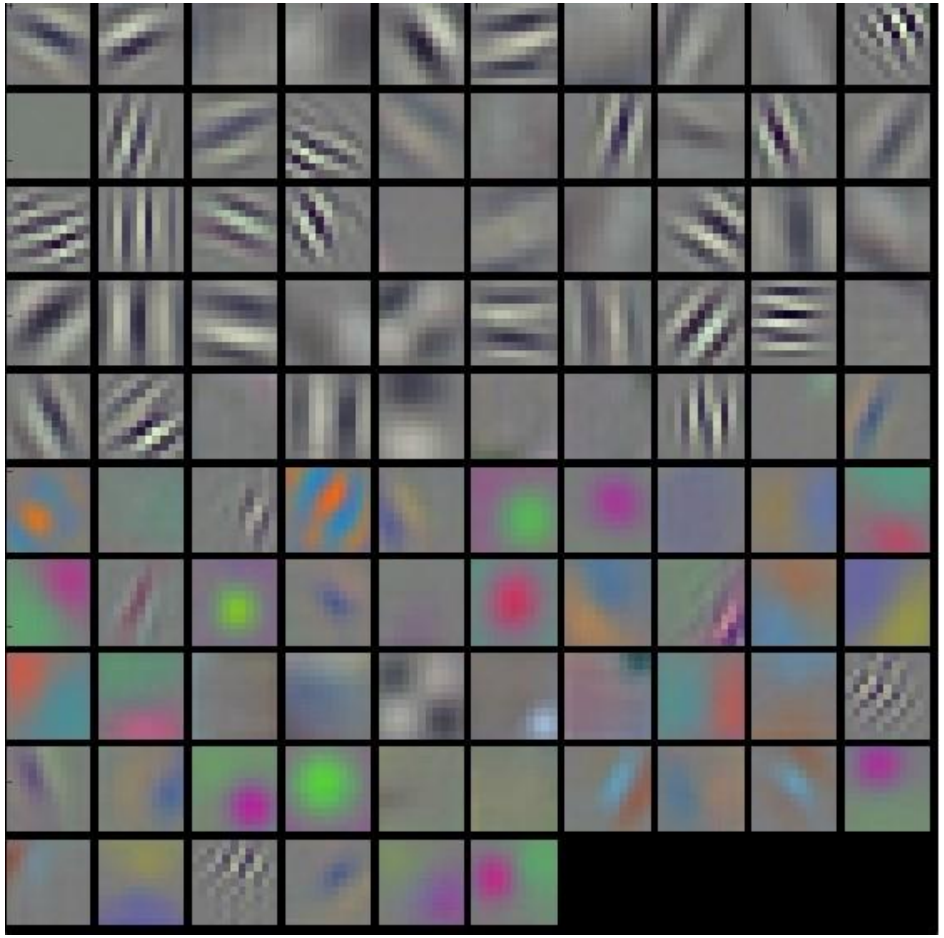


What does the machine learn?



Features

- Typical-looking filters on the trained first layer



11 x 11
(AlexNet)

Features

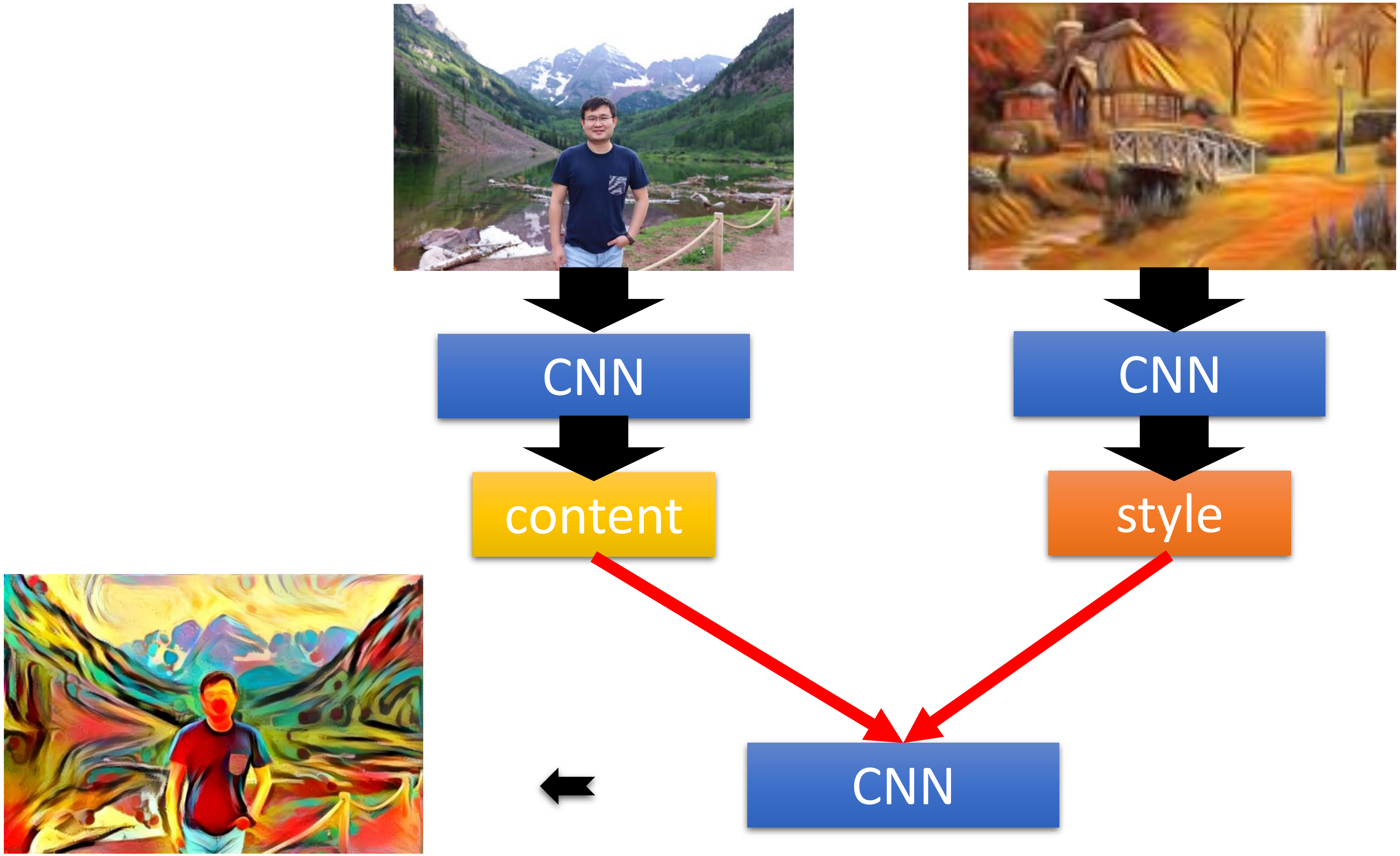
Maximally activating images for some POOL5 (5th pool layer) neurons of an AlexNet.

