Convolutional Neural Networks (卷积神经网络)

历史

Gradient-Based Learning Applied to Document Recognition

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

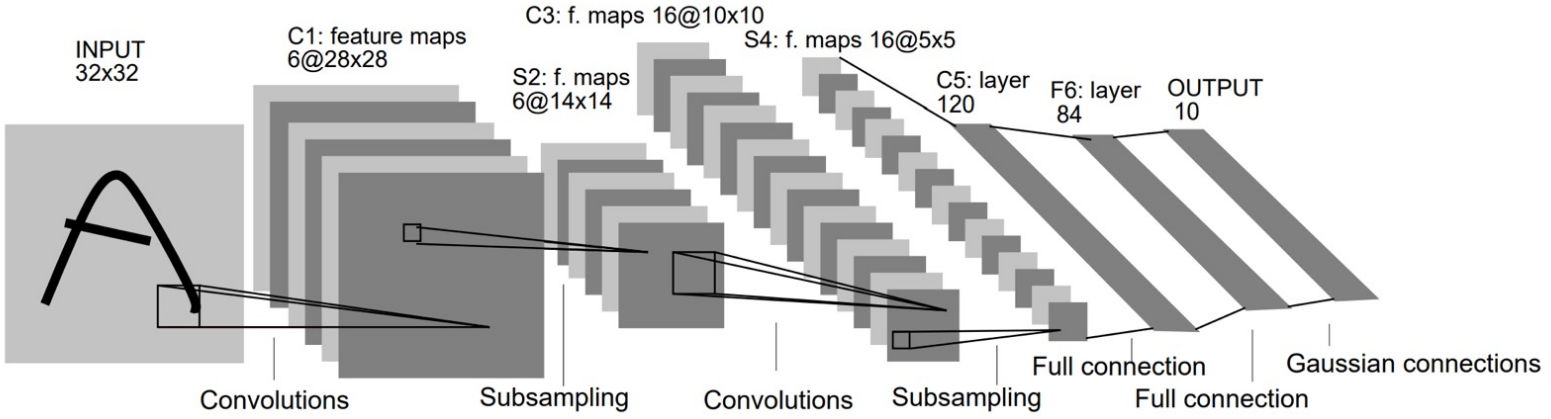
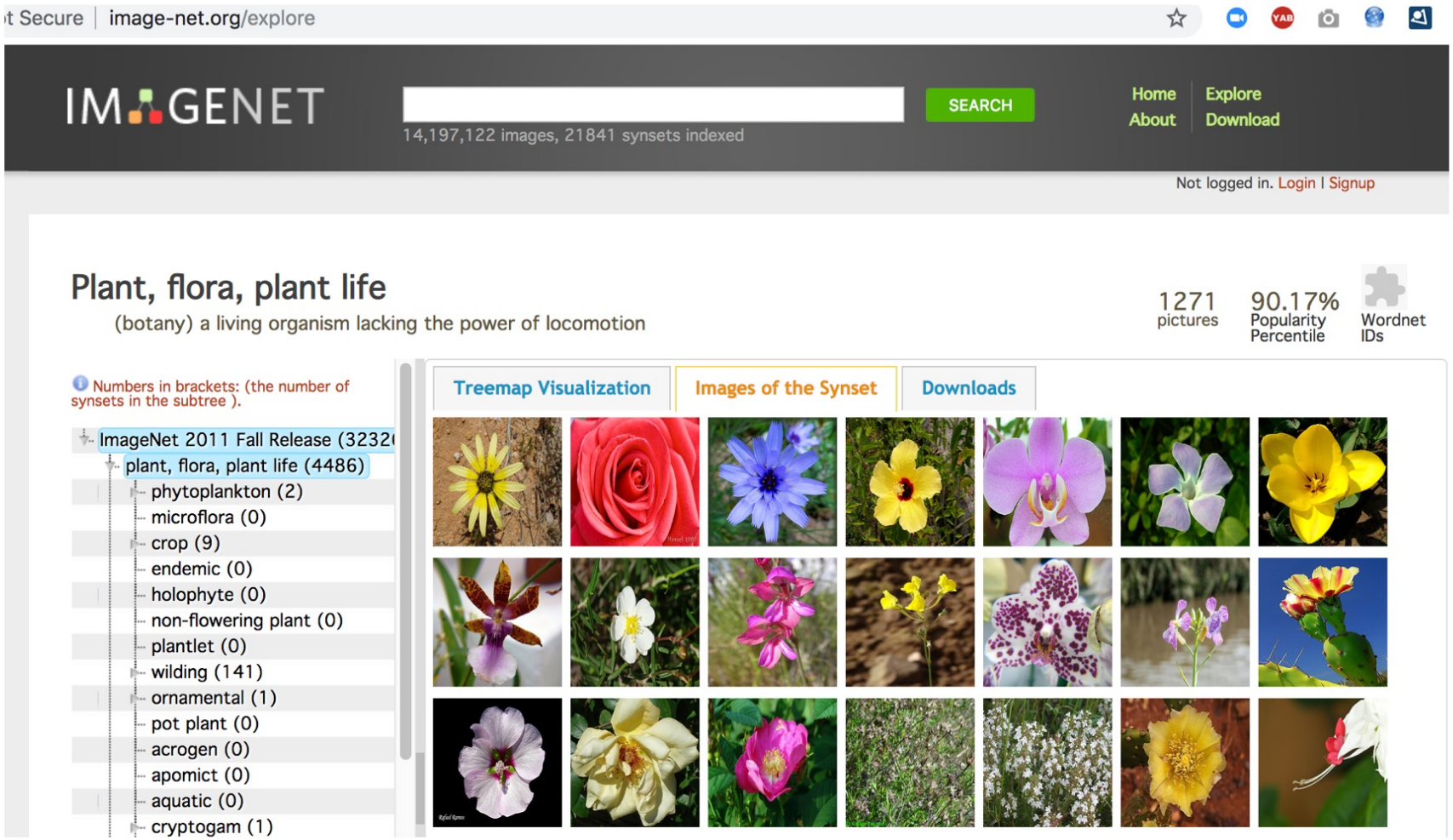


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

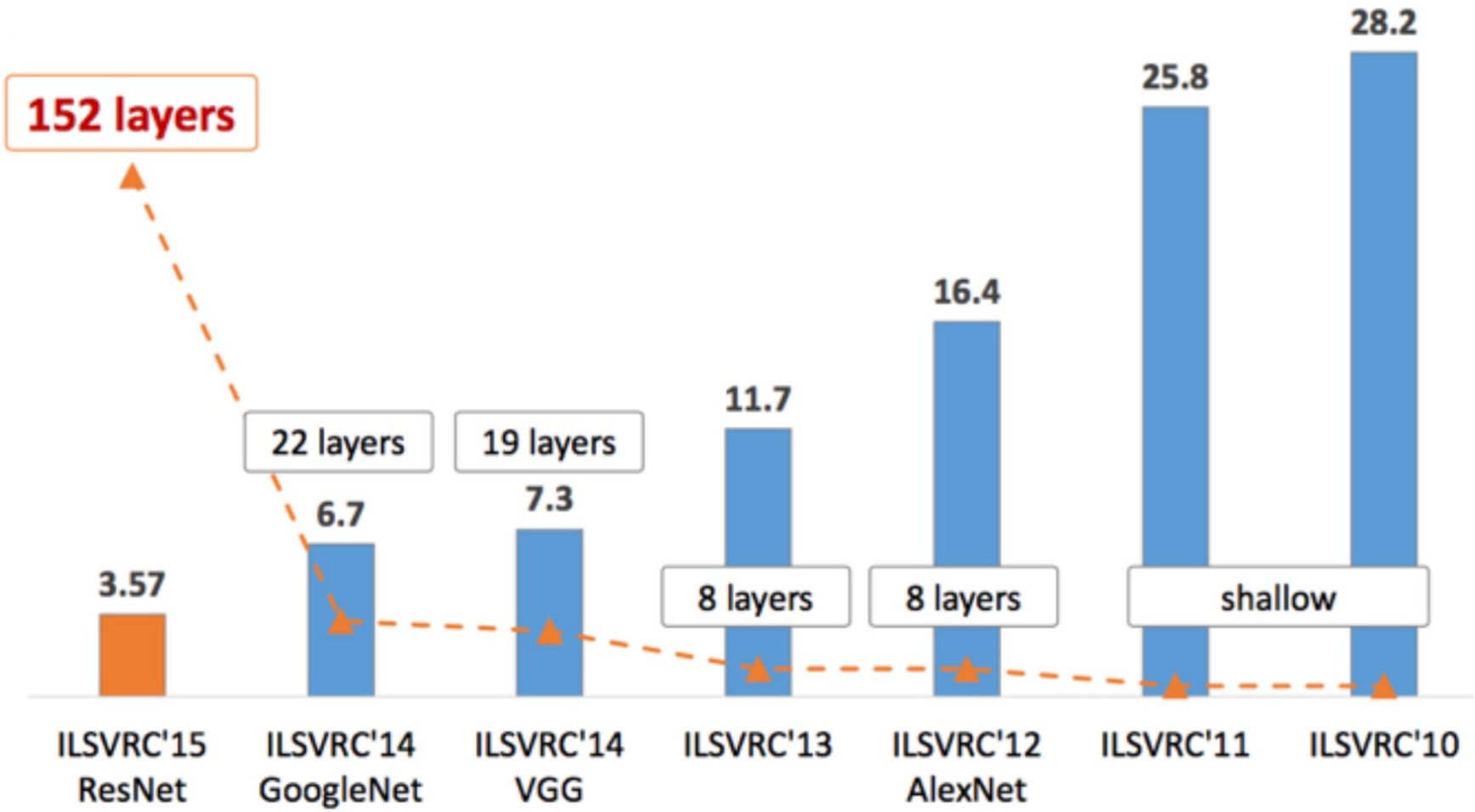
典型的网络结构

ImageNet 比赛

- 从2010年开始, 每年一次
- 1百4十万张图片
- 1000个类别的分类问题:
  https://gist.github.com/yrevar/942d3a0ac09ec9e5eb3a


t Secure | image-net.org/explore
IMAGENET
SEARCH
14,197,122 images, 21841 synsets indexed
Home
Explore
About
Download
Not logged in. Login I Signup
Plant, flora, plant life
(botany) a living organism lacking the power of locomotion
1271
pictures
90.17%
Popularity
Percentile
Wordnet
IDs
Numbers in brackets: (the number of synsets in the subtree).
ImageNet 2011 Fall Release (3232
plant, flora, plant life (4486)
phytoplankton (2)
microflora (0)
crop (9)
endemic (0)
holophyte (0)
non-flowering plant (0)
plantlet (0)
wilding (141)
ornamental (1)
pot plant (0)
acrogen (0)
apomict (0)
aquatic (0)
cryptogam (1)
Treemap Visualization
Images of the Synset
Downloads

Year	CNN	Developed by	Place	Top-5 error rate	No. of parameters
1998	LeNet(8)	Yann LeCun et al			60 thousand
2012	AlexNet(7)	Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever	1st	15.3%	60 million
2013	ZFNet()	Matthew Zeiler and Rob Fergus	1st	14.8%	
2014	GoogLeNet(19)	Google	1st	6.67%	4 million
2014	VGG Net(16)	Simonyan, Zisserman	2nd	7.3%	138 million
2015	ResNet(152)	Kaiming He	1st	3.6%	


152 layers
22 layers
19 layers
8 layers
8 layers
shallow
3.57
6.7
7.3
11.7
16.4
25.8
28.2
ILSVRC'15 ResNet
ILSVRC'14 GoogleNet
ILSVRC'14 VGG
ILSVRC'13
ILSVRC'12 AlexNet
ILSVRC'11
ILSVRC'10

LeNet-5 (1998)

- LeCun 等人发表, 用于银行支票手写数字识别

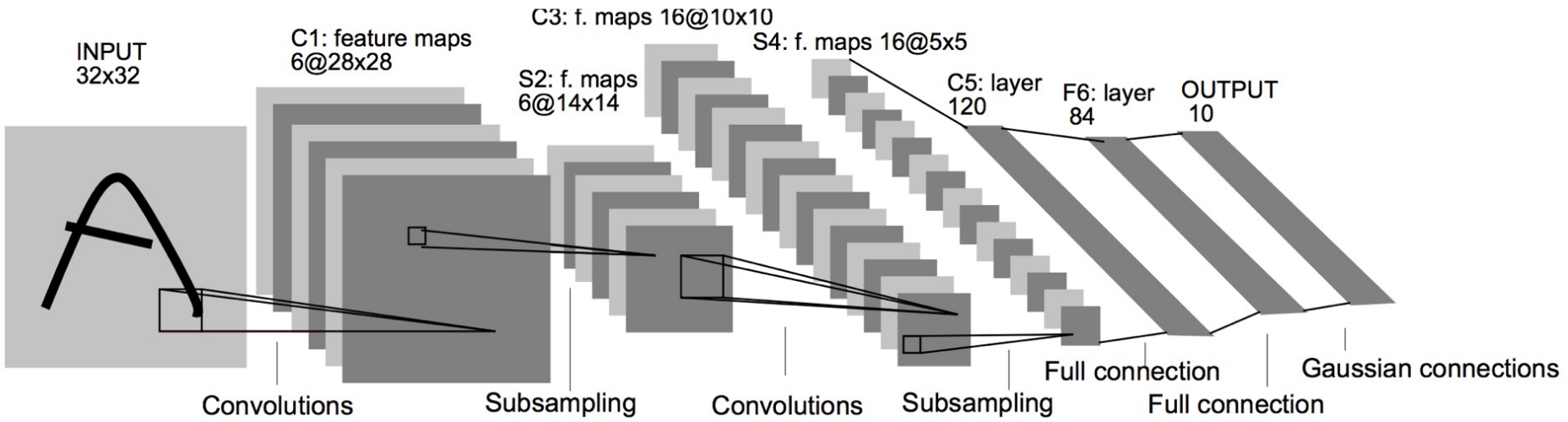


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

AlexNet(2012)

- 2012年ImageNet比赛第一名. Top-5错误率从往届的最好成绩26%降低到15.3%. 同年第二名, 使用非CNN的方法, Top-5的错误率为26.2%
- 类似LeNet的结构, 但是更多层, 每层有更多的滤波器. 使用了11*11, 5*5, 3*3的滤波器尺寸, 使用了Max Pool, Dropout, 数据增强, Relu为激活函数. 使用了带动量(momentum)的SGD.
- 使用了两块Nvidia Geforce GTX 580显卡训练了6天
- 因为使用了两块显卡, 所以网络结构也有两个分支
- Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever 发表

