Typical CNN algorithms

2021.12

Outline

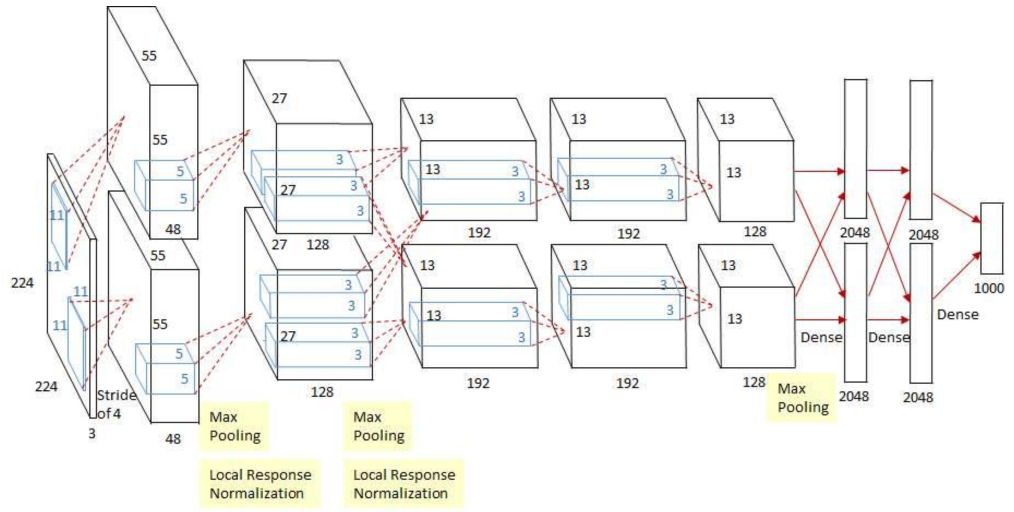
- AlexNet
- VGGNet
- FCN

AlexNet

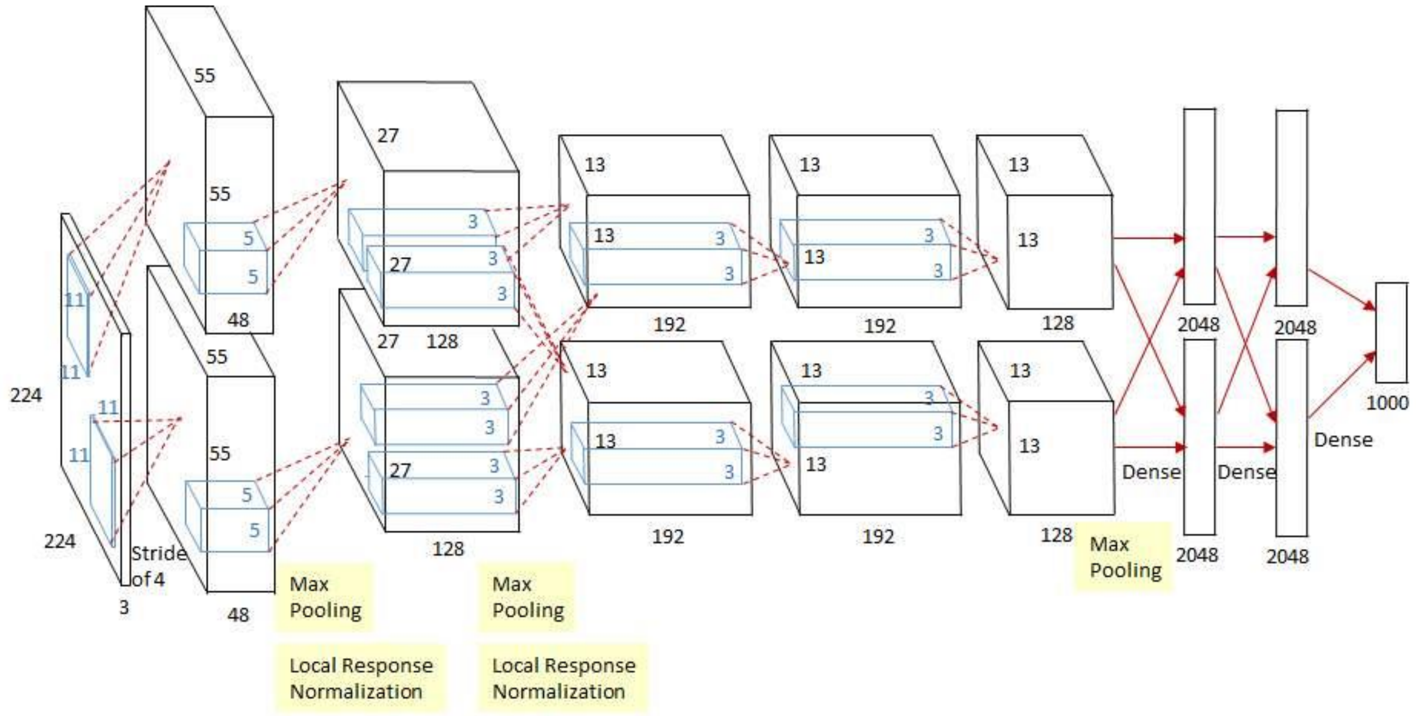
- Alex Krizhevsky, Ilya Sutskever, Geoffrey E.Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, NIPS 2012
- The first time a large scale deep learning model is adopted and effective on large scale computer vision task
- GPU is shown to be very effective on this large deep model, with 2 GPU, 2GB RAM on each GPU, 5GB of system memory
- ImageNet is very important for AlexNet

AlexNet

- 5 convolutional layers and 3 fully connected layers, 3 Max-pooling layers
- 650K neurons, 62M parameters
- Trained on ImageNet - one million images of 1000 categories
- With 2 GPU, training lasts for one week



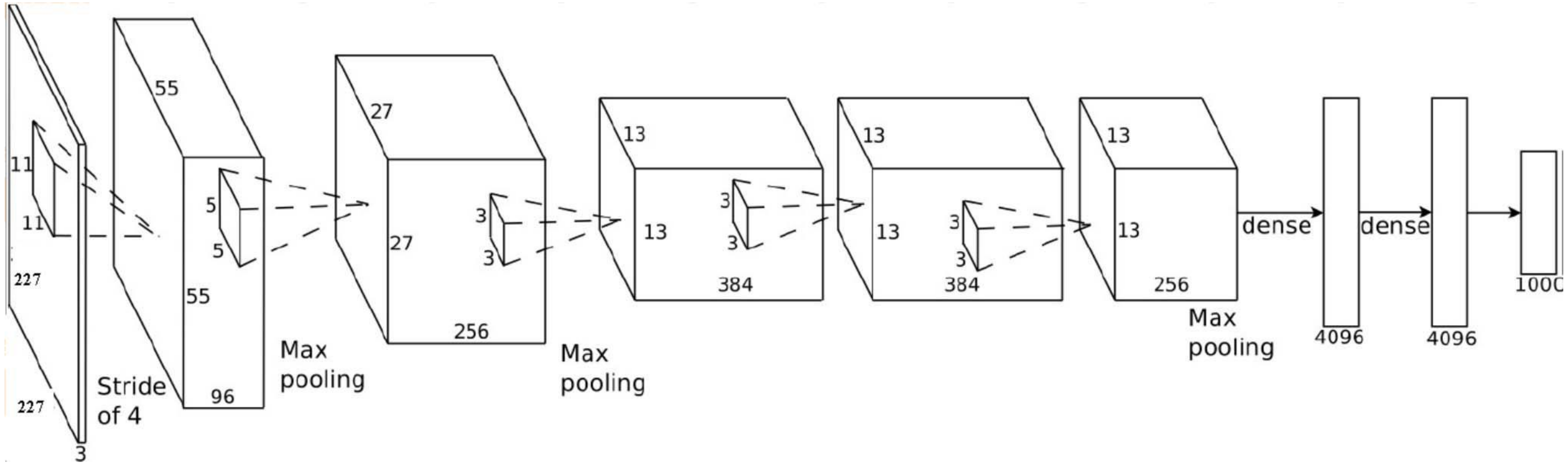
AlexNet



224
224
11
11
11
11
Stride of 4
3
55
55
5
5
48
55
55
5
5
48
Max Pooling
Local Response Normalization
27
27
3
3
3
27
128
3
3
3
27
128
Max Pooling
Local Response Normalization
13
13
3
3
192
13
13
3
3
192
13
13
3
3
192
13
13
3
3
192
13
13
128
13
13
128
Max Pooling
Dense
Dense
Dense
2048
2048
2048
2048
1000