More Applications: Deep Dream

+

=

Given a photo, make its style like famous paintings



02

VGG → CSPNet

从VGG到CSPN

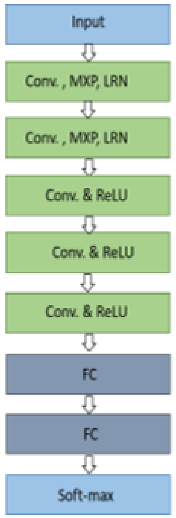
Deep Convolutional Neural Netwc

19 layers

22 layers

8 layers

Inaccuracy: 16.4%

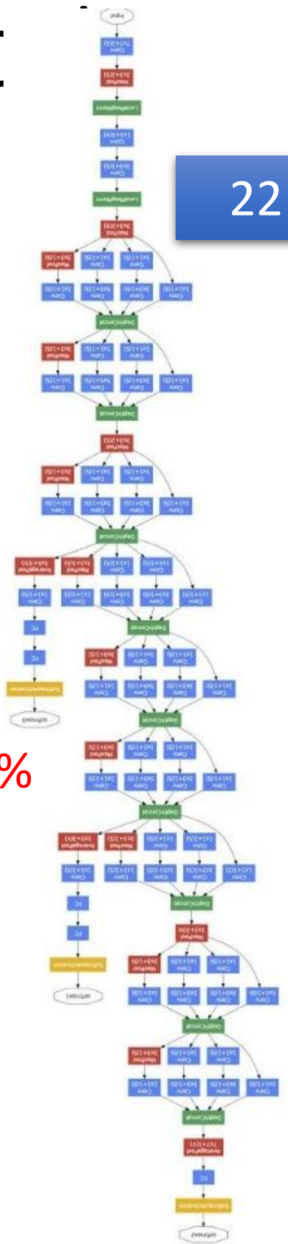


AlexNet (2012)