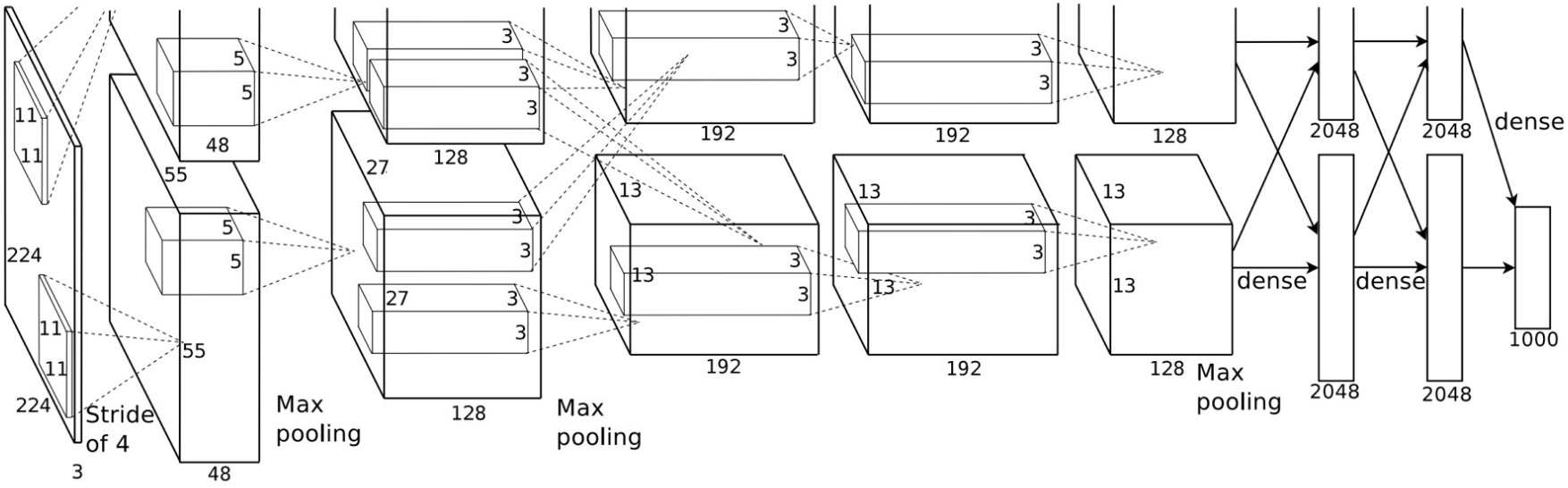


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

ZFNet(2013)

- 采用了AlexNet的机构, 有两个小修改:
- 1. 第一层的滤波器尺寸从11*11减小为7*7
- 2. 第一层的stride(步长)从4降为2

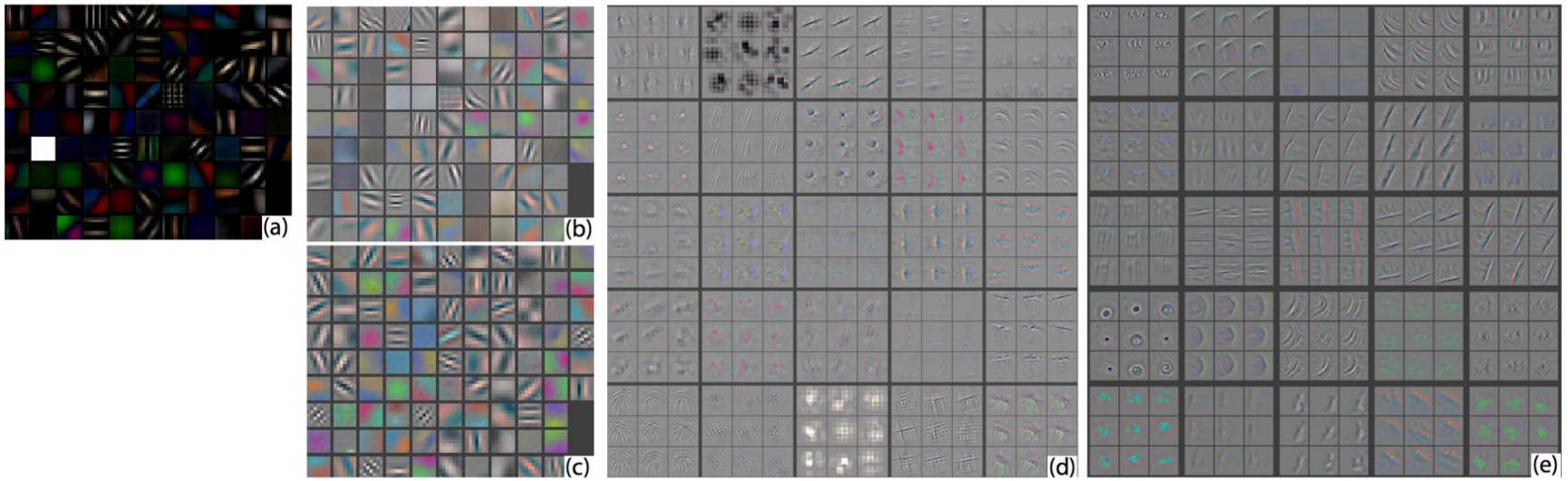


Figure 6. (a): 1st layer features without feature scale clipping. Note that one feature dominates. (b): 1st layer features from (Krizhevsky et al., 2012). (c): Our 1st layer features. The smaller stride (2 vs 4) and filter size (7x7 vs 11x11) results in more distinctive features and fewer "dead" features. (d): Visualizations of 2nd layer features from (Krizhevsky et al., 2012). (e): Visualizations of our 2nd layer features. These are cleaner, with no aliasing artifacts that are visible in (d).

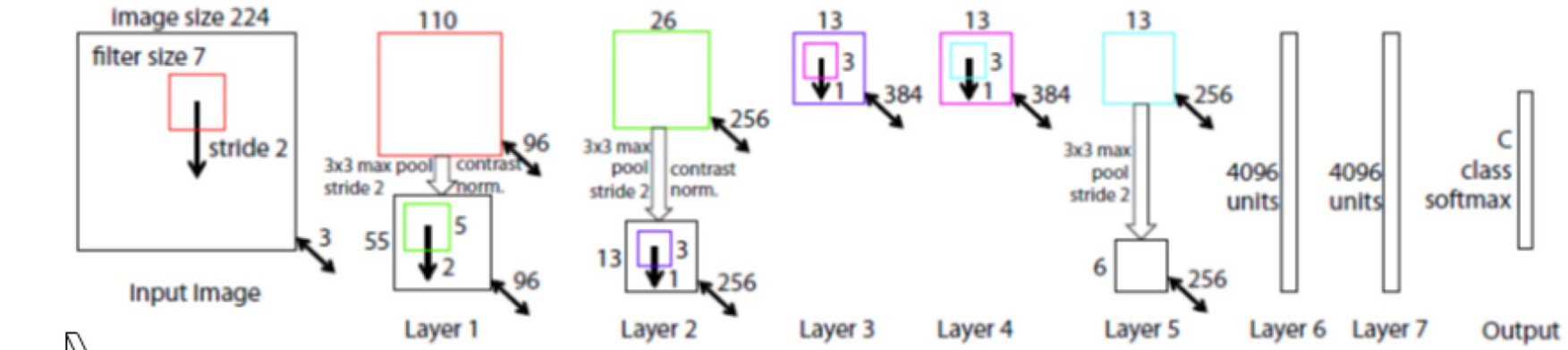
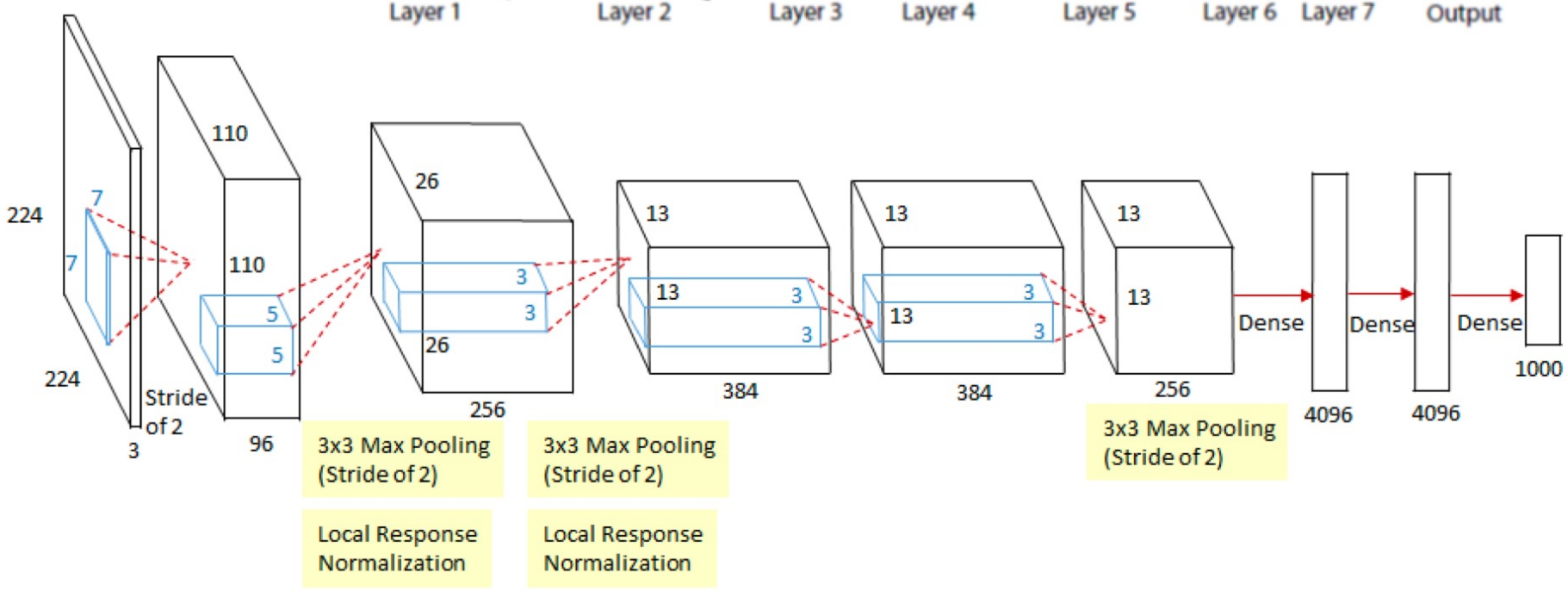
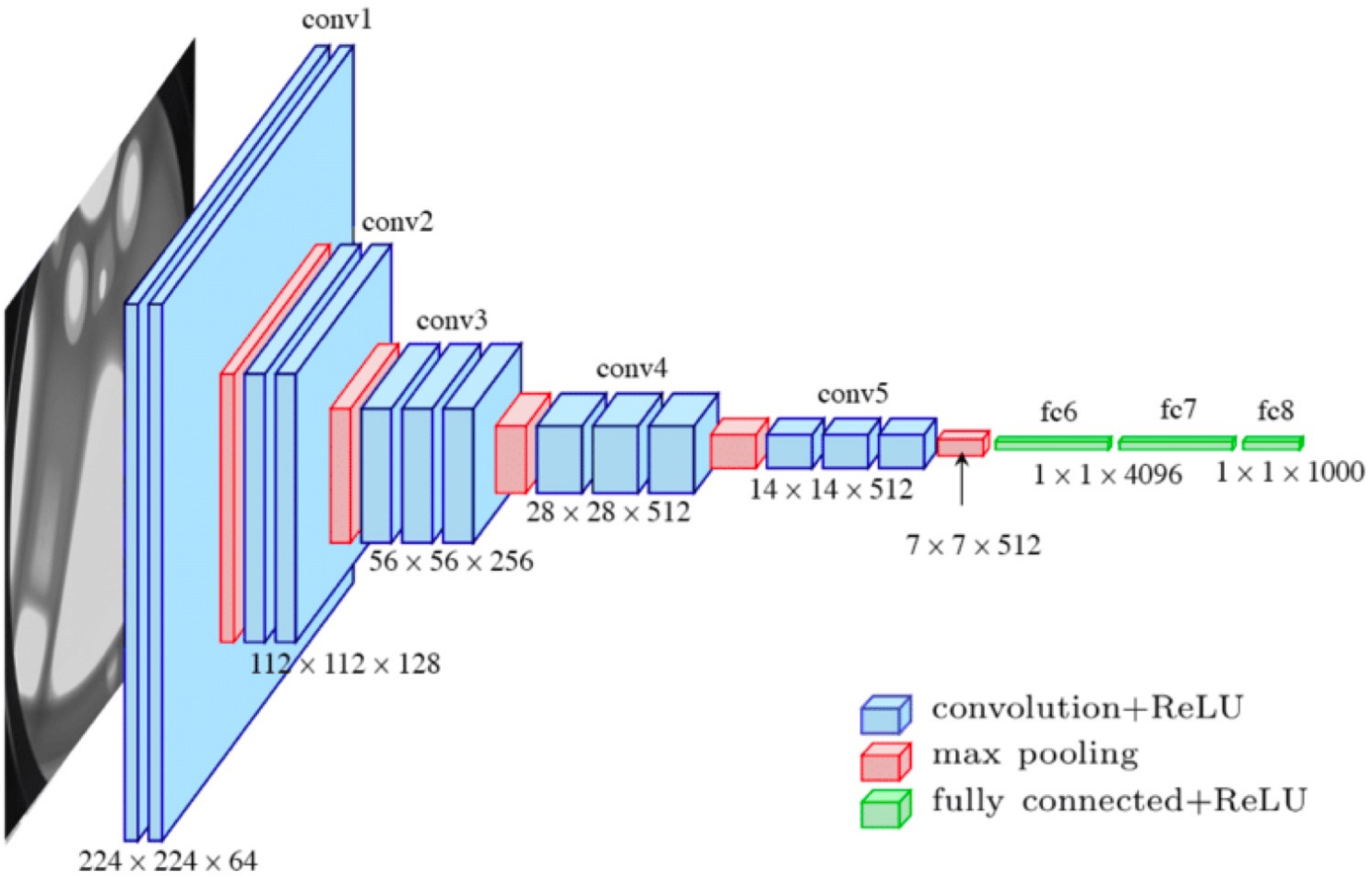


image size 224
filter size 7
stride 2
3
Input Image
110
96
3x3 max pool
stride 2
contrast
norm.
55
5
2
96
Layer 1
26
256
3x3 max
pool
stride 2
contrast
norm.
13
3
1
256
Layer 2
13
3
1
384
Layer 3
13
3
1
384
Layer 4
13
256
3x3 max
pool
stride 2
6
256
Layer 5
4096
units
Layer 6
4096
units
Layer 7
C
class
softmax
Output