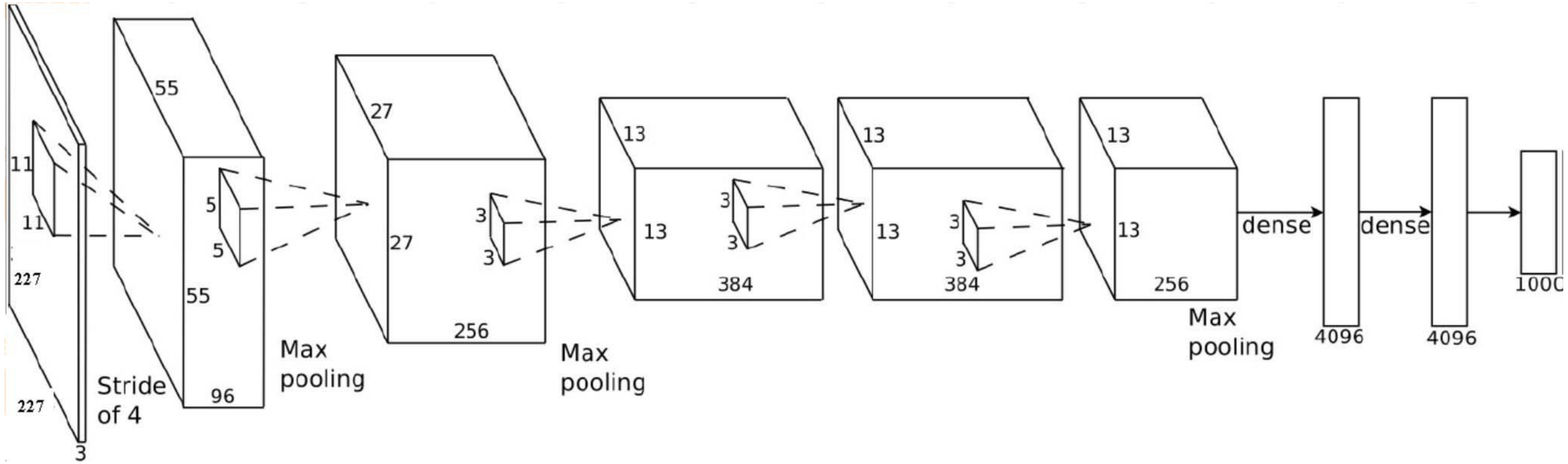
AlexNet

- 8 weight layers, 5 convolutional layers and 3 fully connected layers for learning features
- 3 Max-pooling layers follow first, second, and fifth convolutional layers
- The number of neurons in each layer is given by 253440, 186624, 64896, 64896, 43264, 4096, 4096, 1000



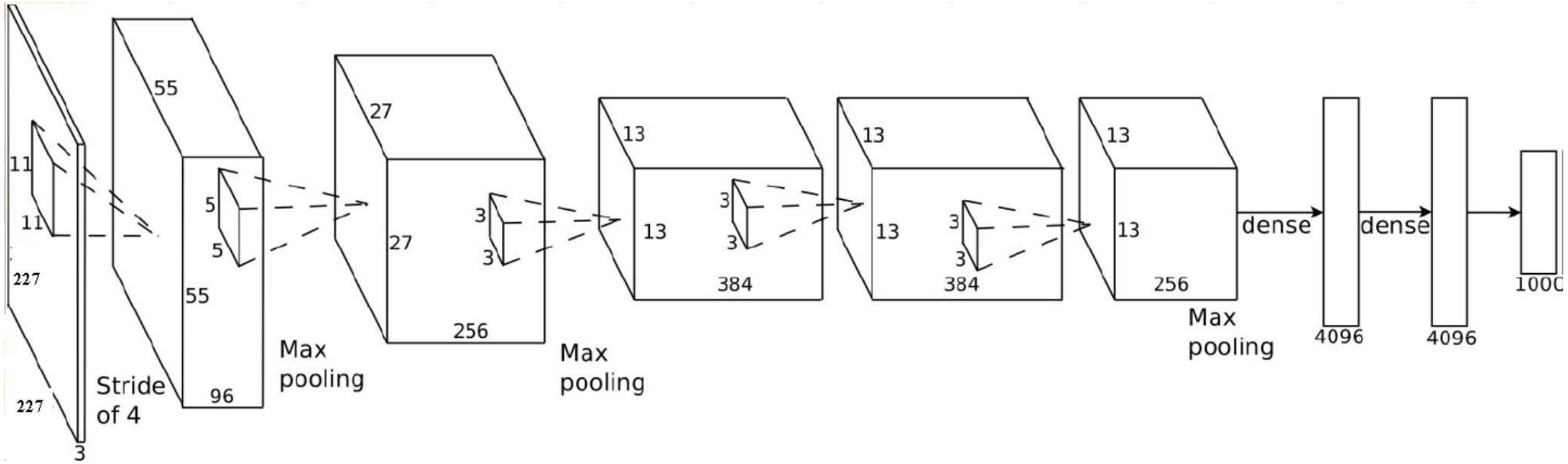
特征图的尺寸

- 对于Max pooling, 2x2个神经元中取一个数值最大的特征值，特征图的尺寸是[(M/2)x(M/2)]
- 例如, 55/2取整后是27, 27/2取整后是13, 13/2取整后是6, 27x27, 13x13, 6x6
- 对于卷积，第一个卷积层的输出特征图55x55
- 55=[(N+2p-k)/s]+1 = [(227+2-11)/4]+1 = 54+1 = 55, 向下取整
- N, 输入尺寸, k, 滤波器尺寸, s, 步长, p, padding



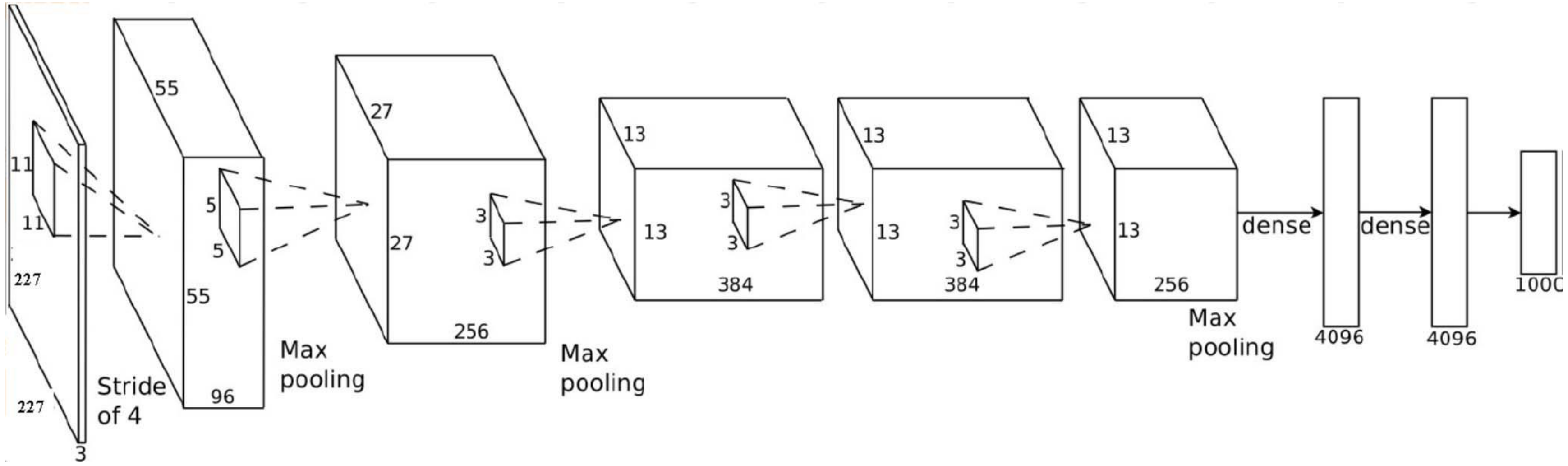
AlexNet 的参量

- 一个神经元产生一个特征值y
- 同一空间位置的一组神经元产生一个特征矢量y
- 一组神经元的数量m，即特征矢量y的维度，即输出特征图的通道数量（channel），96, 256...
- 所有神经元组的数量，即特征图的大小，也表达了与图像的空间对应位置关系
- 例如：55x55, 27x27, 13x13表示神经元组的数量