7.3%

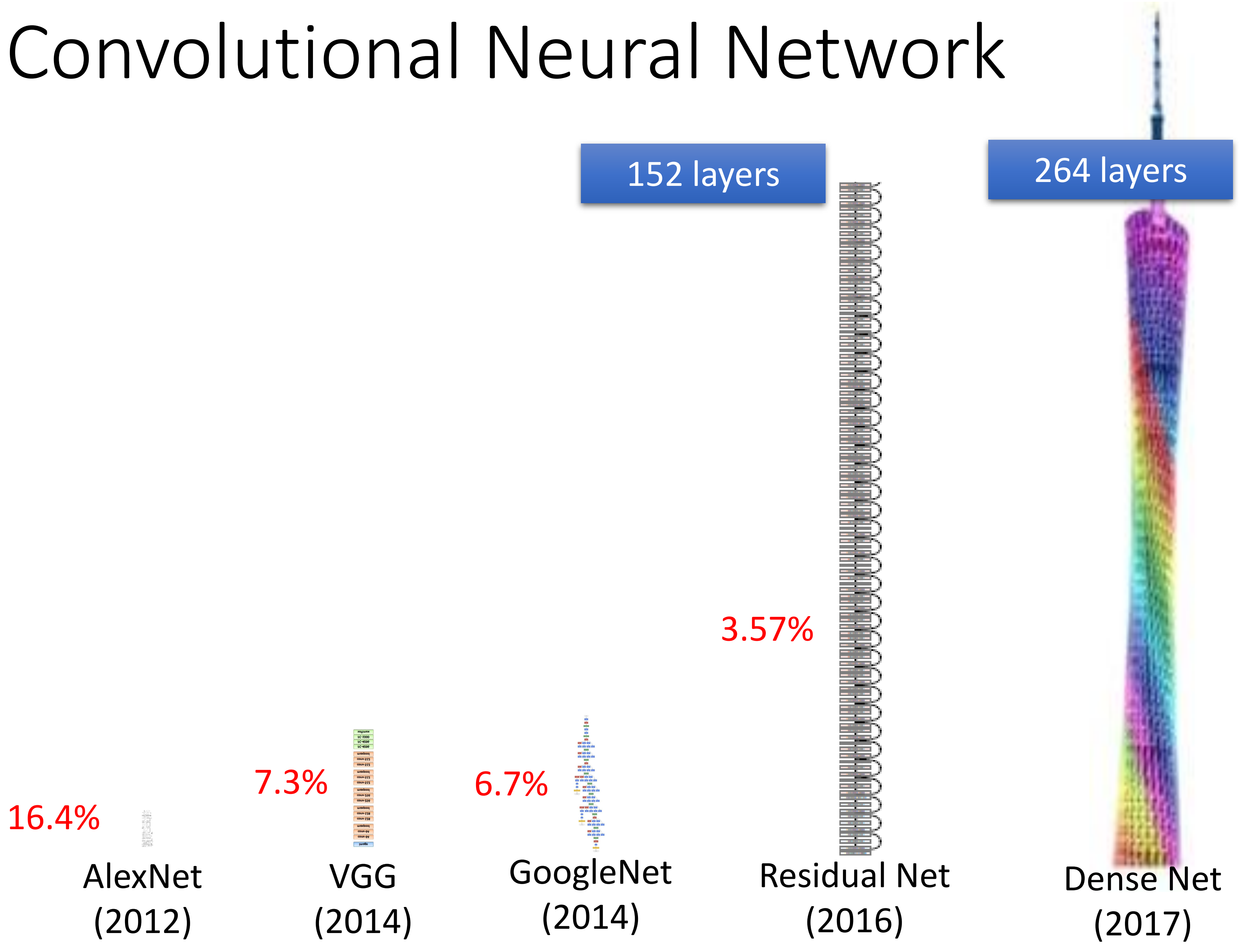
VGG (2014)

6.7%

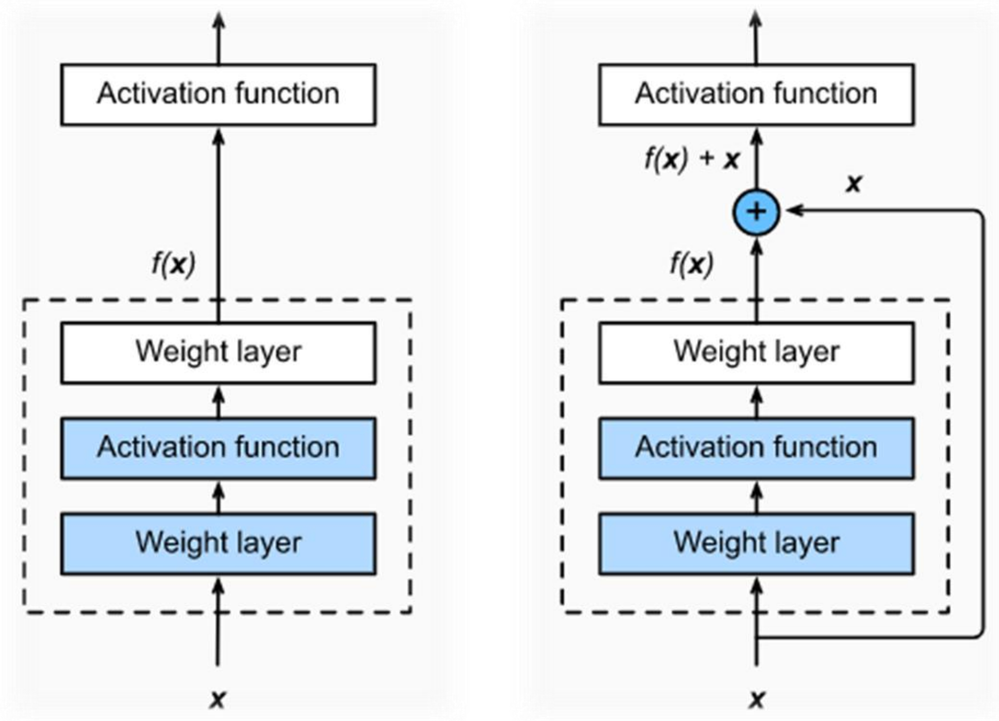


GoogleNet (2014)