224
7
7
224
3
Stride
of 2
110
110
5
5
96
3x3 Max Pooling
(Stride of 2)
Local Response
Normalization
26
3
3
26
256
3x3 Max Pooling
(Stride of 2)
Local Response
Normalization
13
13
3
3
384
13
13
3
3
384
13
13
256
3x3 Max Pooling
(Stride of 2)
Dense
4096
Dense
4096
Dense
1000

VGGNet (2014)

- 2014的亚军, Simonyan和Zisserman发表
- 使用了非常统一的结构, 和AlexNet类似, 使用了更小的滤波器, 使用了大量的滤波器
- 使用了4个GPU训练了3个星期
- 非常普遍的用于特征提取/图像表达
- 但是VGG包含1亿3千8百万参数


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					

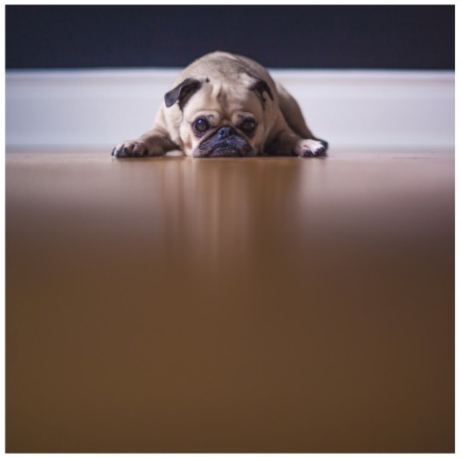
VGG 16

GoogleNet/Inception(2014)

- Top-5 错误率为6.67%
- 使用了Batch Normalization, 图像增强(扭曲), RMSprop优化器
- 使用了比较小的滤波器, 虽然有22层，但是总的参数为4百万, 少于AlexNet的6千万

Inception Net 想要解决的3个问题





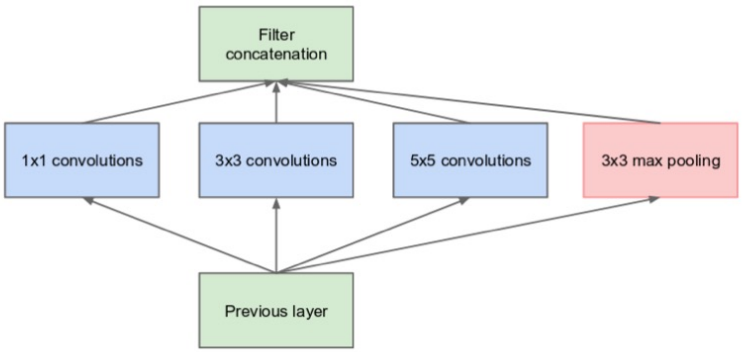
Inception Net 想要解决的3个问题

https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202

- 1. 需要不同尺寸的滤波器对待不同大小的对象

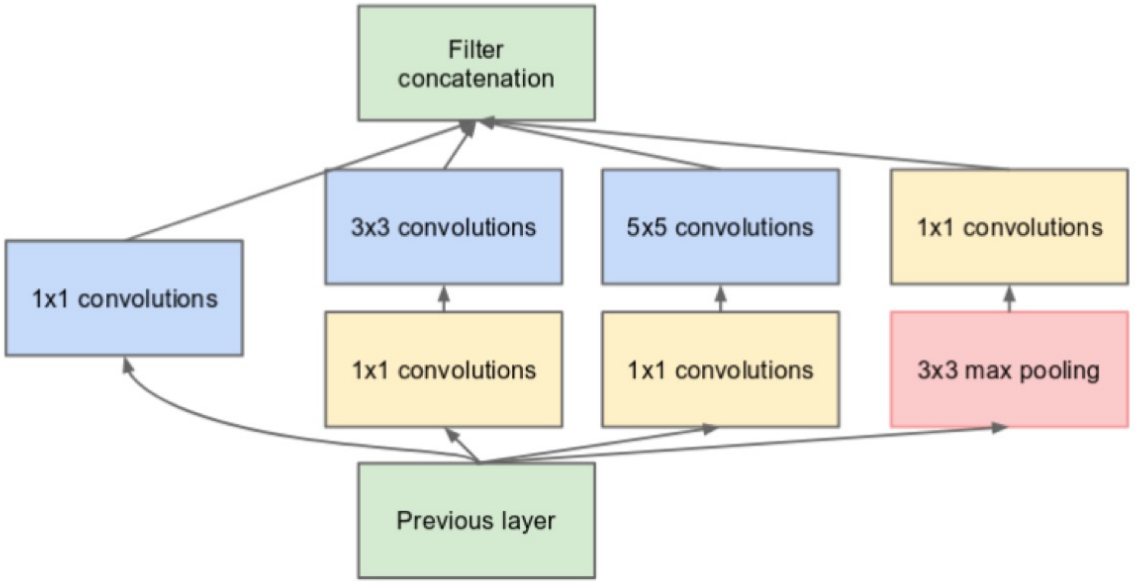
- 2. 太深的网络容易过拟合, 而且容易梯度消失
- 3. 将比较大的滤波器的卷积层直接链接计算量太大

使用不同尺寸的滤波器在同一层



(a) Inception module, naïve version

使用不同尺寸的滤波器在同一层, 并且使用1*1 Convolution 将输入channel压缩减小计算量



(b) Inception module with dimension reductions

Inception V3

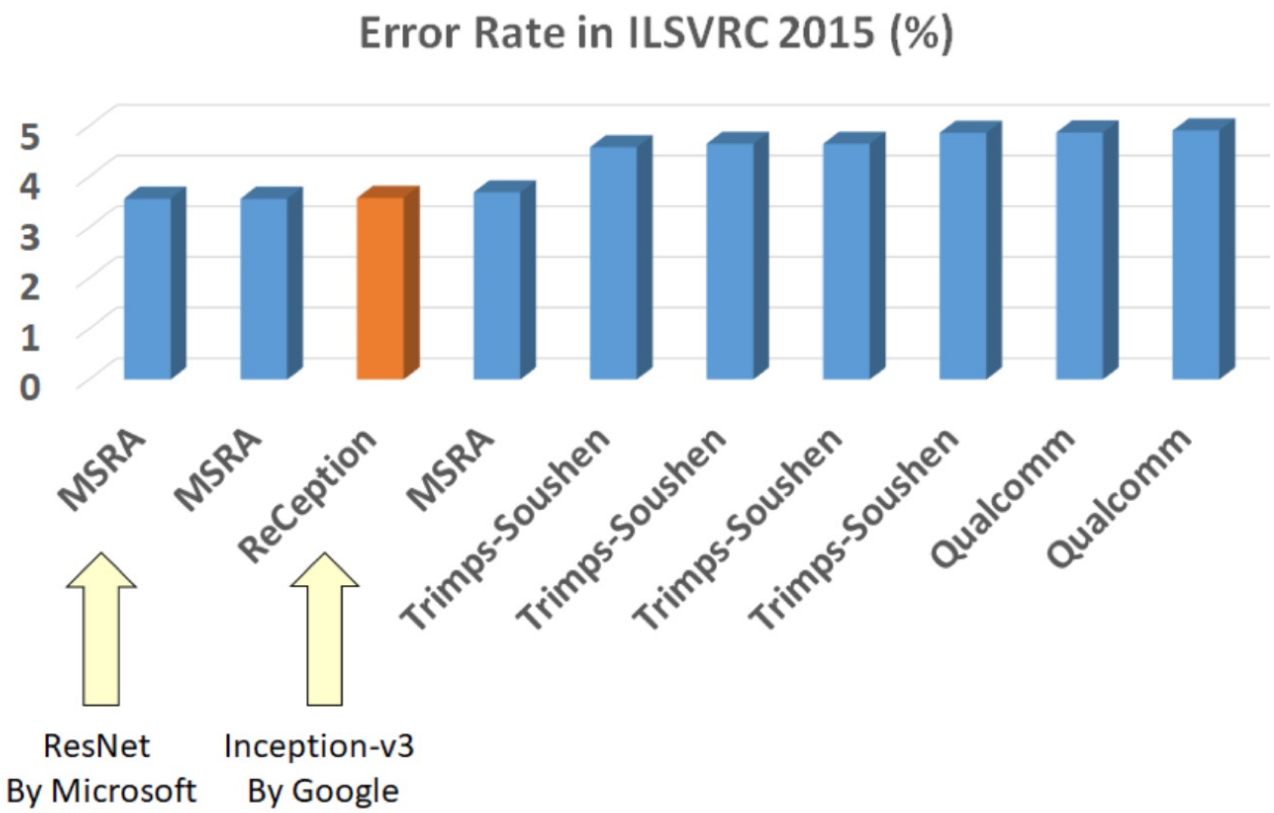


Image Classification Error Rate in ILSVRC 2015

Inception V3

- **Factorizing Convolutions (**卷积分解**)**
- **Efficient Grid Size Reduction**
- **Inception-v3 Architecture**

卷积分解

- 假设Y=conv2D(X,K)
- Y为输出, X为输入, K为卷积核
- 如果 K = P.dot(Q)
- 那么Y=conv2D(conv2D(X, P), Q)

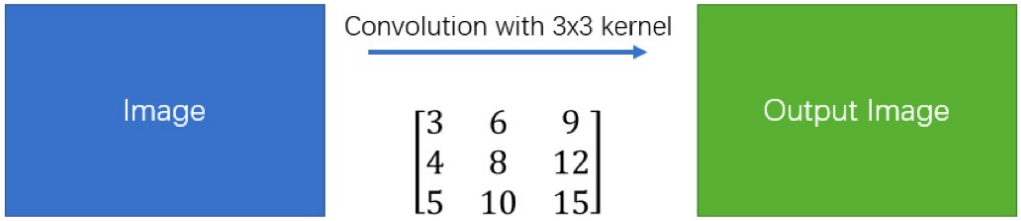
- $\begin{bmatrix}1\\2\\1\end{bmatrix}$ [1, 0, -1] = $\begin{bmatrix}1 & 0 & -1\\2 & 0 & -2\\1 & 0 & -1\end{bmatrix}$

- 分解的做法参数更少, 例如左边是6个参数, 右边是9个

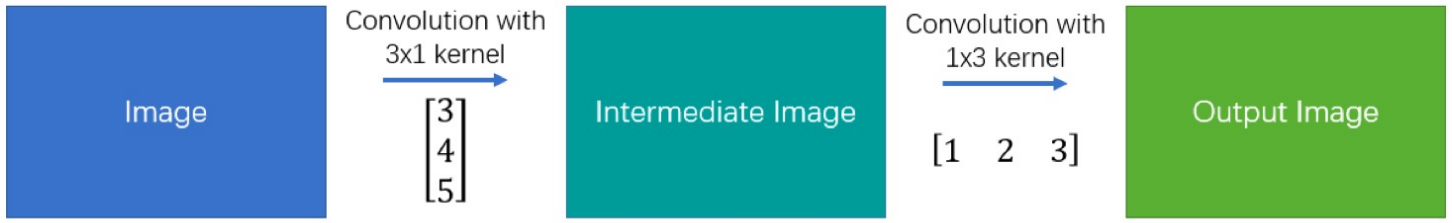
A spatial separable convolution simply divides a kernel into two, smaller kernels. The most common case would be to divide a 3x3 kernel into a 3x1 and 1x3 kernel, like so:

$$\begin{bmatrix} 3 & 6 & 9 \\ 4 & 8 & 12 \\ 5 & 10 & 15 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

Simple Convolution



Spatial Separable Convolution



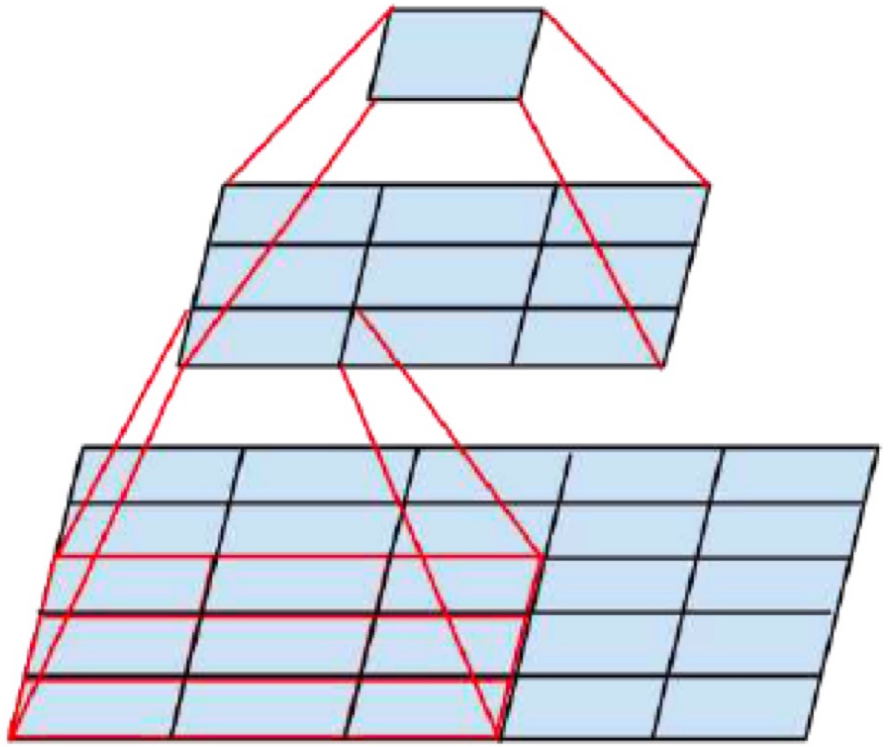
Factorizing Convolutions

- The aim of factorizing Convolutions is to **reduce the number of connections/parameters** without decreasing the network efficiency.