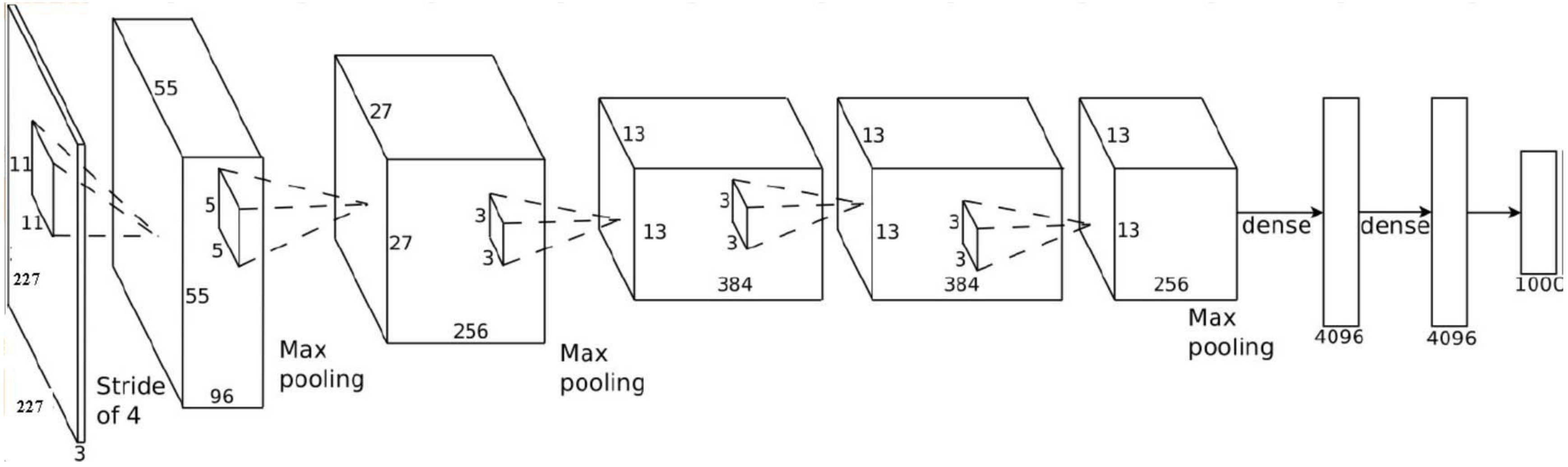
AlexNet的神经元的数量

- 神经元组的数量与一组内神经元的数量相乘，即是一层中所有神经元的数量
- 例如，第5个卷积层，13x13即神经元组的数量，256是一组内包含的神经元数量，因此，13x13x256=43264是第5个卷积层的神经元的总数量
- 类似的，第2个卷积层的神经元的总数量是 27x27x256=186624



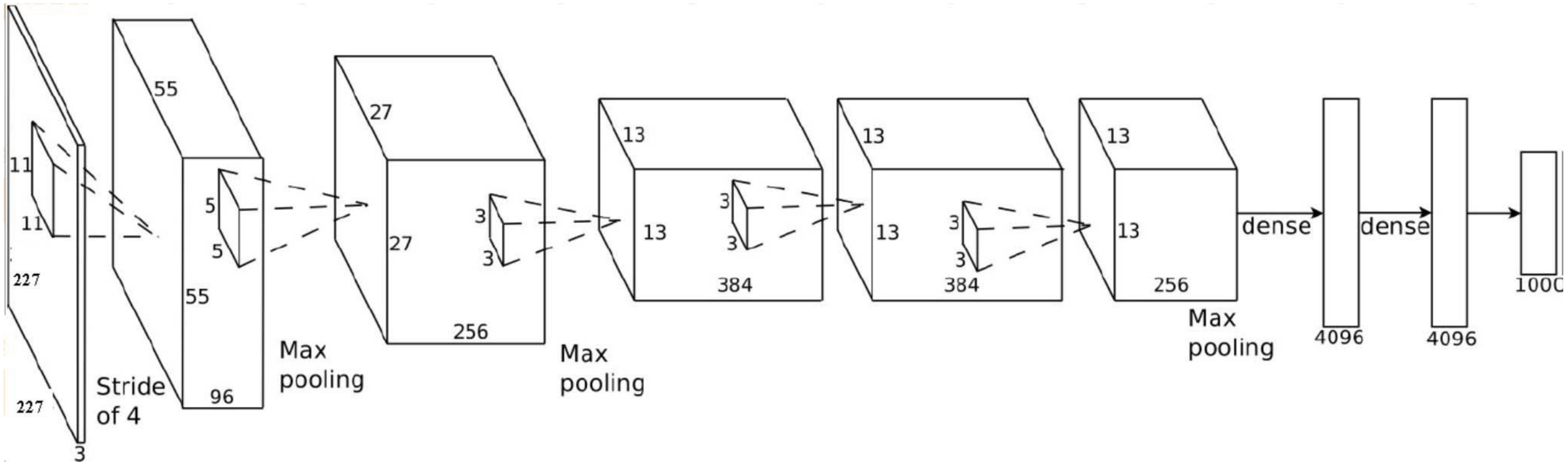
AlexNet的参数变量的数量

- 一个神经元包含的参数量，是指其输入连接的数量，包括局部感受野区域内所有通道的数据连接
- 例如，第5个卷积层的神经元的局部感受野是3x3，还要贯穿第4个卷积层的所有384个特征图，所以3x3x384=3456是第5层一个神经元的参数量，256个神经元一组，各组参数相同，第5层的总参数变量的数量是3456x256=0.9M



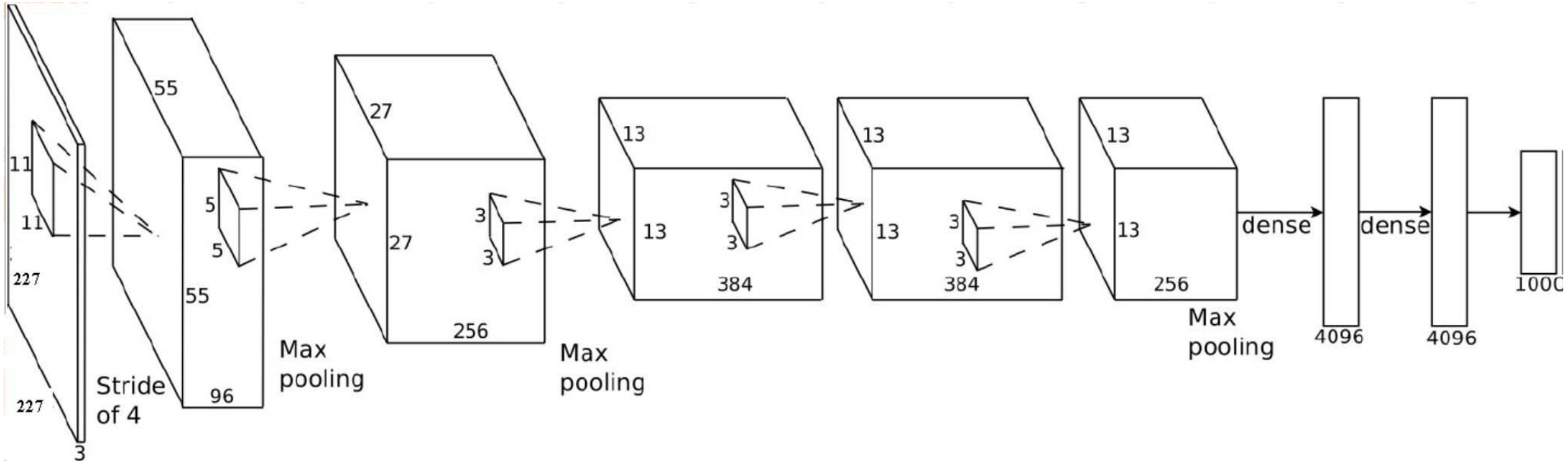
卷积层的参数量

- 类似的，第4个卷积层的参数量是3x3x384x384=1.3M
- 第3个卷积层的参数量是3x3x256x384=0.9M
- 第2个卷积层的参数量是5x5x96x256=0.6M
- 第1个卷积层的参数量是11x11x3x96=0.03M



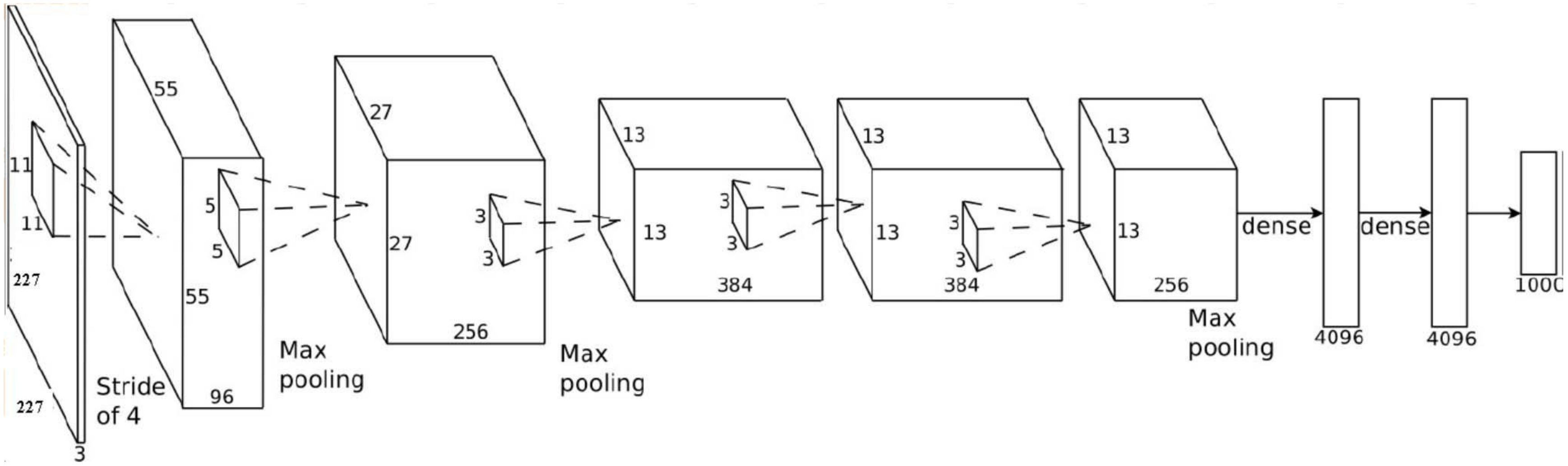
全连接层的参数量

- 卷积层后续3个全连接层
- 第1个全连接层，4096个神经元，每个神经元需要6x6x256=9216个连接参数，共有约9216x4096≈37.7M个参数
- Max pooling将13x13的特征图减小到6x6
- 第2个全连接层，4096个神经元，每个神经元需要4096个连接参数，共约4096x4096≈16.8M