Deep Convolutional Neural Network

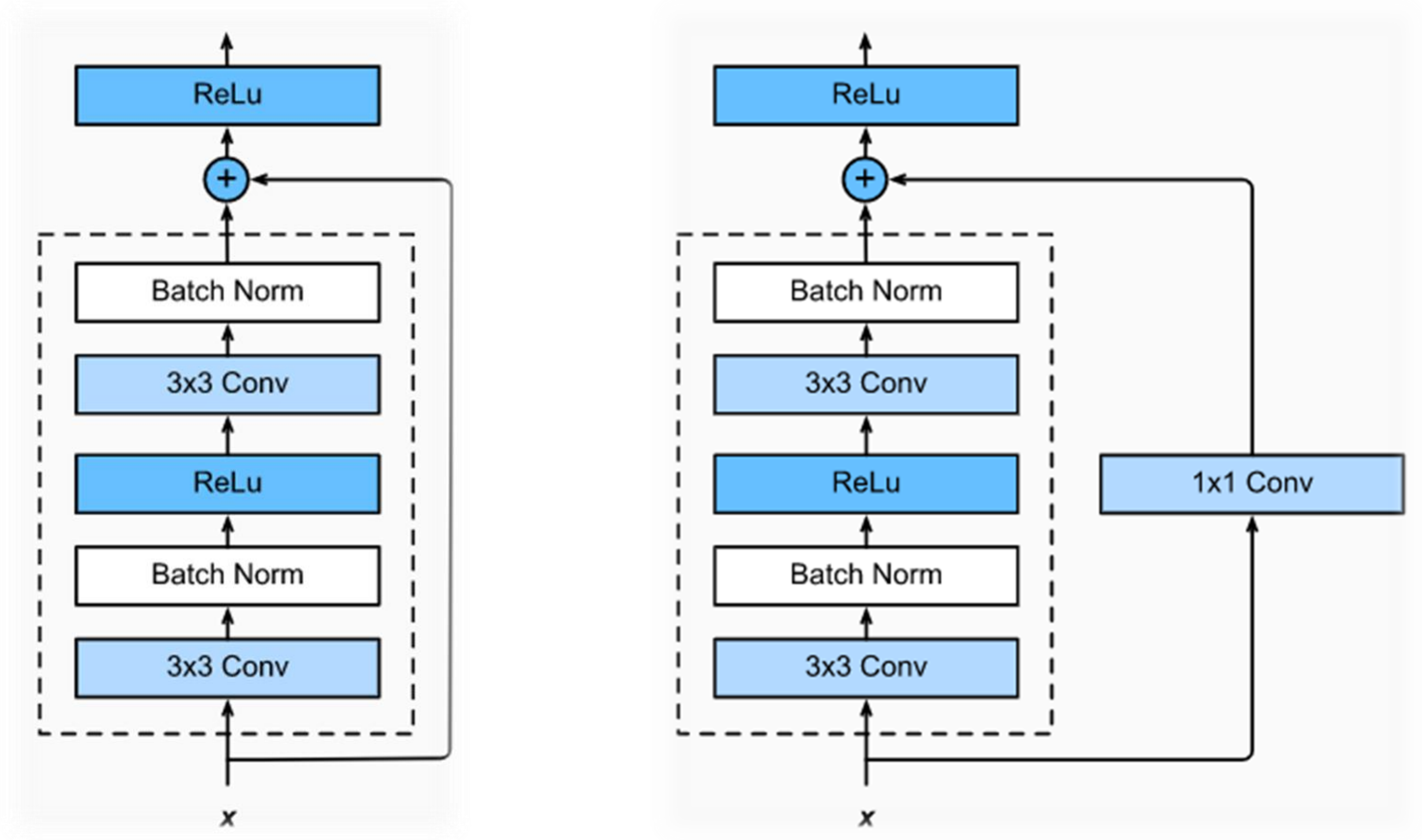


ResNet



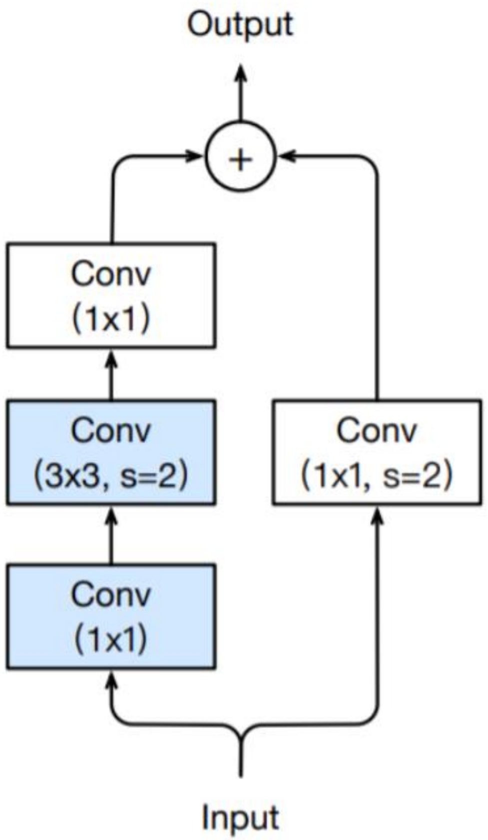
The difference between a regular block (left) and a residual block (right).