Factorization Into Smaller Convolutions

Two 3×3 convolutions replaces one 5×5 convolution as follows:

By using **1 layer of 5×5 filter**, number of parameters = **5×5=25**
By using **2 layers of 3×3 filters**, number of parameters = **3×3+3×3=18**
Number of parameters is reduced by 28%

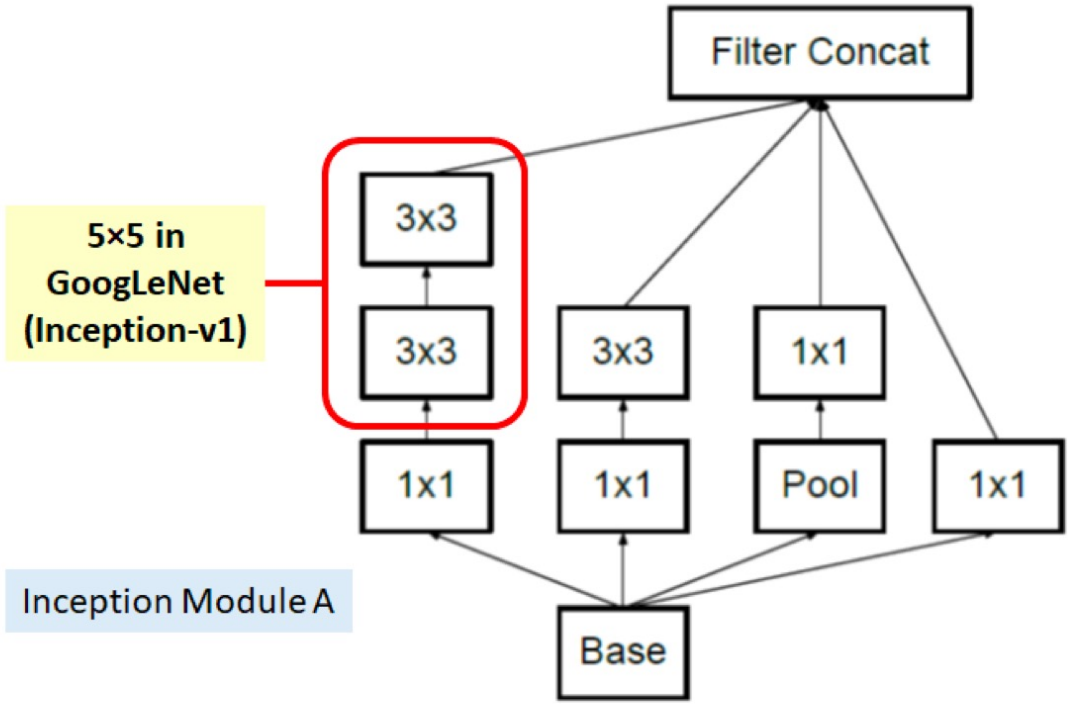


Two 3×3 convolutions replacing one 5×5 convolution

Factorization Into Smaller Convolutions

Two 3×3 convolutions replaces one 5×5 convolution as follows:

By using **1 layer of 5×5 filter**, number of parameters = **5×5=25**
By using **2 layers of 3×3 filters**, number of parameters = **3×3+3×3=18**
Number of parameters is reduced by 28%

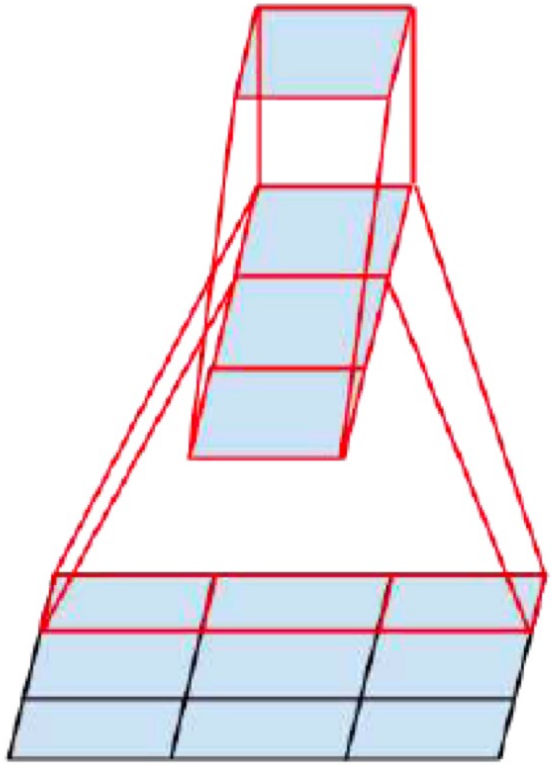


Two 3×3 convolutions replacing one 5×5 convolution

Factorization Into Asymmetric Convolutions

One 3×1 convolution followed by one 1×3 convolution replaces one 3×3 convolution as follows:

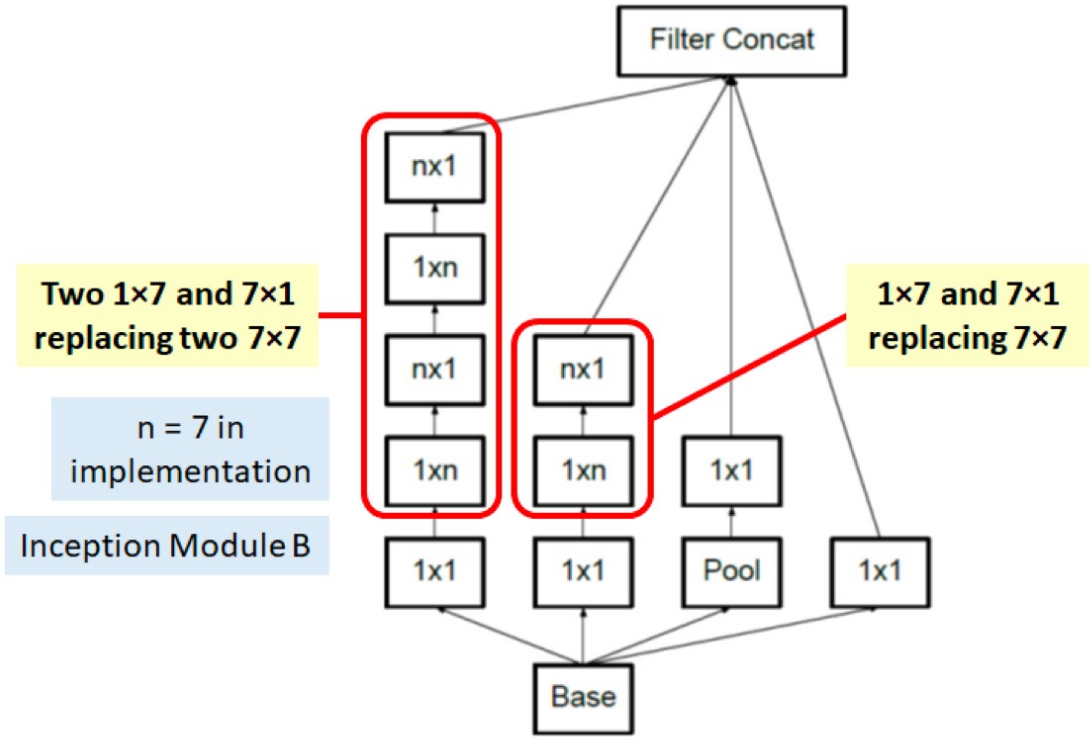
By using **3×3 filter**, number of parameters = **3×3=9**
By using **3×1 and 1×3 filters**, number of parameters = **3×1+1×3=6**
Number of parameters is reduced by 33%



Factorization Into Asymmetric Convolutions

One 3×1 convolution followed by one 1×3 convolution replaces one 3×3 convolution as follows:

By using **3×3 filter**, number of parameters = **3×3=9**
By using **3×1 and 1×3 filters**, number of parameters = **3×1+1×3=6**
Number of parameters is reduced by 33%



卷积分解

- 假设Y=conv2D(X,K)
- Y为输出, X为输入, K为卷积核
- 如果 K = P.dot(Q)
- 那么Y=conv2D(conv2D(X, P), Q)

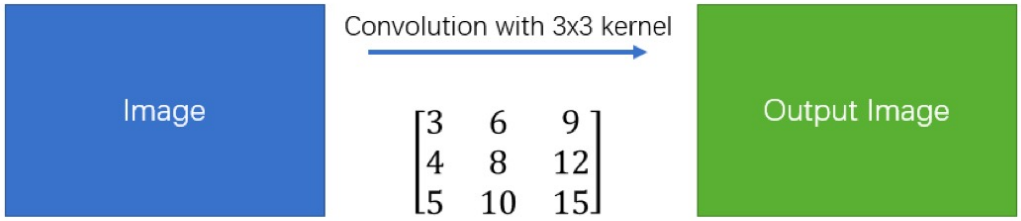
- $\begin{bmatrix}1\\2\\1\end{bmatrix}$ [1, 0, -1] = $\begin{bmatrix}1 & 0 & -1\\2 & 0 & -2\\1 & 0 & -1\end{bmatrix}$

- 分解的做法参数更少, 例如左边是6个参数, 右边是9个

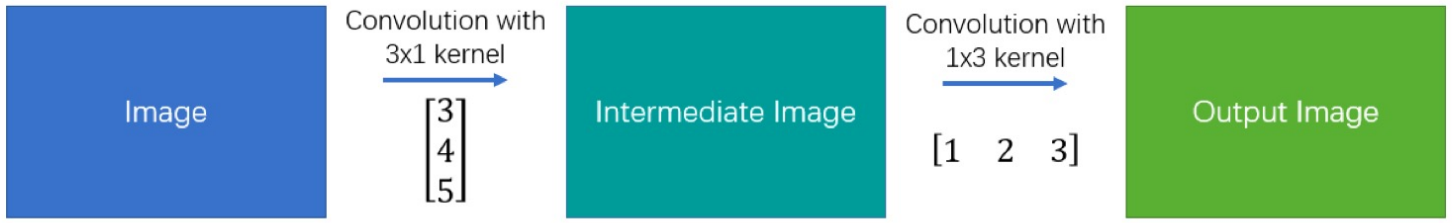
A spatial separable convolution simply divides a kernel into two, smaller kernels. The most common case would be to divide a 3x3 kernel into a 3x1 and 1x3 kernel, like so:

$$\begin{bmatrix} 3 & 6 & 9 \\ 4 & 8 & 12 \\ 5 & 10 & 15 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