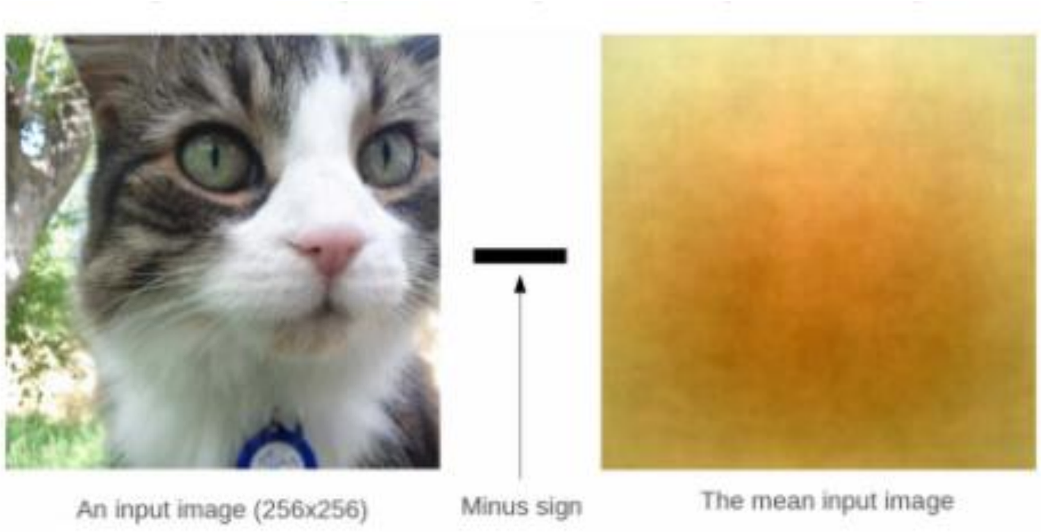
AlexNet的总参数量

- 第3个全连接层，也是最后的分类器层，分类1000类，使用1000个神经元，每个神经元需要4096个连接参数，共约1000x4096≈4.1M
- 3个全连接层共有约58.6M个参数
- AlexNet共有约62.3M个参数

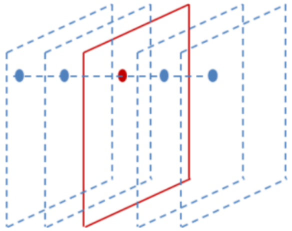


Normalization

- Normalize the input by subtracting the mean image on the training set

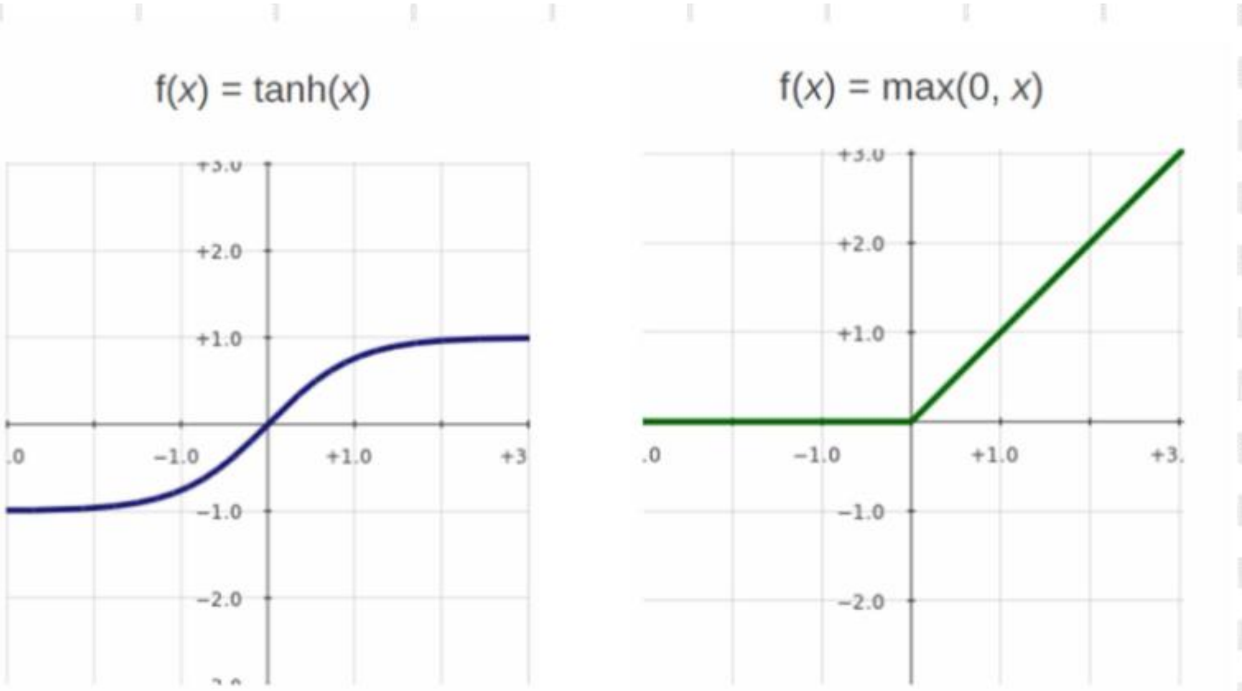


- LRN?



ReLU

- Choice of activation function



Very bad (slow to train)

Very good (quick to train)

dropout

- Independently set each hidden unit activity to zero with 0.5 probability
- Do this in the two globally-connected hidden layers