ResNet

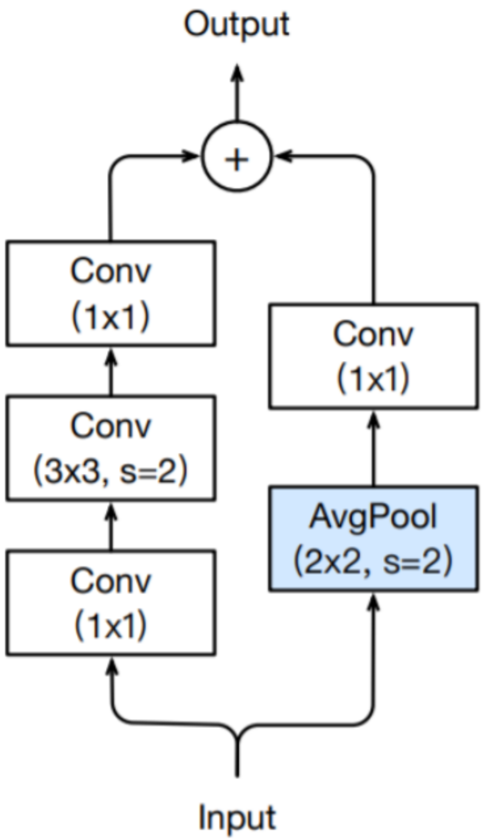


Left: regular ResNet block; Right: ResNet block with 1x1 convolution

ResNet

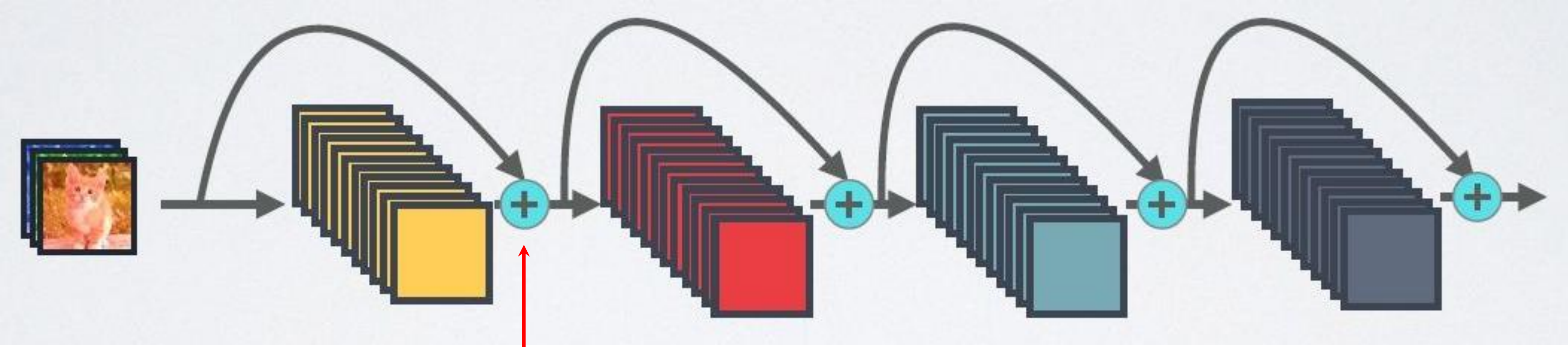


(a) ResNet-B



(c) ResNet-D

ResNet

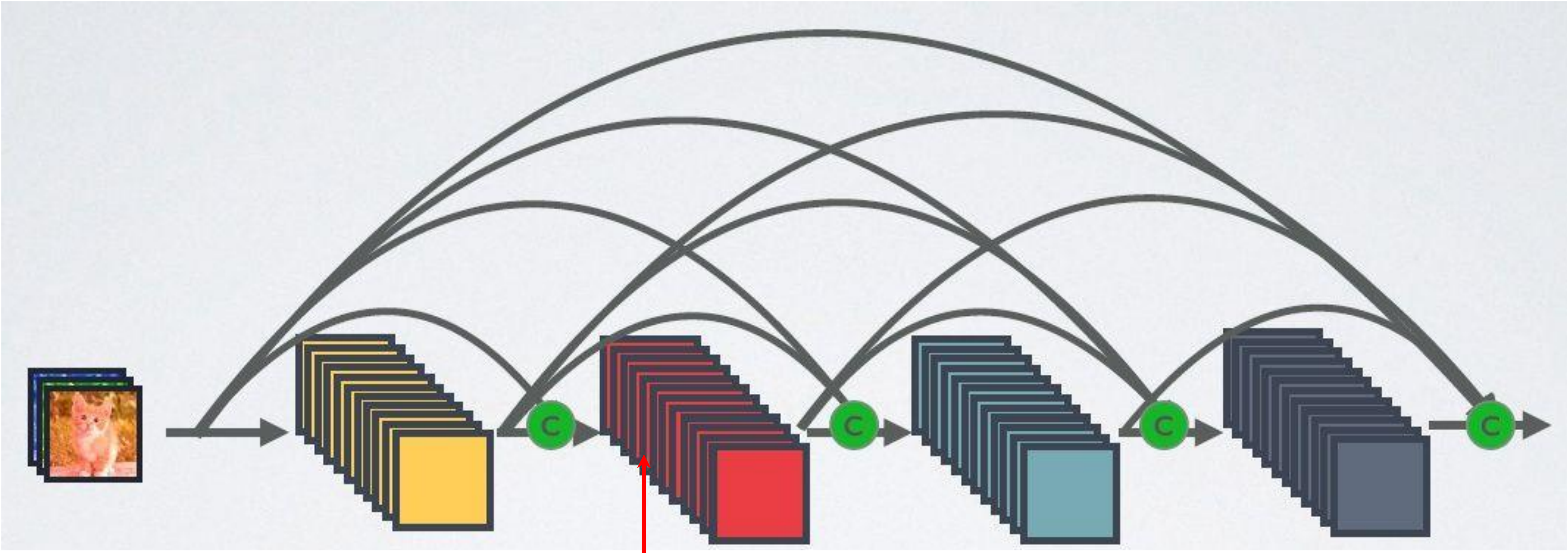


Element-wise Addition

1、减轻了vanishing-gradient
2、加强了feature的传递