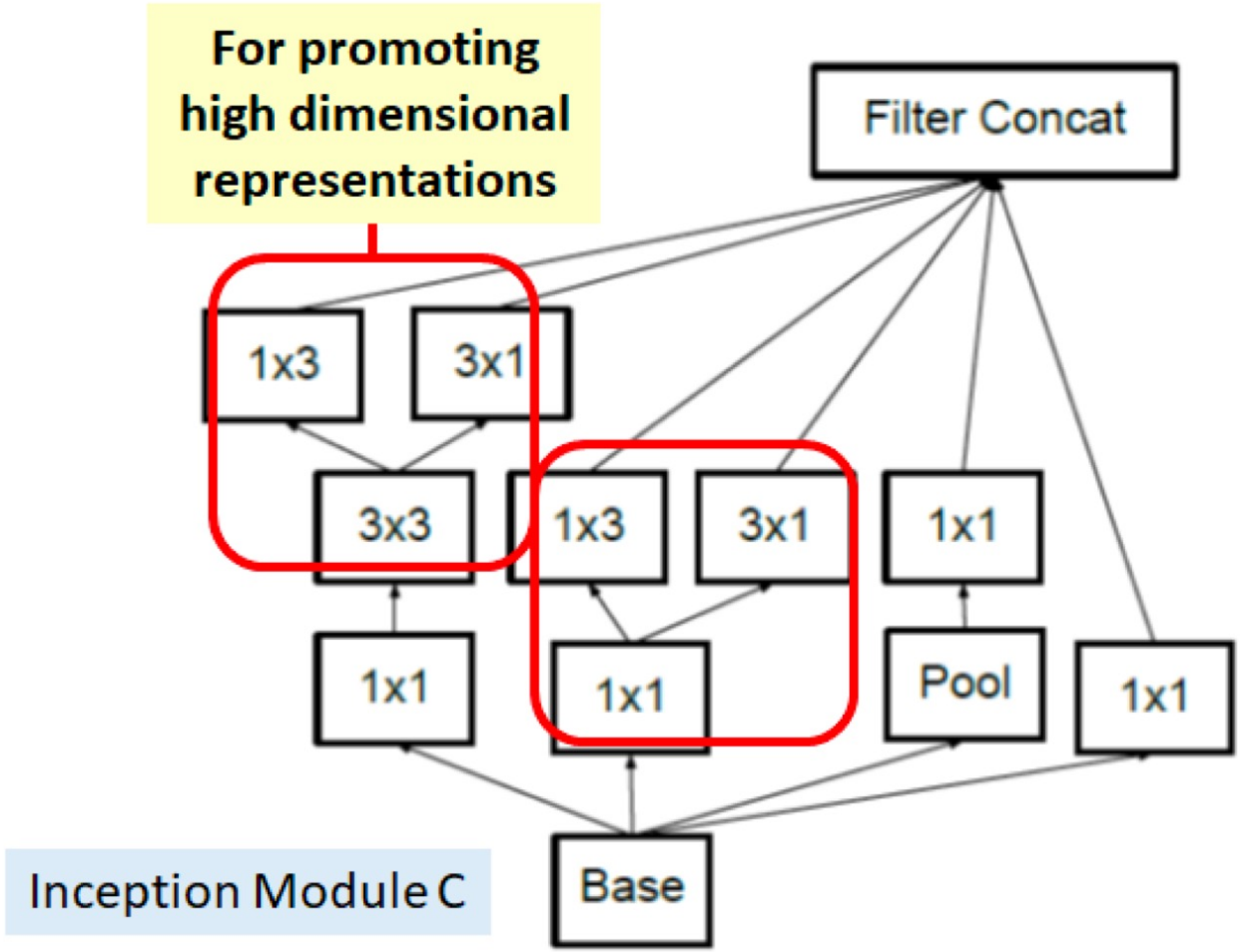
Simple Convolution



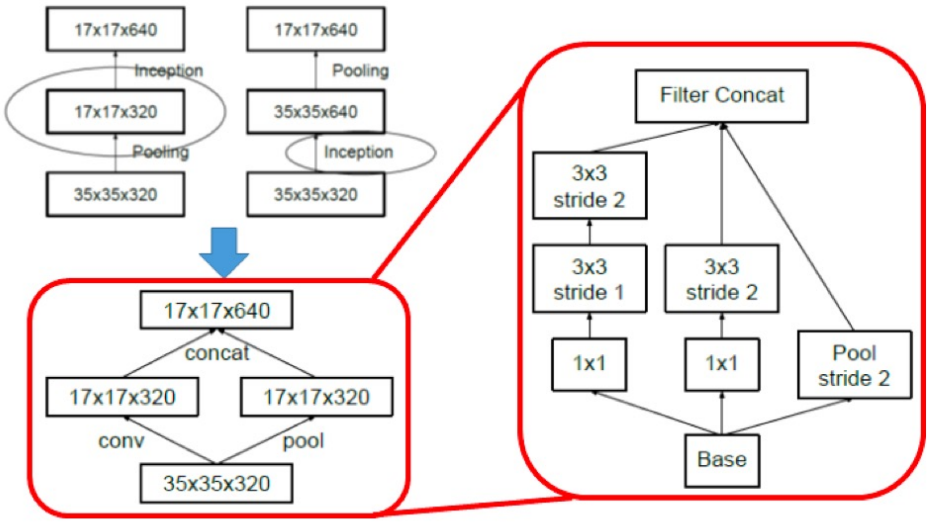
Spatial Separable Convolution



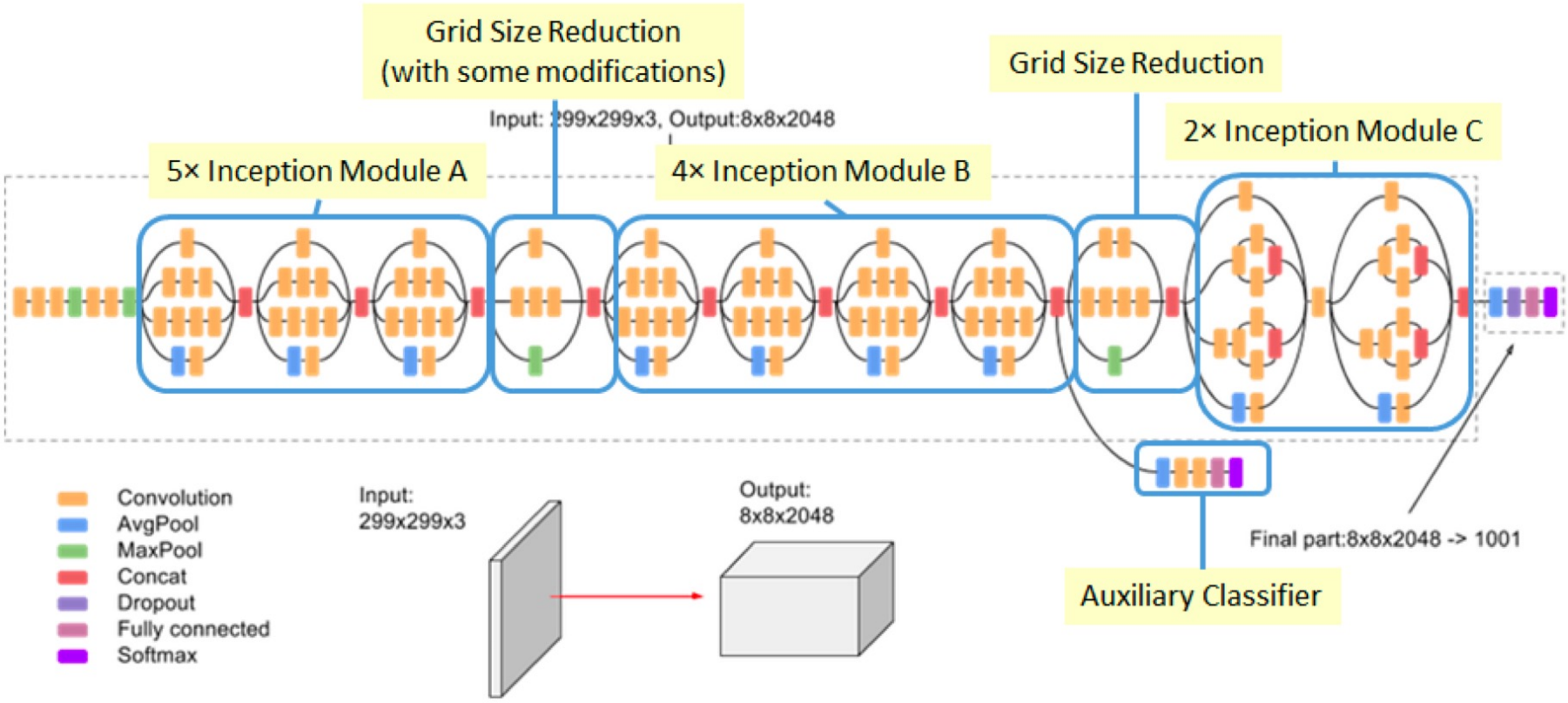
Promoting high dimensional representations



Efficient Grid Size Reduction



- **Conventionally**, such as AlexNet and VGGNet, the feature map downsizing is done by max pooling. But the drawback is either **too greedy by max pooling followed by conv layer**, or **too expensive by conv layer followed by max pooling**.
- With the efficient grid size reduction, **320 feature maps** are done by **conv with stride 2**. **320 feature maps** are obtained by **max pooling**. And these 2 sets of feature maps are **concatenated as 640 feature maps** and go to the next level of inception module.


Grid Size Reduction
(with some modifications)
Input: 299x299x3, Output:8x8x2048
Grid Size Reduction
2× Inception Module C
5× Inception Module A
4× Inception Module B
Convolution
AvgPool
MaxPool
Concat
Dropout
Fully connected
Softmax
Input:
299x299x3
Output:
8x8x2048
Final part:8x8x2048 -> 1001
Auxiliary Classifier

ResNet(2015)

- Kaiming He 等发表
- 包含 “skip connections”, batch normalization
- 特别深, 152 层
- Top-5 错误率为3.57%, 超过人类的能力

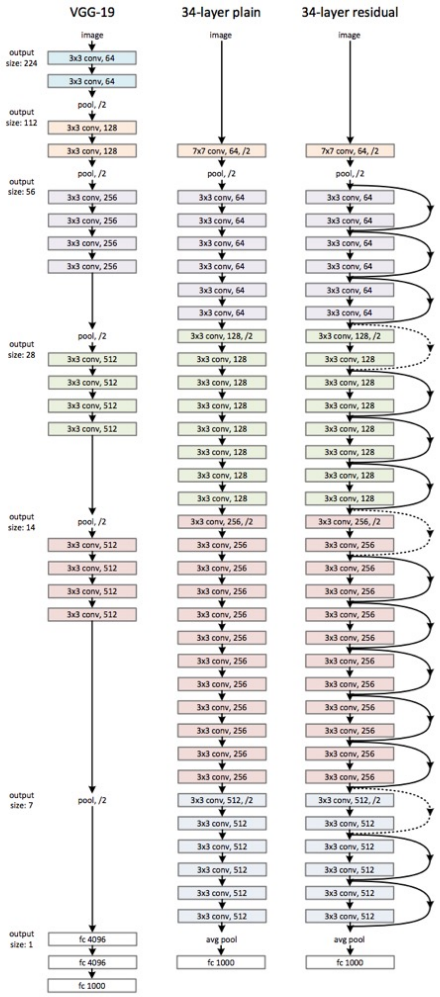


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Mid-**

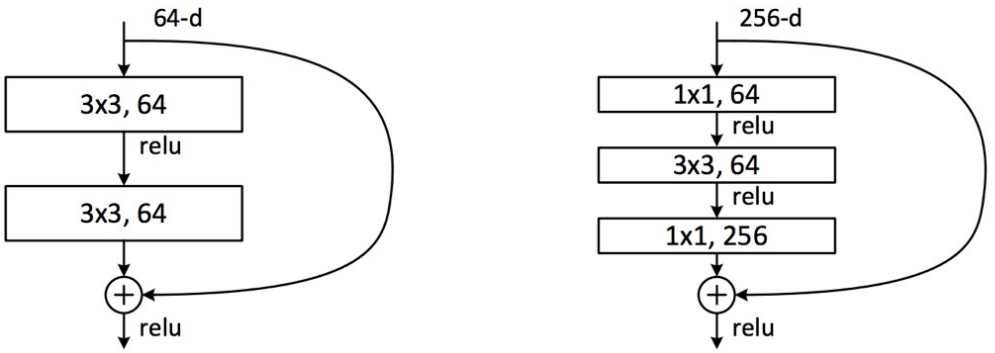
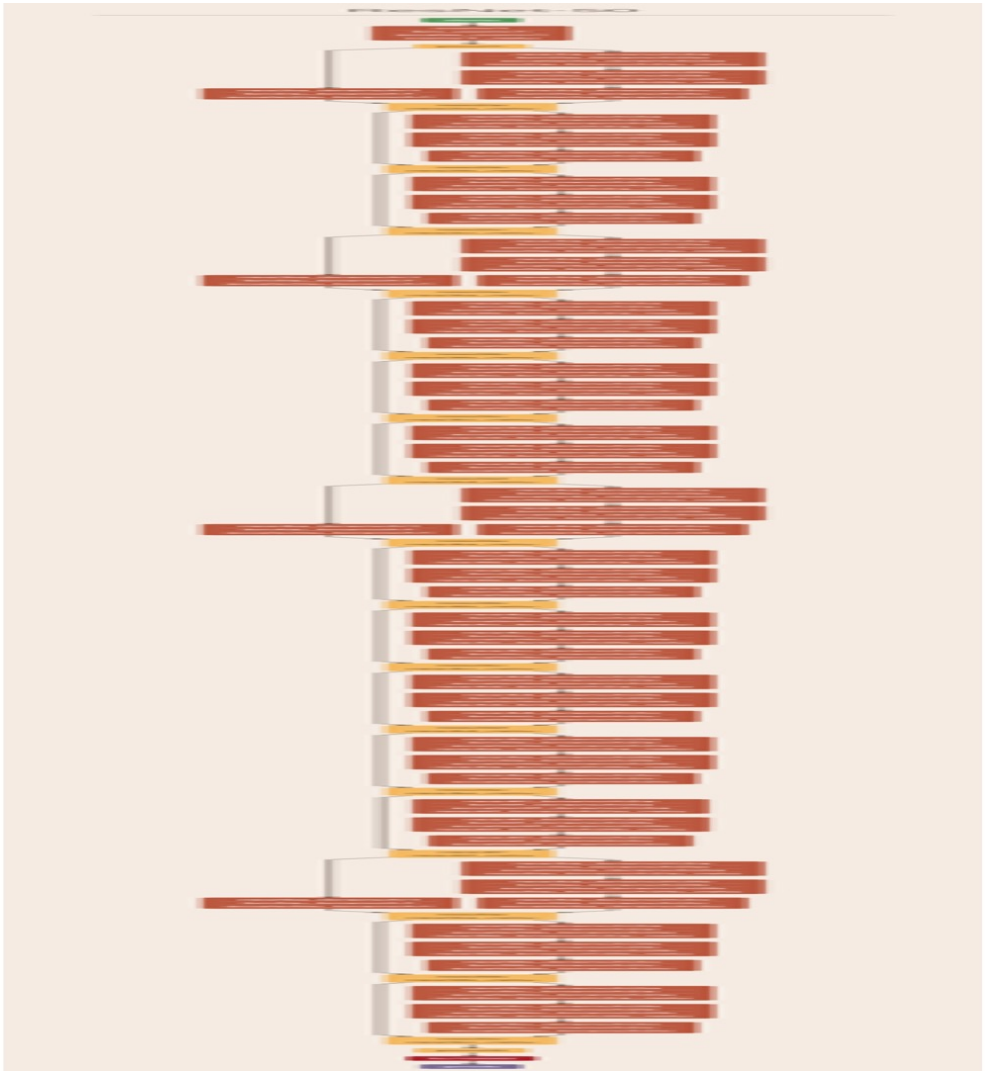


Figure 5. A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.

Res50



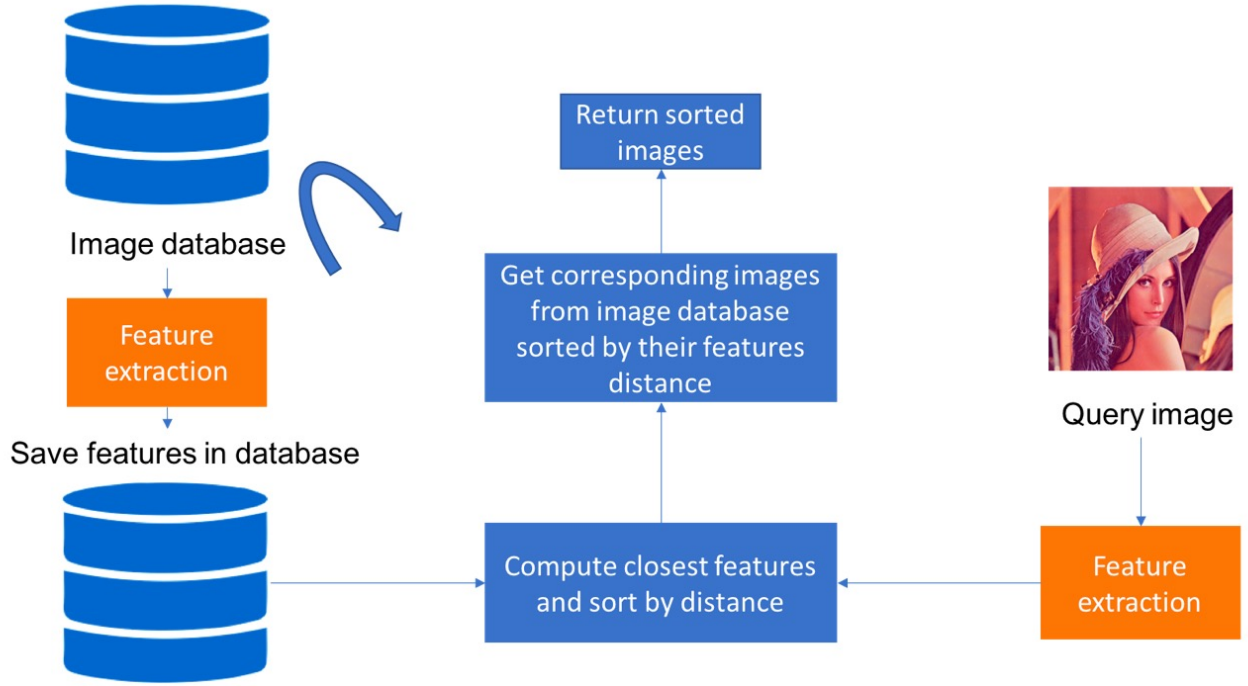
基于内容的图像检索

- **基于内容的图像检索**（CBIR）系统使得用户能够在图像数据集中找到与查询图像类似的图像。最着名的CBIR系统是Google图像搜索功能。我们一起使用**keras深度学习框架在MNIST数据集上执行图像检索**。
- 我们的CBIR系统将基于**卷积去噪自动编码器**。它是一类**无监督的深度学习**算法。