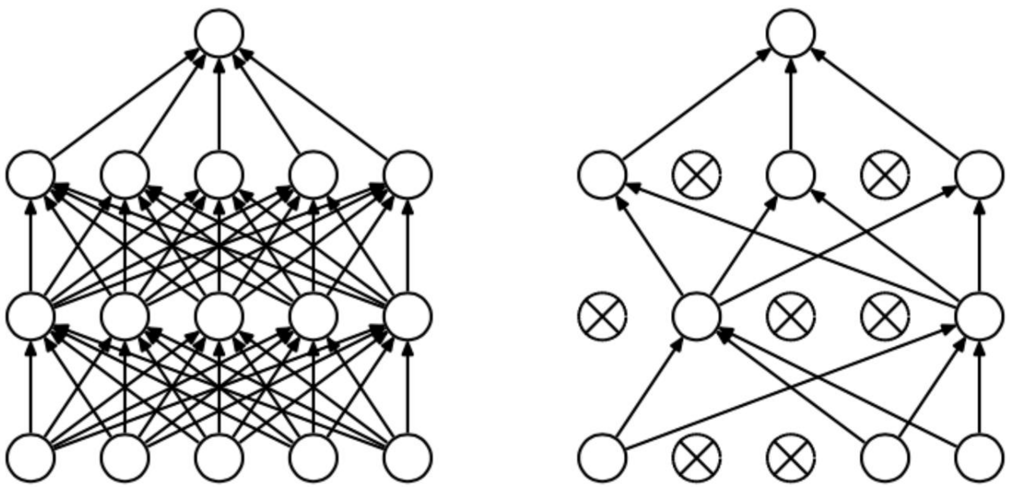
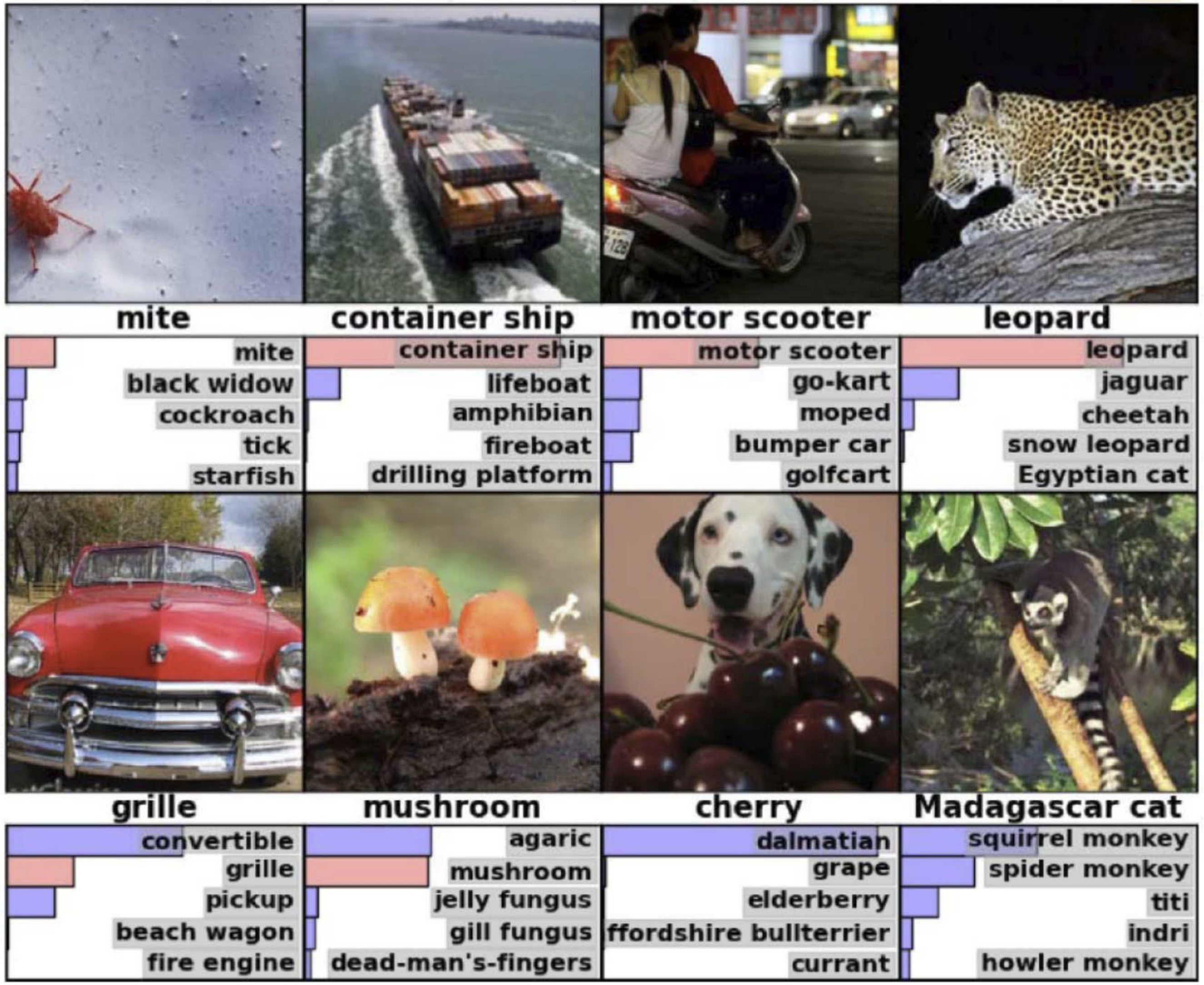
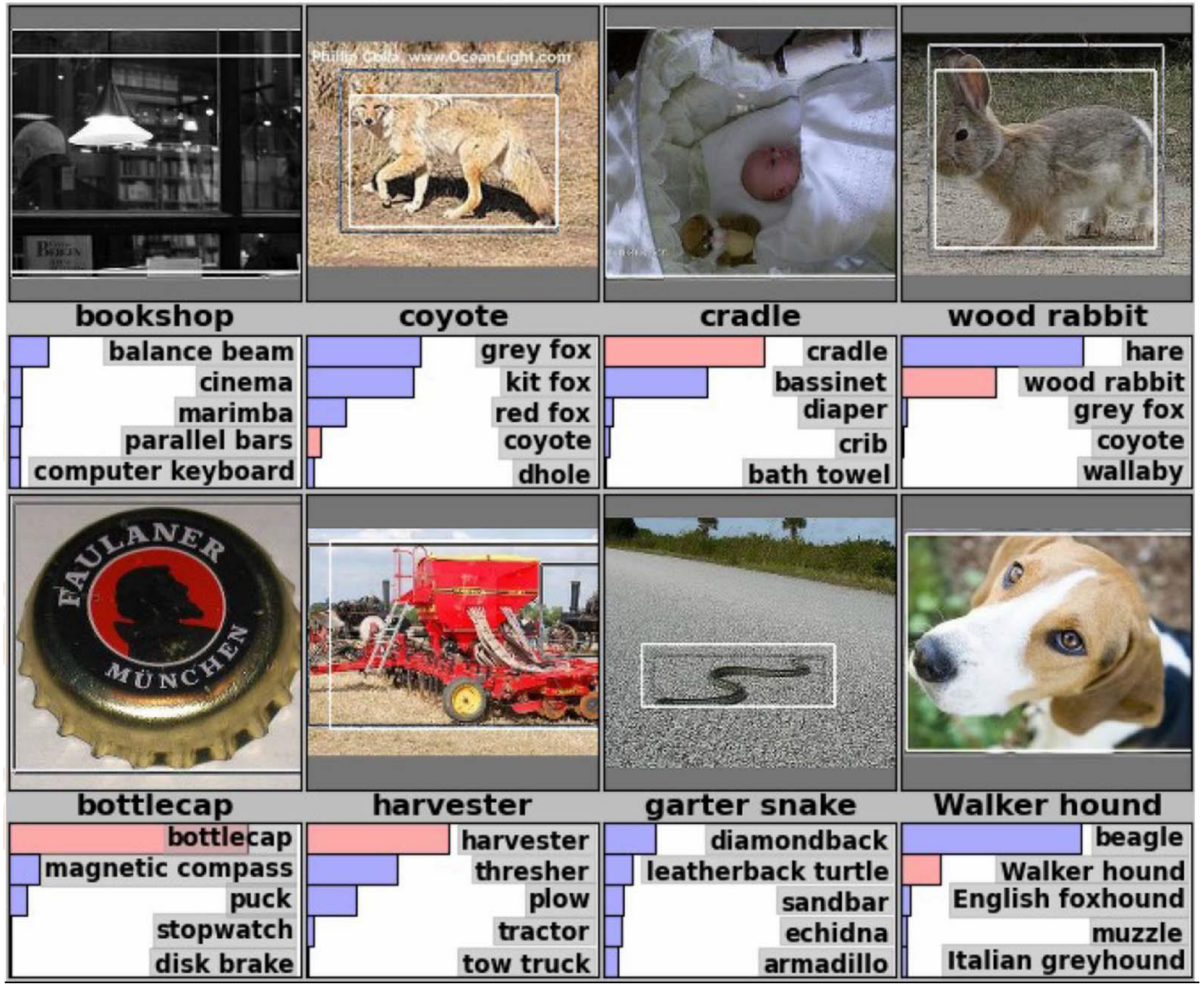