Kaiming He, 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 (CVPR), pages 770–778, 2016.

DenseNet



Channel-wise Concatenation

1、减轻了vanishing-gradient
2、加强了feature的传递
3、更有效地利用了feature
4、一定程度上较少了参数数量

Gao Huang, 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 (CVPR), pages 4700–4708, 2017

DenseNet： Dense Block

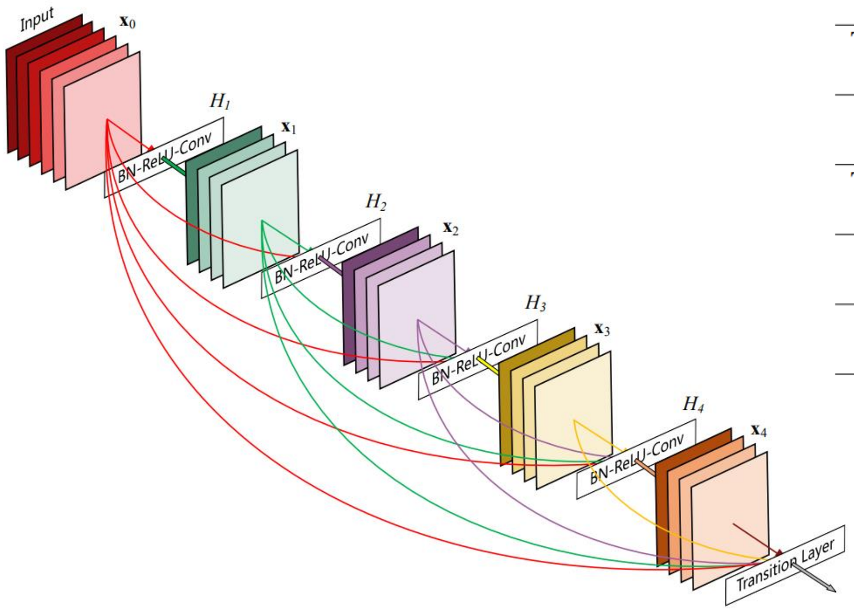
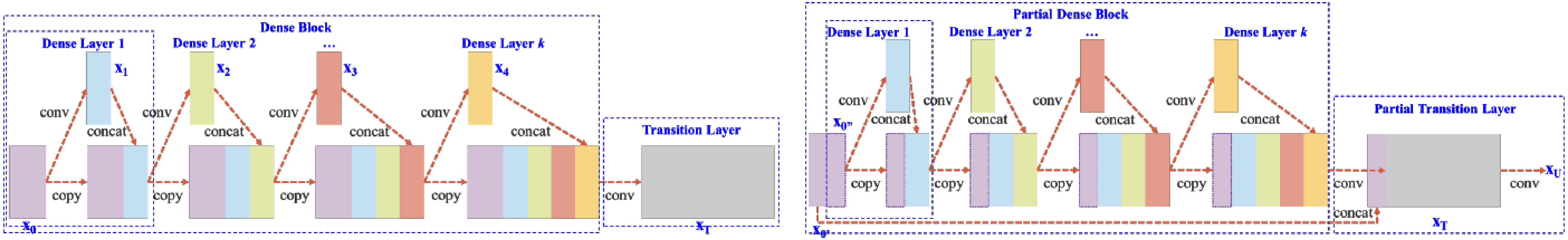