两个图像检索框架：基于文本和基于内容

- 基于文本的方法可以追溯到20世纪70年代。在这样的系统中，图像由文本注释，然后由数据库管理系统使用文本来执行图像检索。这种方法有两个缺点:
  - 人工注释需要相当大的人力开销。
  - 由于人类感知的主观性而导致的注释不准确性。
- 为克服基于文本的检索系统中的上述缺点，在20世纪80年代初引入了基于内容的图像检索（CBIR）。**在CBIR中，图像通过其视觉内容（例如颜色，纹理，形状）来索引。**

首先从图像数据库中**提取特征**并存储它。然后我们计算与**查询图像**相关的特征。最后，我们**检索具有最近特征的图像**



基于内容的图像检索的特征提取

- 基于内容的图像检索的**关键**是**特征提取**。这些特征对应于我们在高层次上表示图像的方式。如何描述图像上的颜色？它的质地？它上面的形状？我们提取的功能还应该允许有效地检索图像。当我们有一个数据量巨大的图像数据库，尤其如此。
- 有很多方法可以提取这些功能。
- 一种方法是使用我们称之为**手工制作的功能**。实例是：**颜色直方图** 来定义颜色，**方向梯度直方图** 以限定的形状。
- 其他描述符如**SIFT**和**SURF**已被证明对图像检索应用具有鲁棒性。

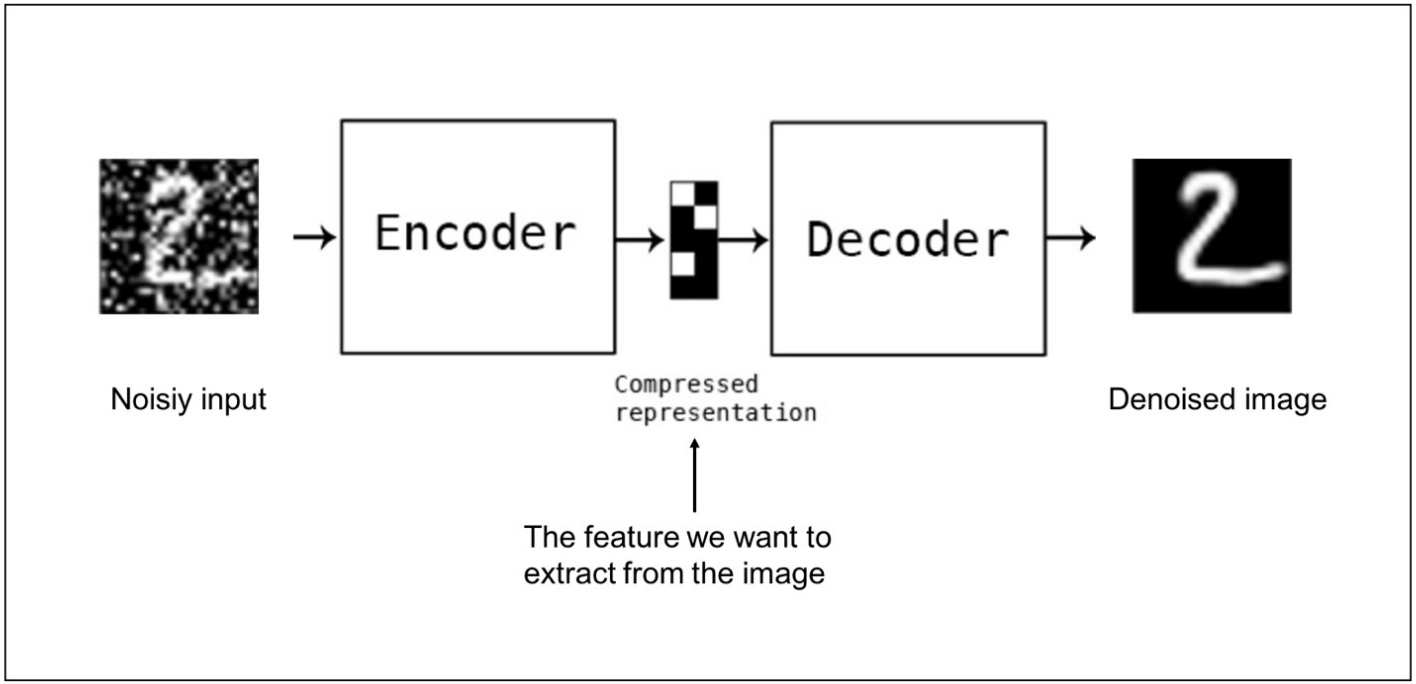
基于内容的图像检索的特征提取

- 另一种可能性是使用深度学习算法。在这篇研究论文中(https://arxiv.org/pdf/1404.1777.pdf)，作者证明了为分类目的而训练的**卷积神经网络**（CNN）可用于提取图像的“神经代码”。这些神经代码是用于描述图像的特征。
- 研究表明这种方法在许多数据集上的表现与最先进的方法一样。这种方法的问题是我们首先需要标记数据来训练神经网络。**标签任务可能是昂贵且耗时的**。为我们的图像检索任务生成这些“神经代码”的另一种方法是使用无监督的深度学习算法。这是**去噪自动编码器的**来源。

去噪自动编码器

- **降噪自动编码器**是值**用于图像去噪的馈神经网络**。**降噪自动编码器**学习用于训练它的图像上的重要特征。然后，**它可用于**从类似图像中**提取特征**。

去噪自动编码器


Encoder
Decoder
Noisiy input
Compressed representation
Denoised image
The feature we want to extract from the image

去噪自动编码器用于图像检索

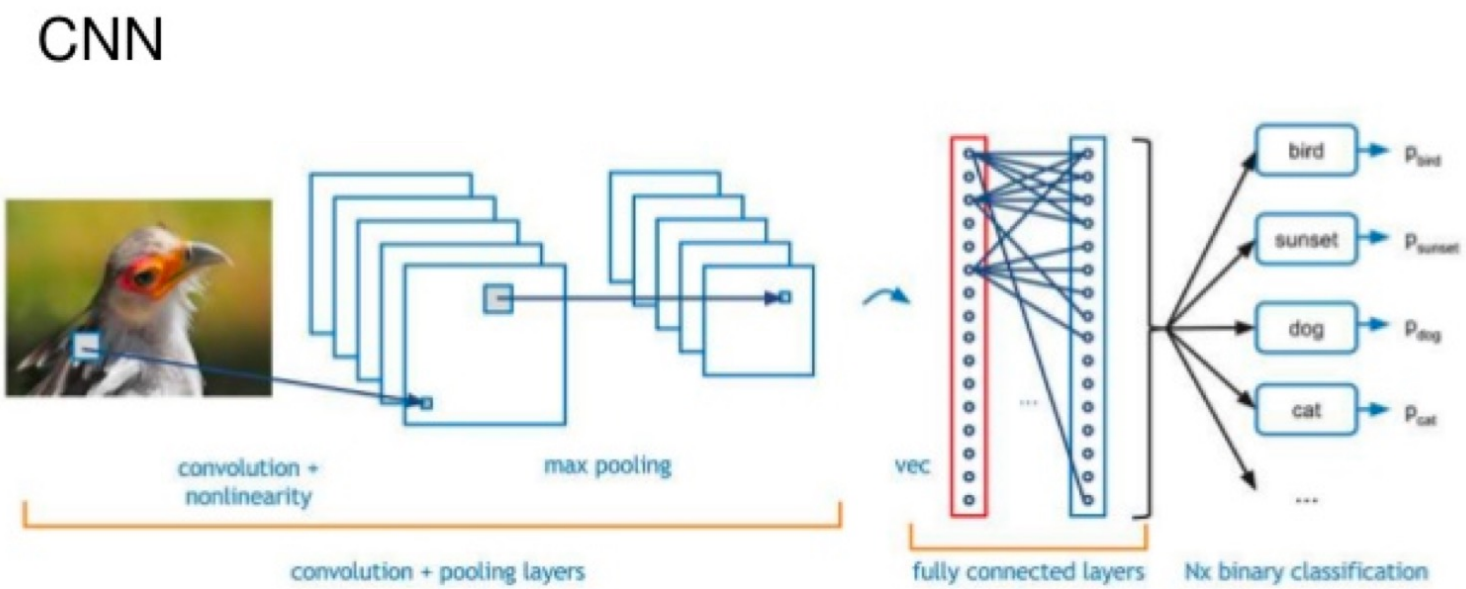
查询图片

前10个检索到的图像

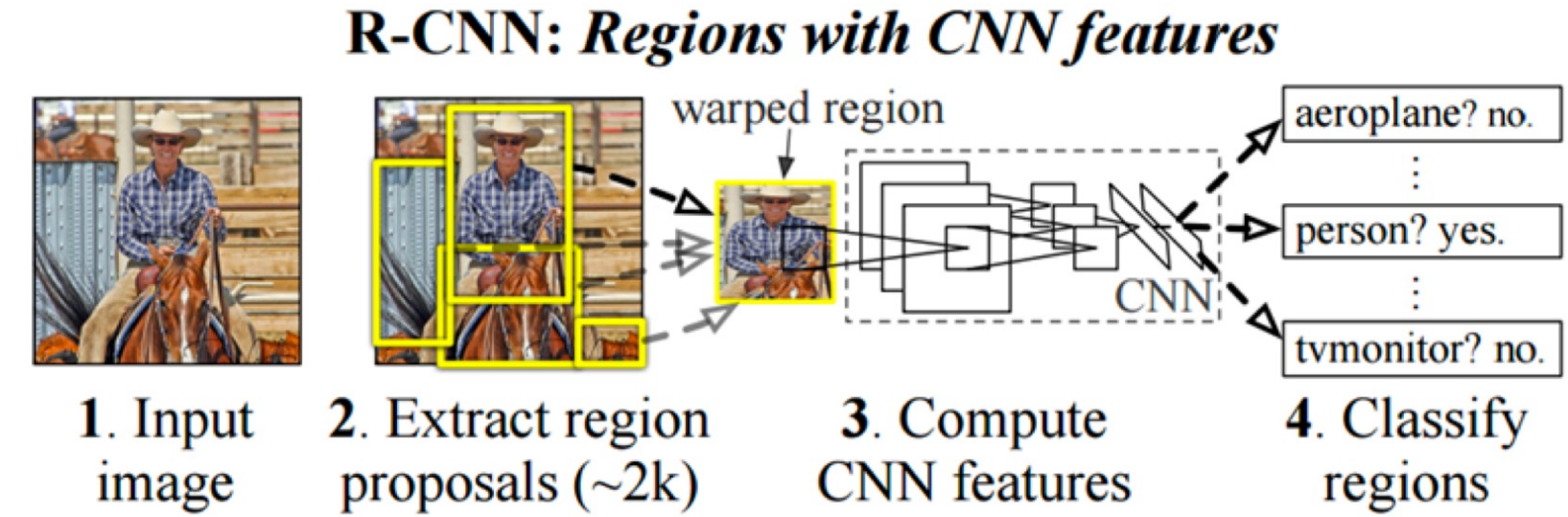
使用卷积神经网络进行语义图像嵌入

- 图像分类
- 目标检测
- 图像分割
- 图像合成
- 风格化转移
- 自动驾驶
- 图像检索
- 图像去噪
- 图像描述自动生成
- 图像超分辨率重构
- 图像深度预测
- 工业探伤
- ……

图像分类


CNN
convolution + nonlinearity
max pooling
vec
bird
sunset
dog
cat
convolution + pooling layers
fully connected layers
Nx binary classification