Image classification result



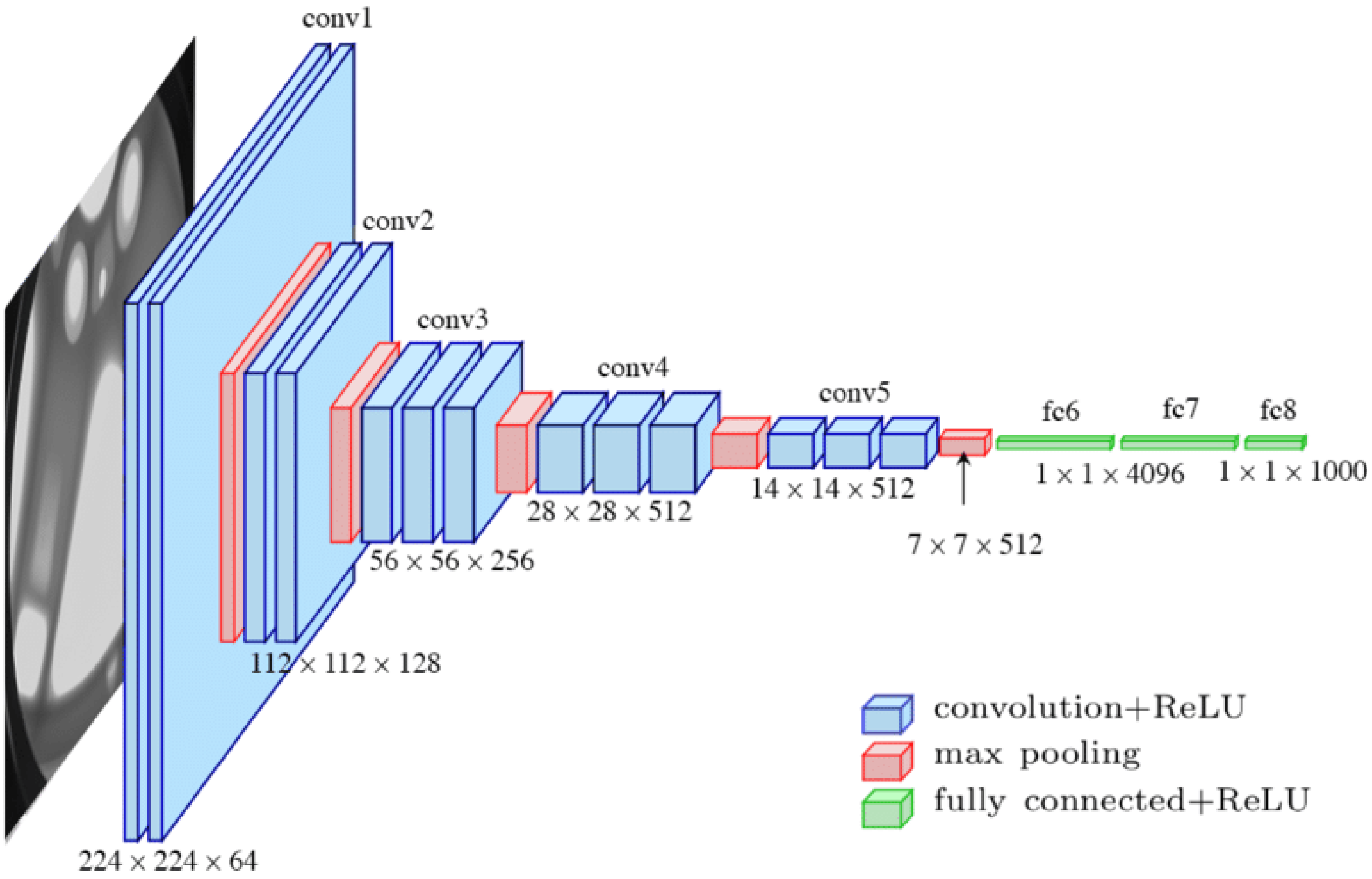
Detection result



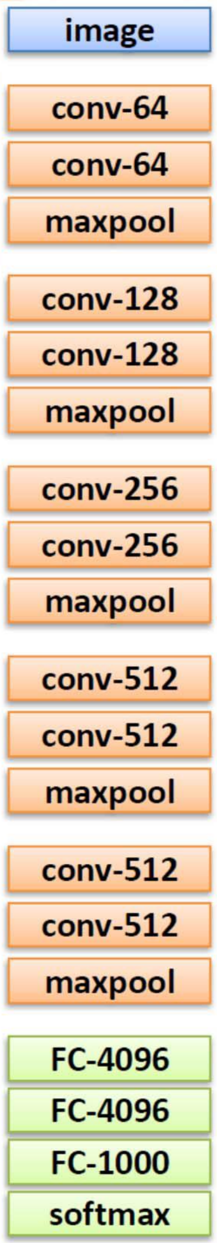
VGG 16

- Karen Simonyan & Andrew Zisserman, Oxford, ICLR15, “VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION”
- Localisation task: 1st place, 25.3% error
- Classification task: 2nd place, 7.3%
- Depth up to 19 layers

VGG 16


conv1
conv2
conv3
conv4
conv5
fc6
fc7
fc8
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
1 × 1 × 4096
1 × 1 × 1000
convolution+ReLU
max pooling
fully connected+ReLU

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

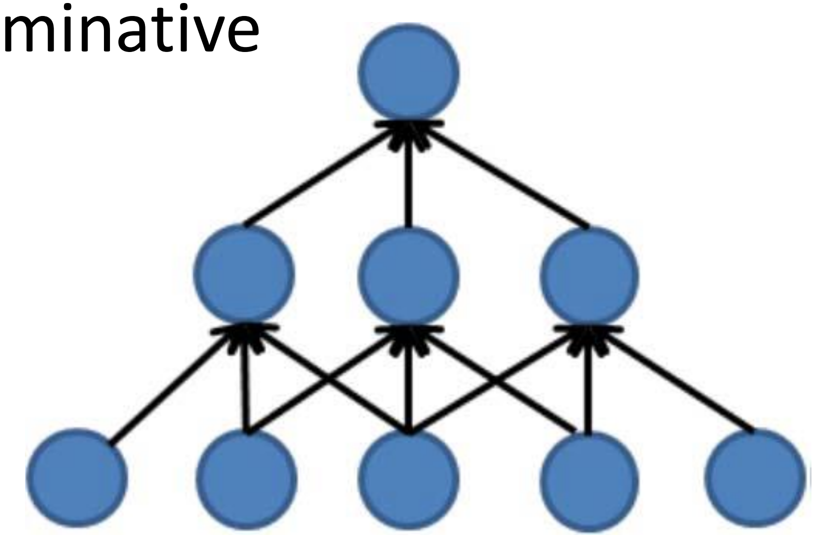


Key design choices

- Apply 3 x 3 filter for all layers
- 3 x 3 filter: the smallest size to capture the notion of left/right, up/down, center
- Rather than using relatively large receptive fields in AlexNet (11 x 11 & 5 x 5)
- Conv. stride 1 – no loss of information
- 3 fully-connected (FC) layers
- 5 max-pool layers (x2 reduction)
- No normalisation

Convolution decomposition

- Stacked conv. layers have a large receptive field
  - two 3x3 layers – 5x5 receptive field
  - three 3x3 layers – 7x7 receptive field
- More non-linearity: incorporate three non-linear rectification layers instead of a single one
- Makes the decision function more discriminative
- Less parameters to learn



Advantages of VGGnet

- Decreases the number of parameters
  - Three 3 x 3 filters and one 7 x 7 filter
  - Weight numbers: 3 x 3 x 3 = 27, 7 x 7 = 49
- The incorporation of 1 x 1 filters also increases the nonlinearity of the decision function without affecting the receptive fields of the conv. layers
  - 1 x 1 convolution filters: only for one feature vector
- Normal architecture, architectural simplicity
- Better to have deeper layers, smaller receptive window size

Results

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
C	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D	256	256	27.0	8.8
	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
E	256	256	27.3	9.0
	384	384	26.9	8.7
	[256;512]	384	**25.5**	**8.0**

Conclusion

- VGG proposed a standard CNN structure with 3x3 filters to improve nonlinearity and depth
- Presented convolution decomposition method

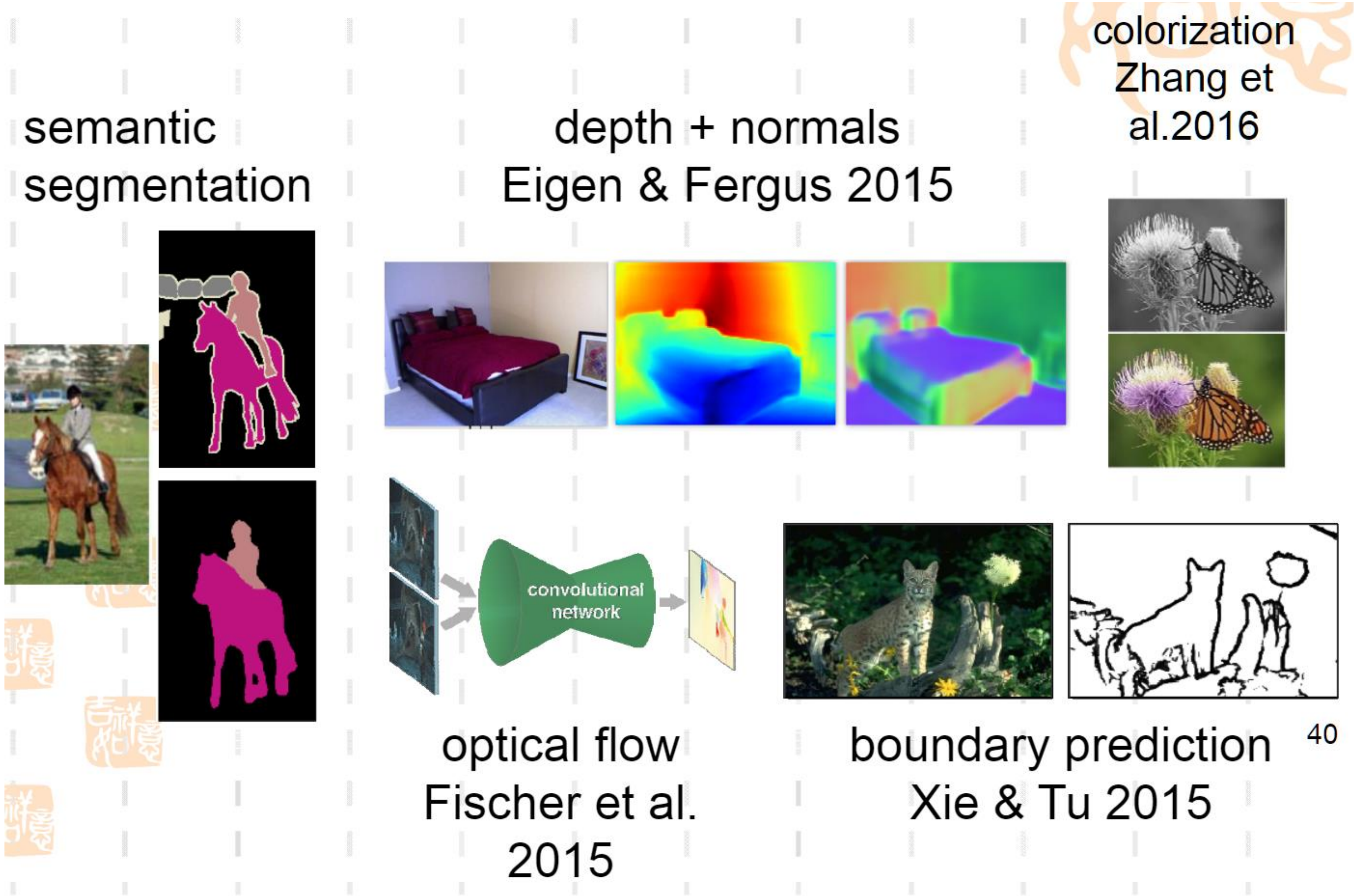
Conclusion

- VGG proposed a standard CNN structure with 3x3 filters to improve nonlinearity and depth
- Presented convolution decomposition method