Figure 1: A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

CSPNet

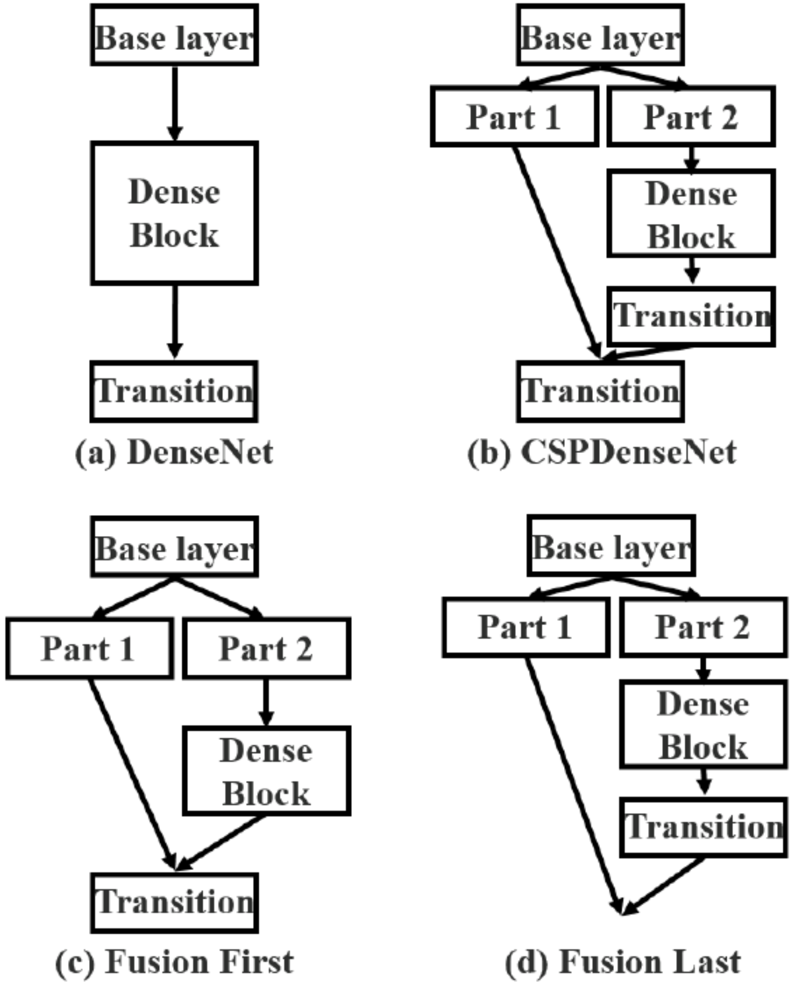


(a) DenseNet (b) Cross Stage Partial DenseNet

Figure 2: Illustrations of (a) DenseNet and (b) our proposed Cross Stage Partial DenseNet (CSPDenseNet). CSPNet separates feature map of the base layer into two part, one part will go through a dense block and a transition layer; the other one part is then combined with transmitted feature map to the next stage.

Wang, Chien-Yao, et al. "CSPNet: A New Backbone that can Enhance Learning Capability of CNN." arXiv preprint arXiv:1911.11929 (2019).

CSPNet


Base layer
Dense
Block
Transition
(a) DenseNet
Base layer
Part 1
Part 2
Dense
Block
Transition
Transition
(b) CSPDenseNet
Base layer
Part 1
Part 2
Dense
Block
Transition
(c) Fusion First
Base layer
Part 1
Part 2
Dense
Block
Transition
(d) Fusion Last