目标检测



R-CNN workflow

图像分割

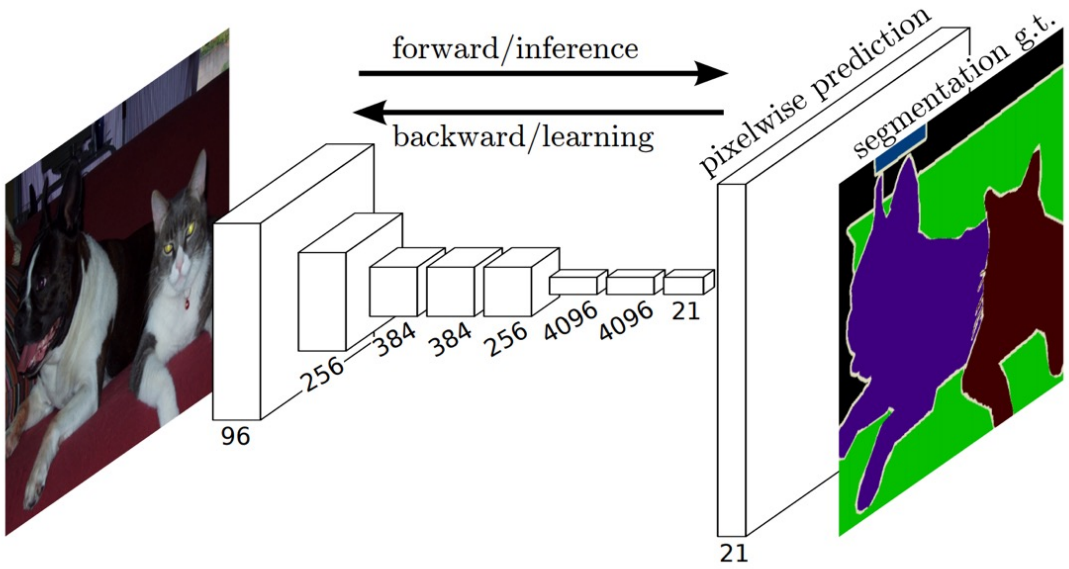
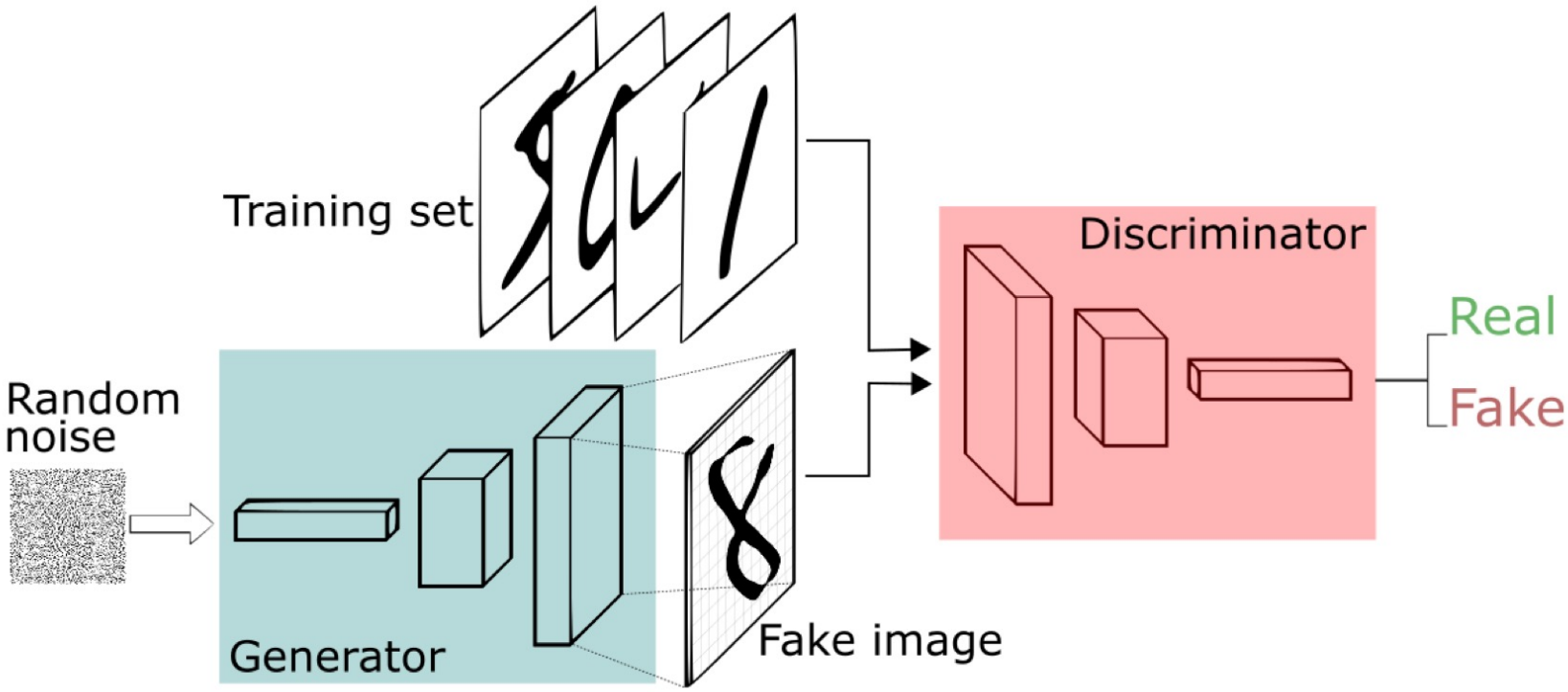
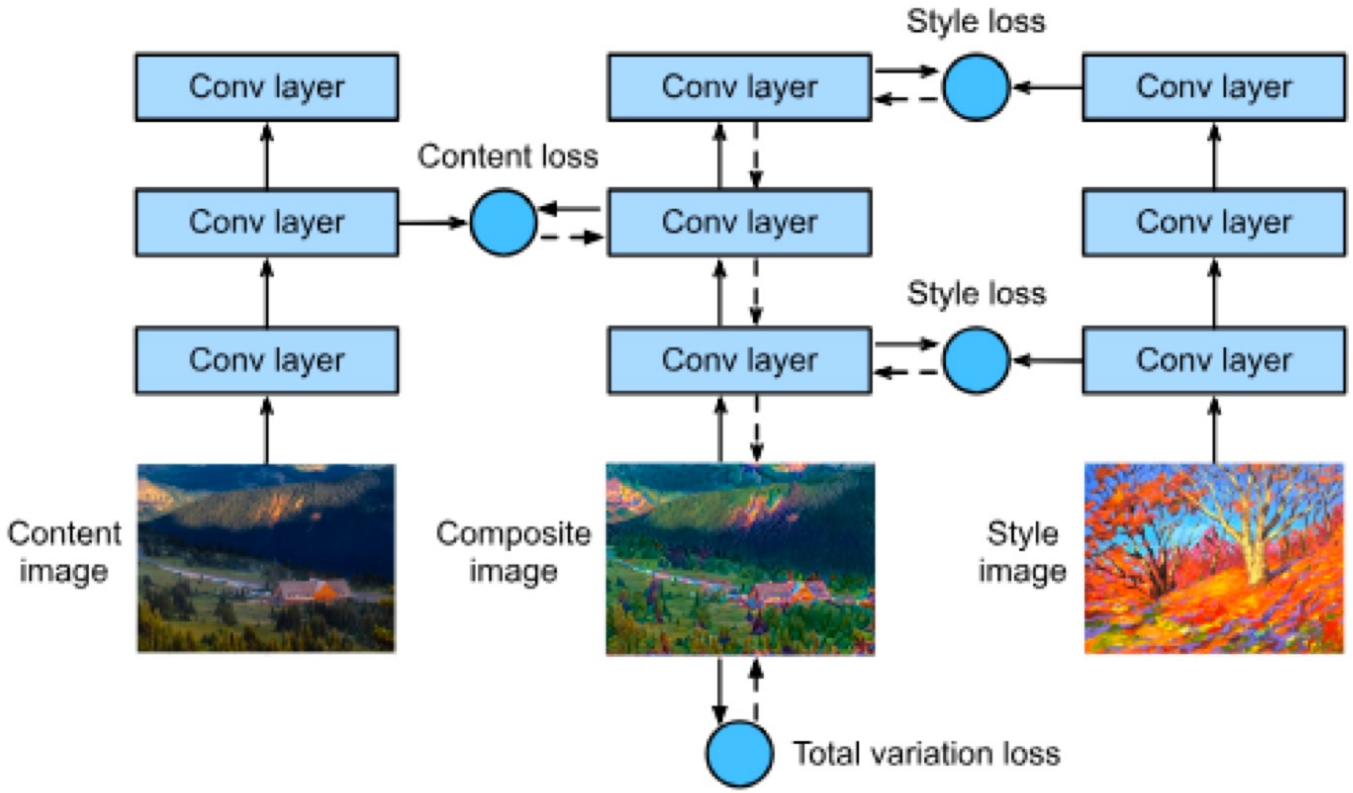


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

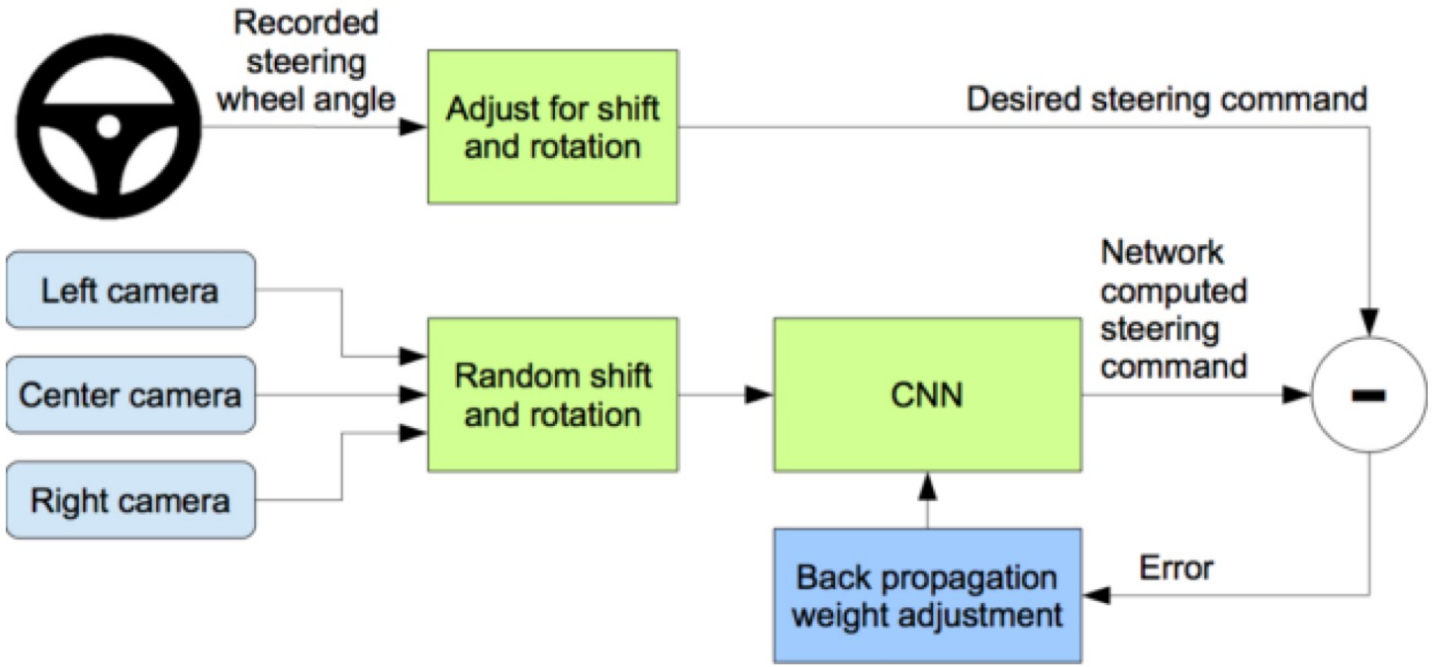
图像合成


Training set
Discriminator
Real
Fake
Random noise
Generator
Fake image

风格化转移


Style loss
Conv layer
Conv layer
Conv layer
Content loss
Conv layer
Conv layer
Conv layer
Style loss
Conv layer
Conv layer
Conv layer
Content image
Composite image
Style image
Total variation loss

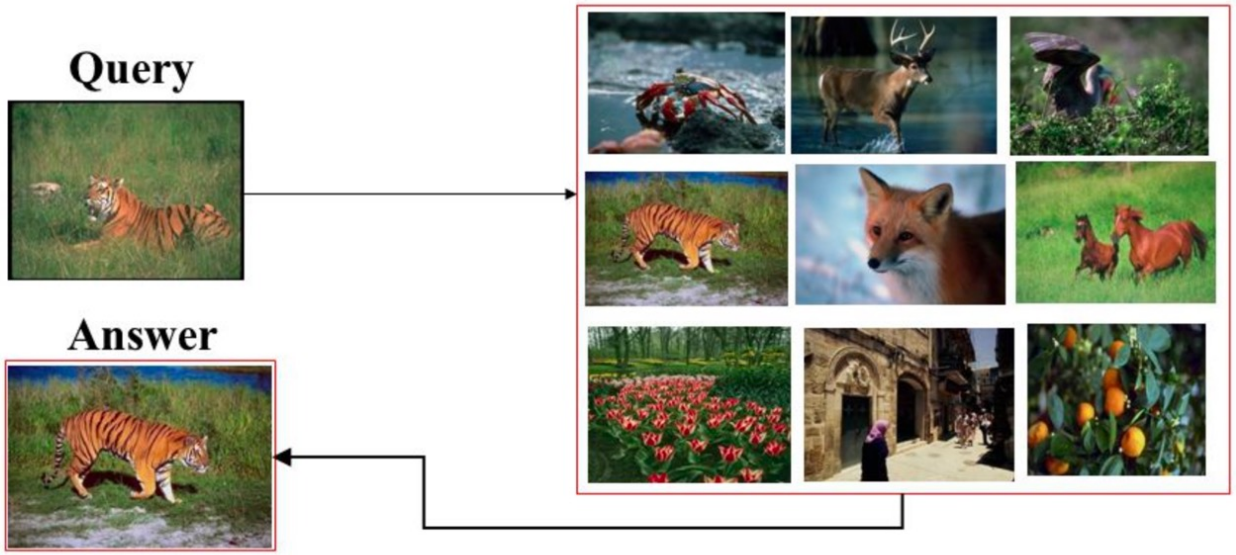
自动驾驶


Recorded steering wheel angle
Adjust for shift and rotation
Desired steering command
Left camera
Center camera
Right camera
Random shift and rotation
CNN
Network computed steering command
Back propagation weight adjustment
Error