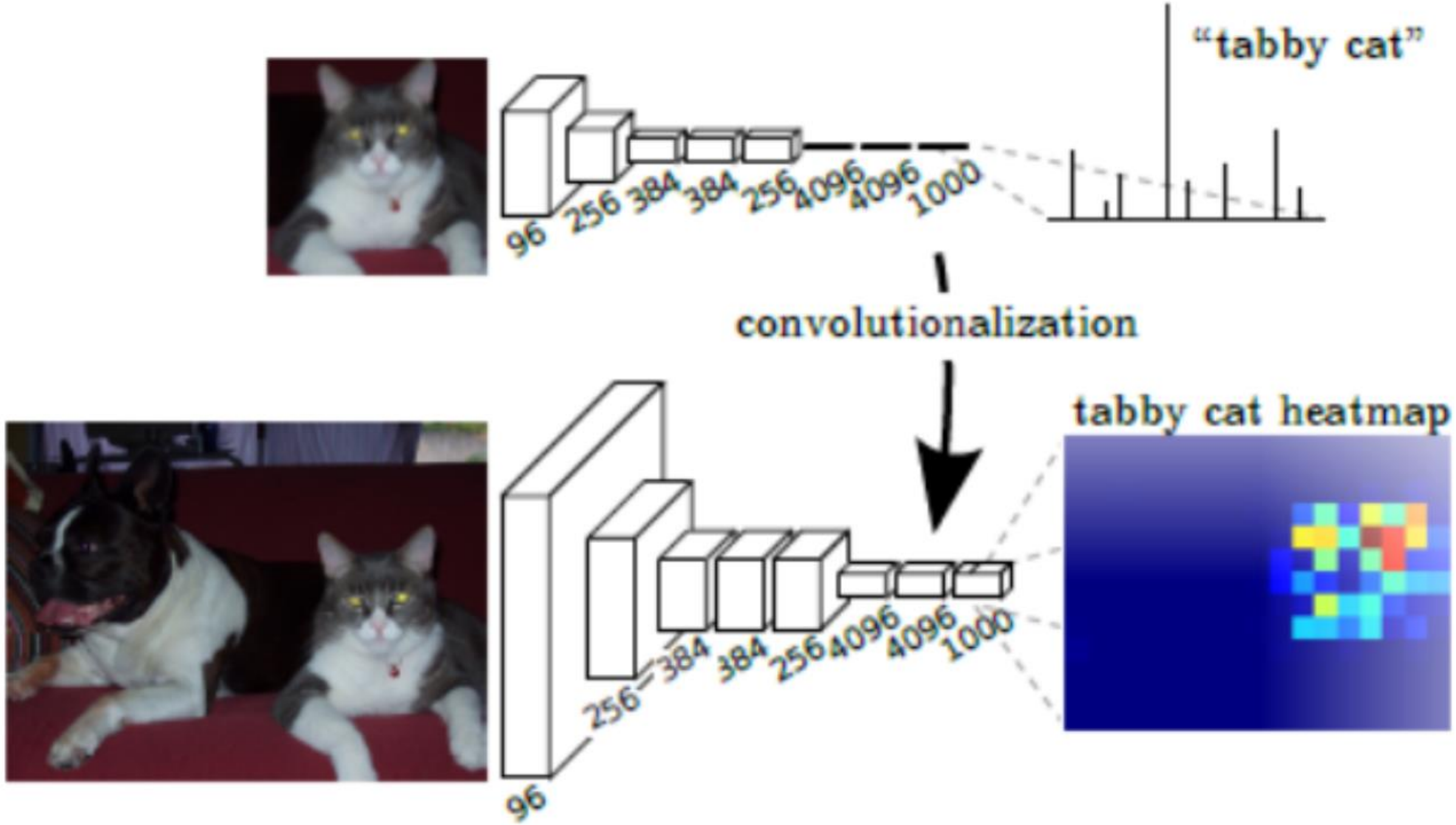
Fully Convolutional Network(FCN)

- Jonathan Long, Evan Shelhamer, Trevor Darrell, UC Berkeley
- Jonathan Long, Evan Shelhamer, Trevor Darrell, UC Berkeley
- CVPR 2016 best paper honorable mention

Pixel in,Pixel out



FCN



"tabby cat"
96
256
384
384
256
4096
4096
1000
convolutionalization
tabby cat heatmap
96
256
384
384
256
4096
4096
1000

Semantic Segmentation

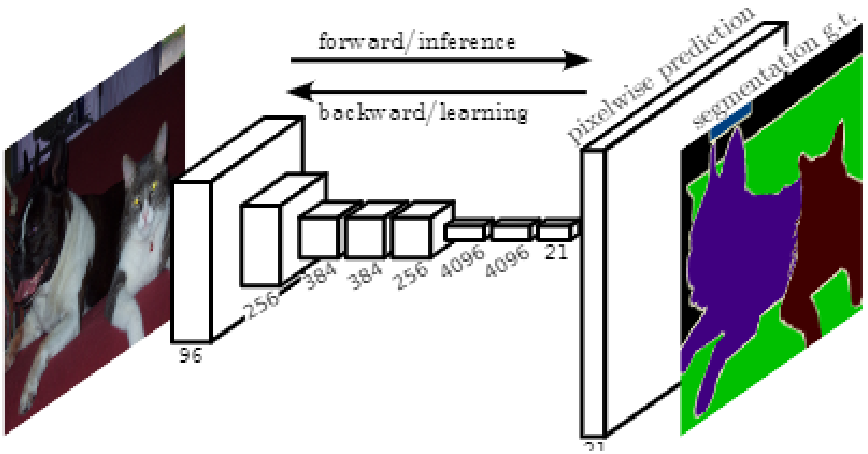
- Arbitrary-sized inputs
- Single label output per pixel
- Dense prediction on 2D feature map

Becoming fully convolutional

- Fix fc layers as convnets
- Final fc output becomes 2D feature map
- Size of 2D feature map varies with input
- Single label output per final pixel

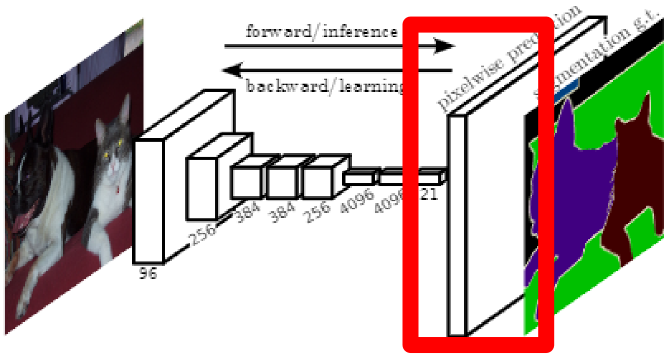
Becoming fully convolutional

- Fix fc layers as convnets
- Final fc output becomes 2D feature map
- Size of 2D feature map varies with input
- Single label output per final pixel

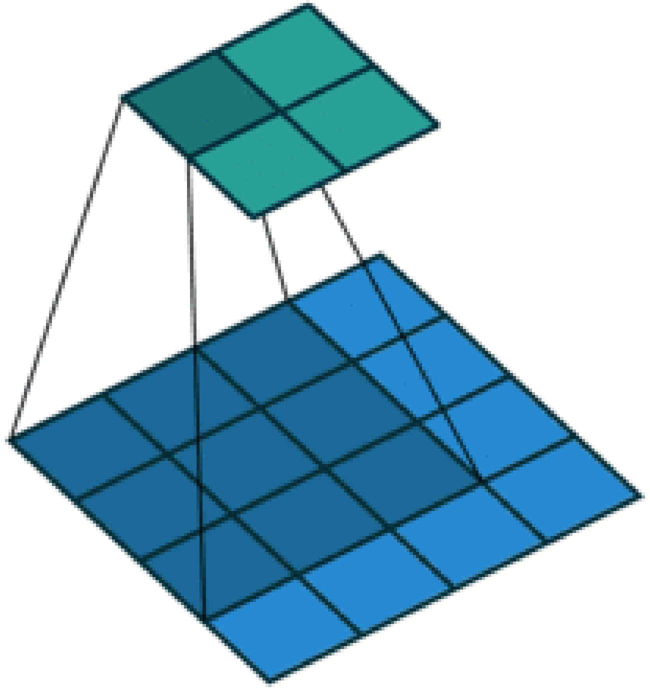


Unsampling output

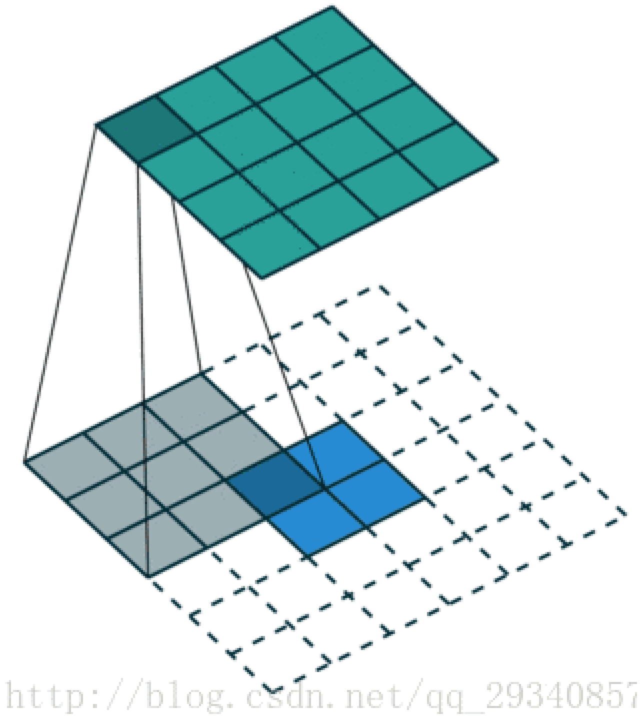
- Upsampling with factor f is convolution with a fractional input stride of 1= 1/f , deconvolution
- Performed in-network for end-to-end learning by backpropagation from pixelwise loss
- The deconvolution filters are learned