CSPNet

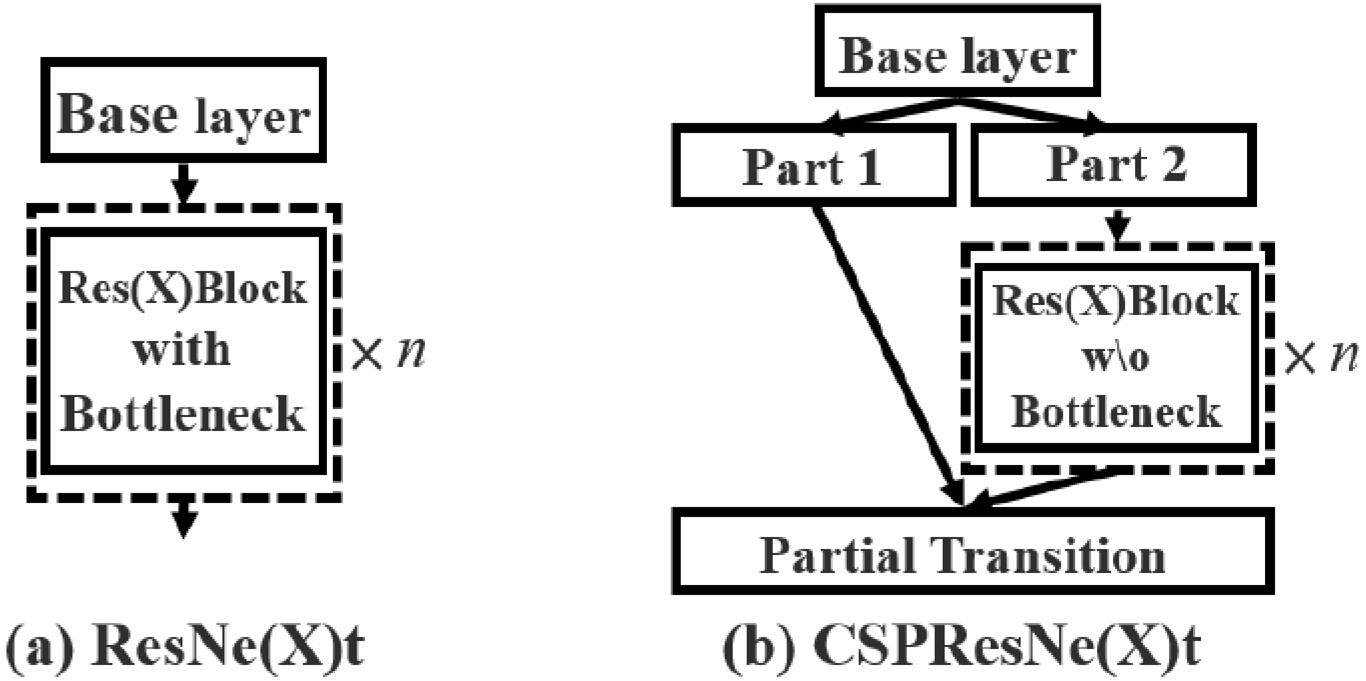


Figure 5: Applying CSPNet to ResNe(X)t.

CSPNet

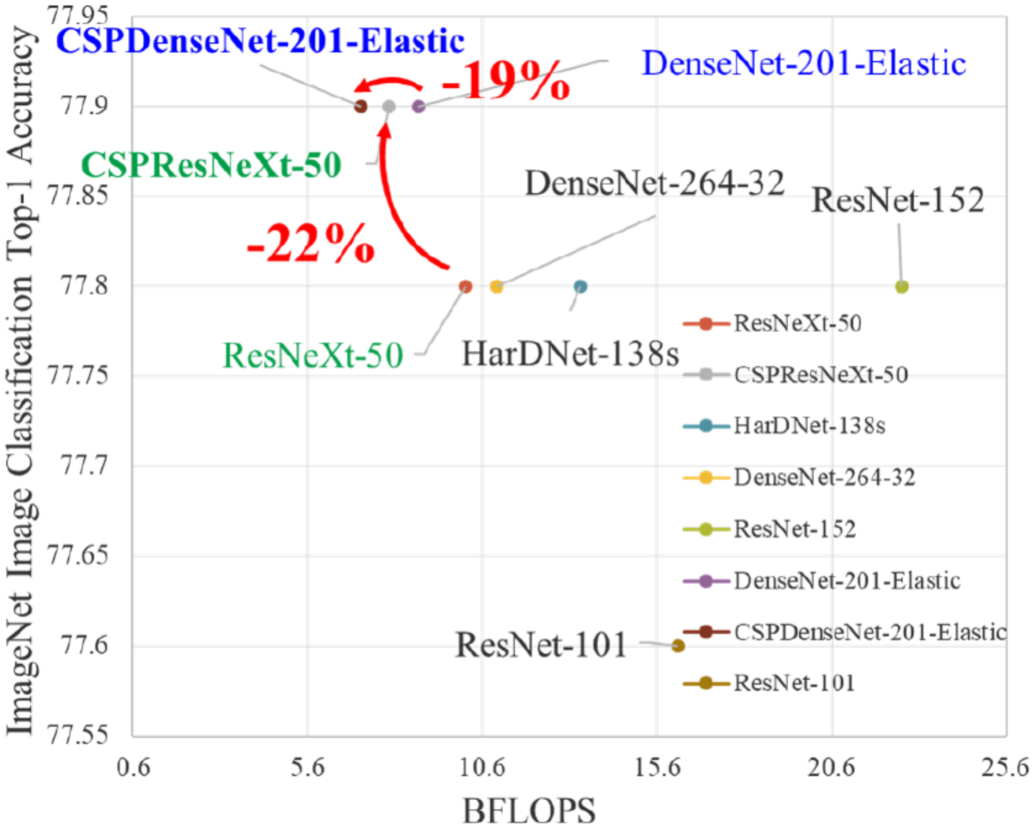


Table 3: Compare with state-of-the-art methods on ImageNet.

Model	#Parameter	BFLOPs	Top-1	Top-5
ResNet-10 [7]	5.24M	2.273	63.5%	85.0%
CSPResNet-10	2.73M	1.905 **(-16%)**	**65.3%**	**86.5%**
ResNeXt-50 [39]	22.19M	10.11	77.8%	**94.2%**
CSPResNeXt-50	20.50M	7.93 **(-22%)**	**77.9%**	94.0%
HarDNet-138s [1]	35.5M	13.4	77.8%	-
DenseNet-264-32 [11]	27.21M	11.03	77.8%	93.9%
ResNet-152 [7]	60.2M	22.6	77.8%	93.6%
DenseNet-201-Elastic [36]	19.48M	8.77	**77.9%**	**94.0%**
CSPDenseNet-201-Elastic	20.17M	7.13 **(-19%)**	**77.9%**	**94.0%**

Q&A