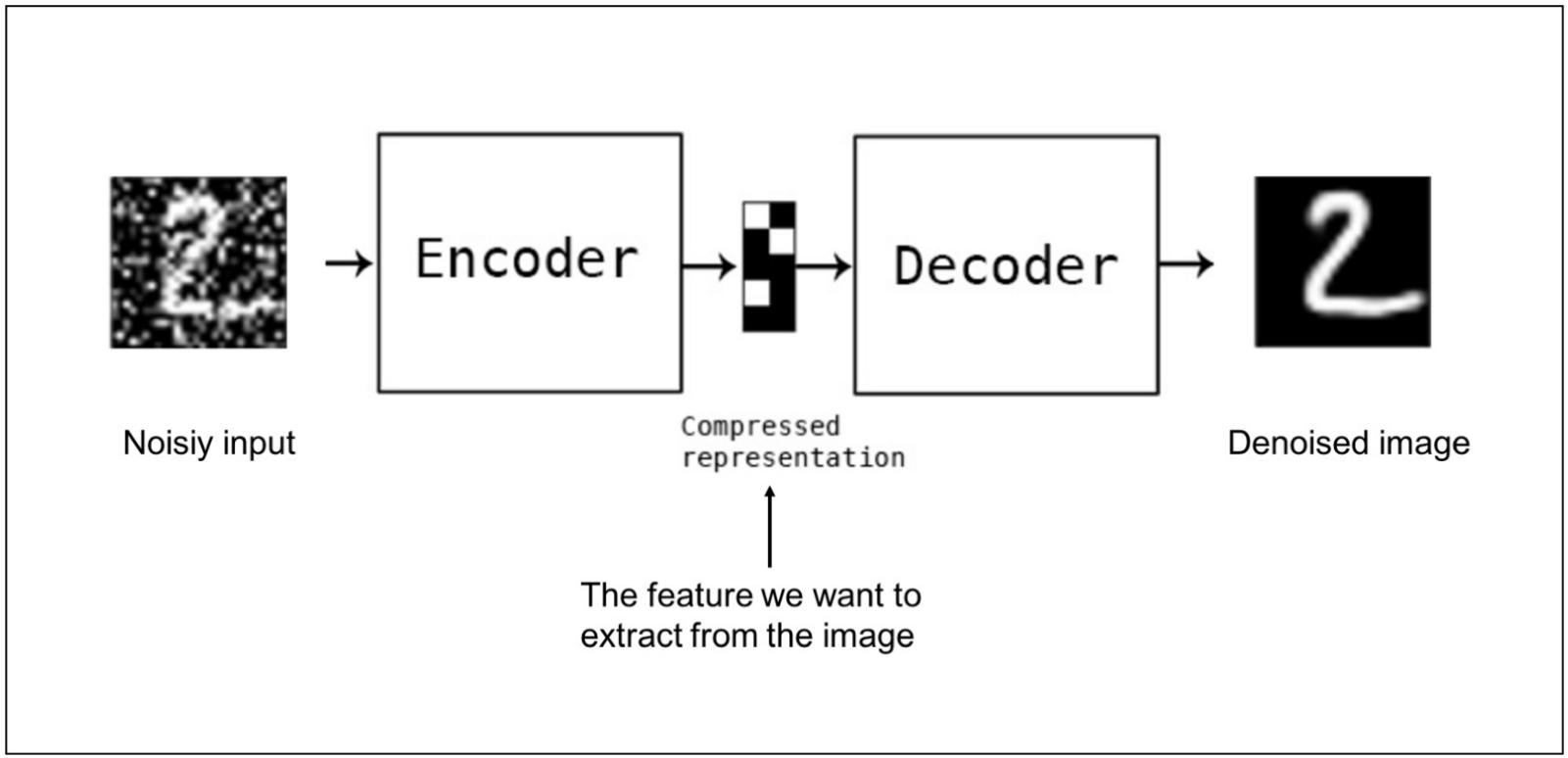
图像检索

Content-based Image Retrieval

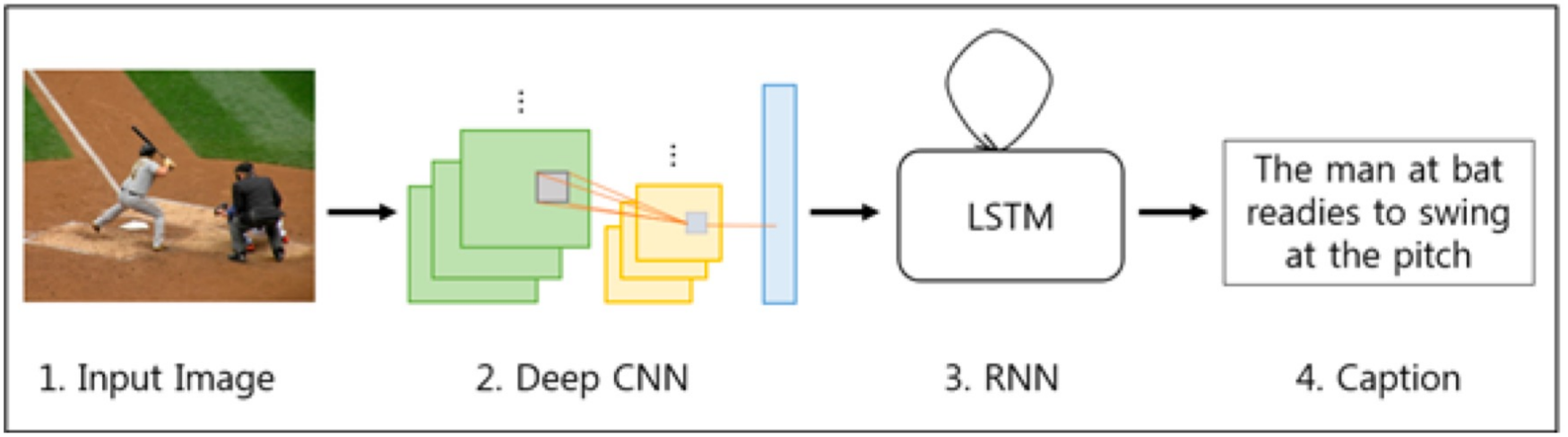
Given a query image, try to find visually similar images from an image database



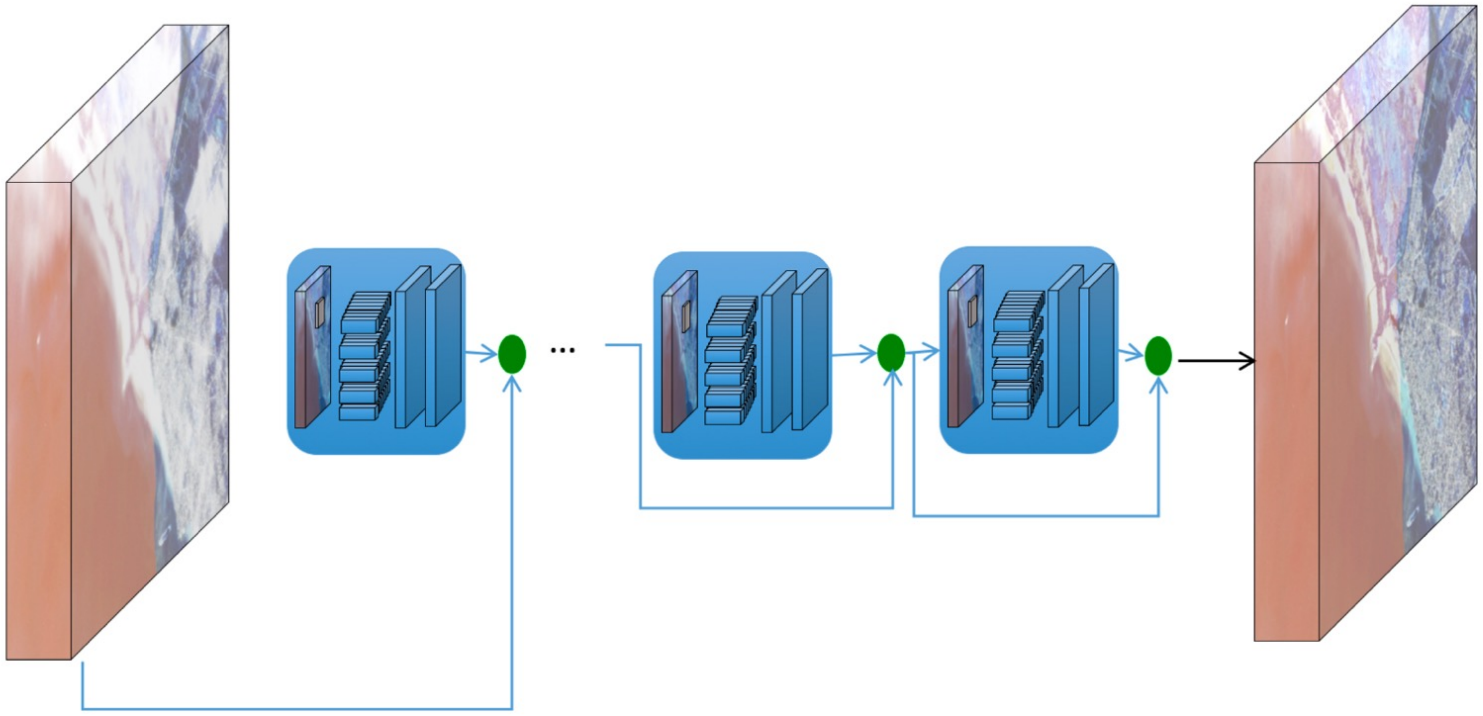
图像去噪


Encoder
Decoder
Noisiy input
Compressed representation
Denoised image
The feature we want to extract from the image

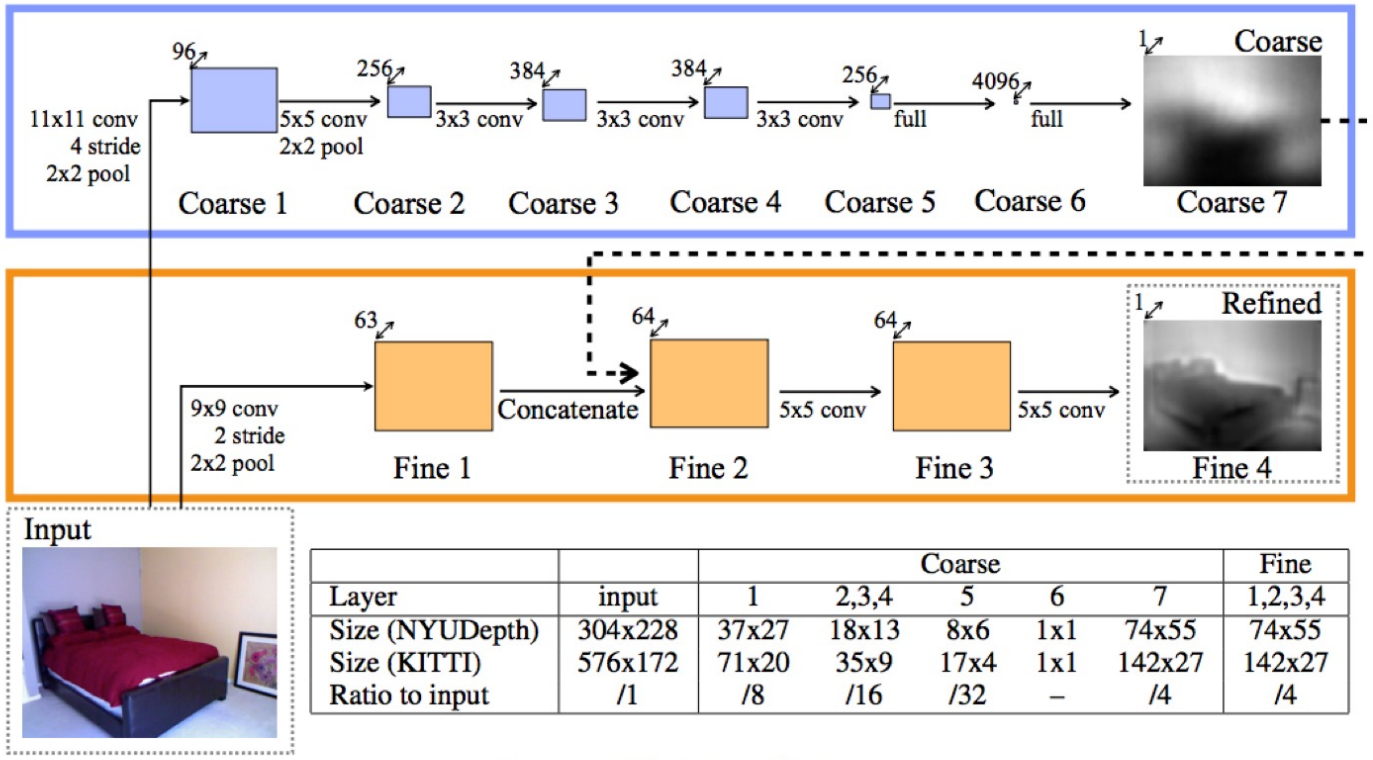
图像描述自动生成


LSTM
The man at bat readies to swing at the pitch
1. Input Image
2. Deep CNN
3. RNN
4. Caption

图像超分辨率重构



图像深度预测



		Coarse					Fine
Layer	input	1	2,3,4	5	6	7	1,2,3,4
Size (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Size (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Ratio to input	/1	/8	/16	/32	–	/4	/4

Figure 1: Model architecture.

工业探伤

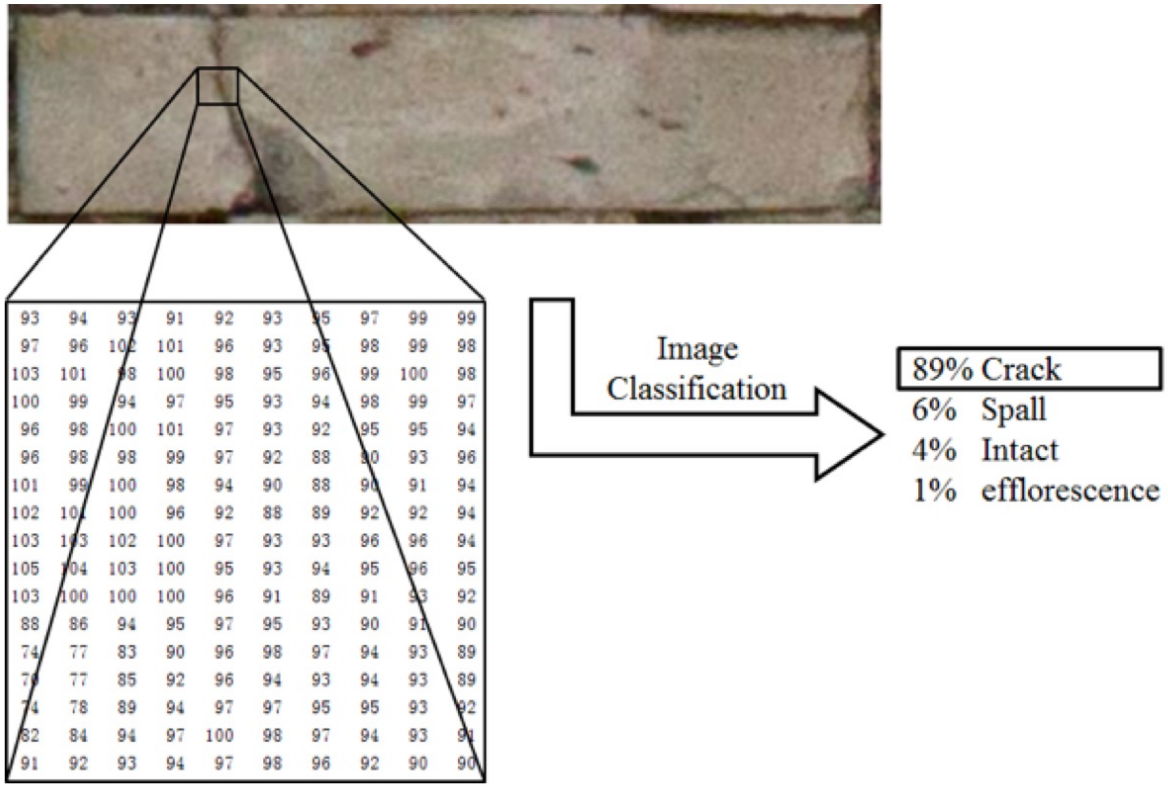


Image
Classification
89% Crack
6% Spall
4% Intact
1% efflorescence

The End