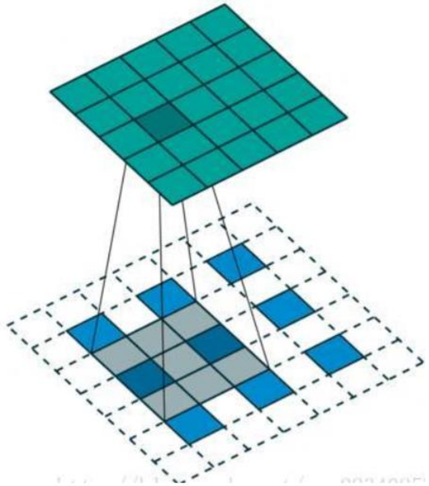
deconvolution



卷积



反卷积



Skip layers

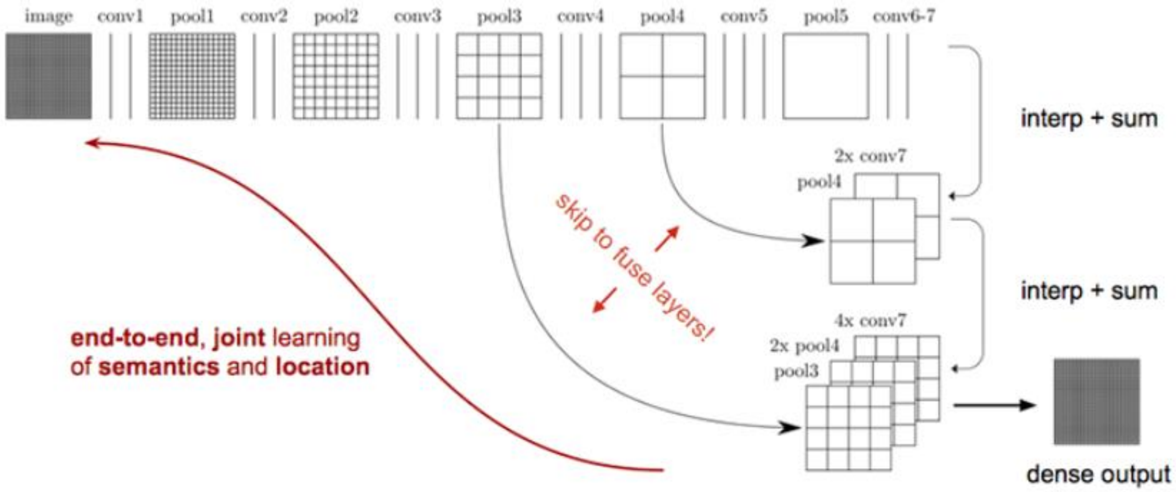
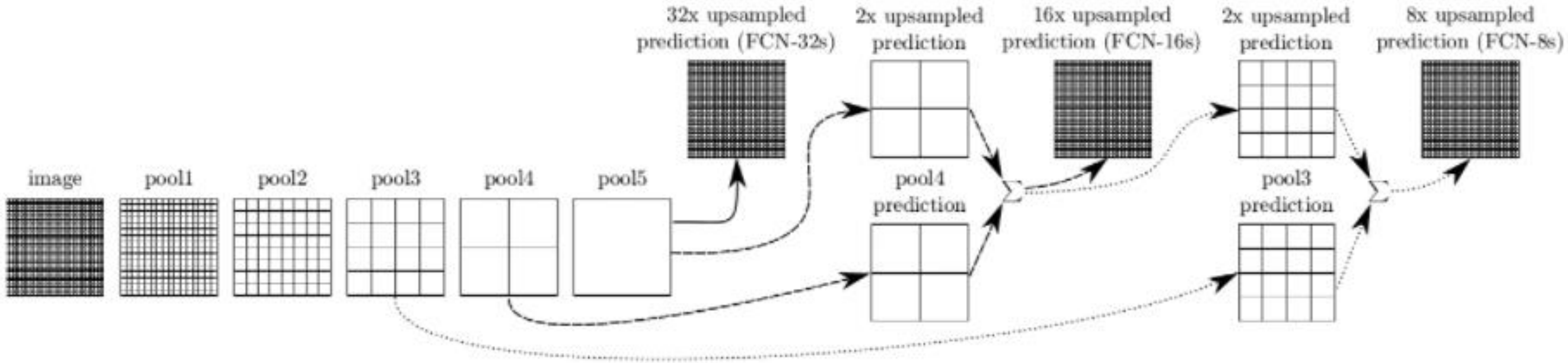
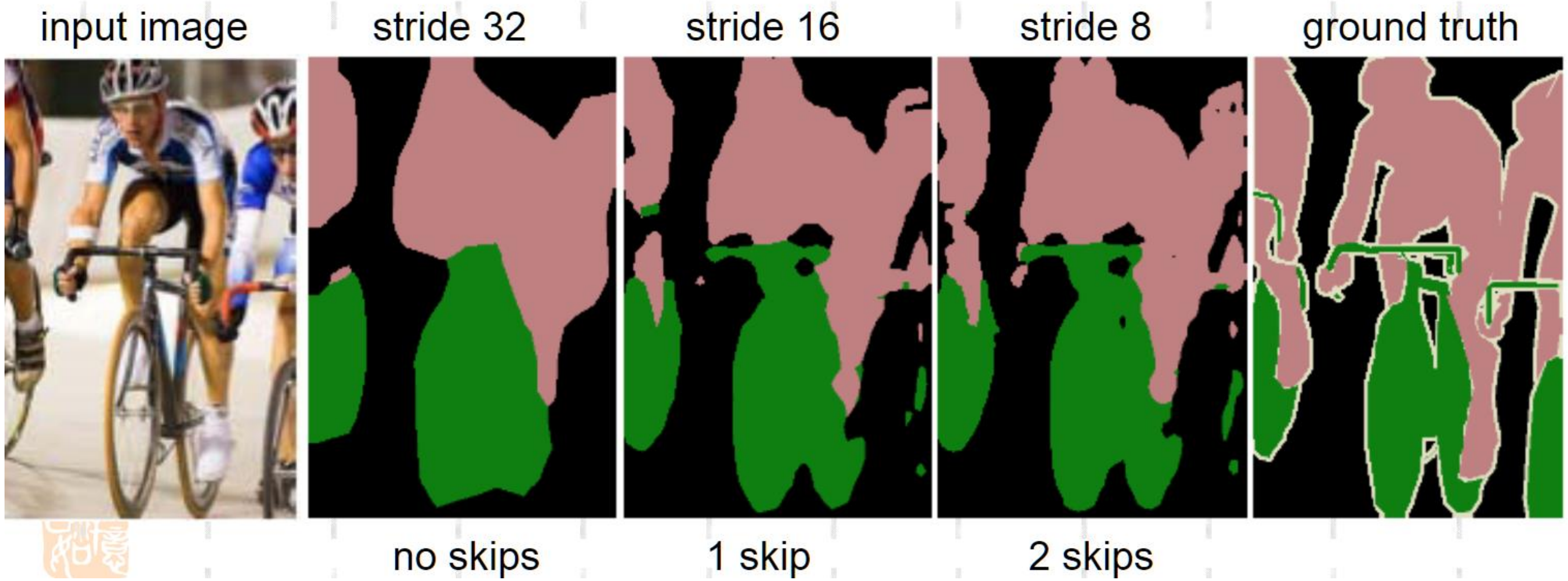


image
conv1
pool1
conv2
pool2
conv3
pool3
conv4
pool4
conv5
pool5
conv6-7
interp + sum
2x conv7
pool4
skip to fuse layers!
interp + sum
4x conv7
2x pool4
pool3
end-to-end, joint learning
of semantics and location
dense output



32x upsampled
prediction (FCN-32s)
2x upsampled
prediction
16x upsampled
prediction (FCN-16s)
2x upsampled
prediction
8x upsampled
prediction (FCN-8s)
image
pool1
pool2
pool3
pool4
pool5
pool4
prediction
pool3
prediction

Skip layer refinement



Skip layer refinement