法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。



关注 小象学院

第11讲 通用场景下的图像分割

计算机视觉

主讲人：叶梓

上海交通大学博士
主要研究方向：机器学习、深度学习、人工智能

本章结构

- 语义分割
  - FCN
  - DeepLab
- 三大数据集介绍
  - Pascal VOC
  - MSCOCO
  - Cityscapes
- 代码演示
  - DeepLab

图像分割

- 所谓图像分割指的是根据灰度、颜色、纹理和形状等特征把图像划分成若干互不交迭的区域，
- 并使这些特征在同一区域内呈现出相似性，而在不同区域间呈现出明显的差异性。

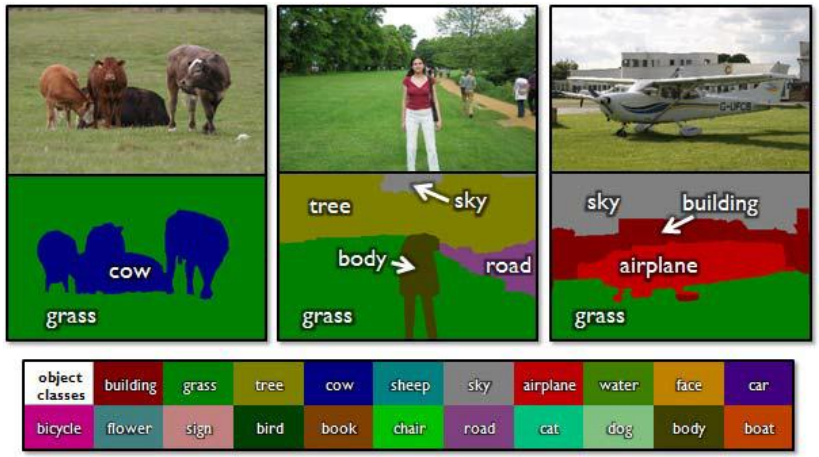
Pascal VOC数据集

- 视觉识别类竞赛的鼻祖，是包含了物体分类、目标检测、图像分割等任务。后续的ImageNet竞赛的任务设置就基本沿用了它的设定。
- 给定自然图片， 从中识别出特定物体，待识别的物体有20类：
  - person
  - bird, cat, cow, dog, horse, sheep
  - aeroplane, bicycle, boat, bus, car, motorbike, train
  - bottle, chair, dining table, potted plant, sofa, tv/monitor

语义分割（Semantic Segmentation）

什么是语义分割

- 目标
  - 从像素水平（pixel-level）上，理解、识别图片的内容
  - 根据语义信息分割



- 输入
  - 图片
- 输出
  - 同尺寸的分割标记（像素水平）
  - 每个像素会被识别为一个类别（category）

语义分割（Semantic Segmentation）

语义分割的用处

- 机器人视觉和场景理解
- 辅助/自动驾驶https://youtu.be/rB1BmBOkKTw
- 医学X光





语义分割（Semantic Segmentation）

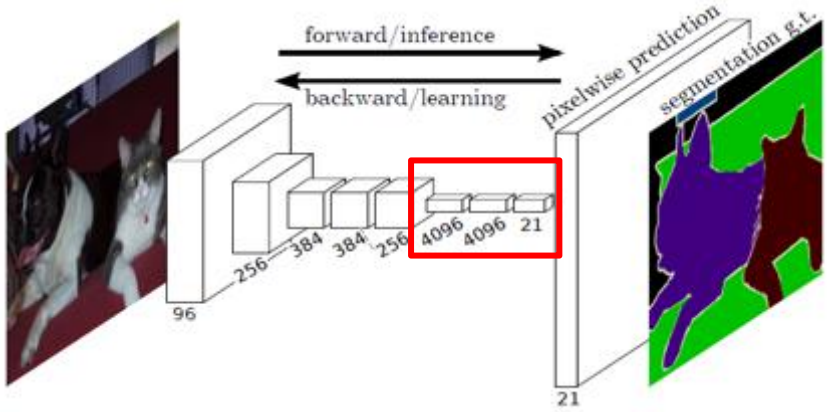
算法研究阶段

- 2015之前：手工特征+图模型（CRF）
- 2015开始：深度神经网模型
  - 思路：改进CNN，并使用预训练CNN层的参数
  - 传统CNN的问题
    - 后半段网络无空间信息
    - 输入图片尺寸固定
  - 全卷积网络（Fully Convolutional Networks）
    - 所有层都是卷积层
    - 解决降采样后的低分辨率问题

语义分割（Semantic Segmentation）

全卷积网络（Fully Convolutional Networks-FCN）

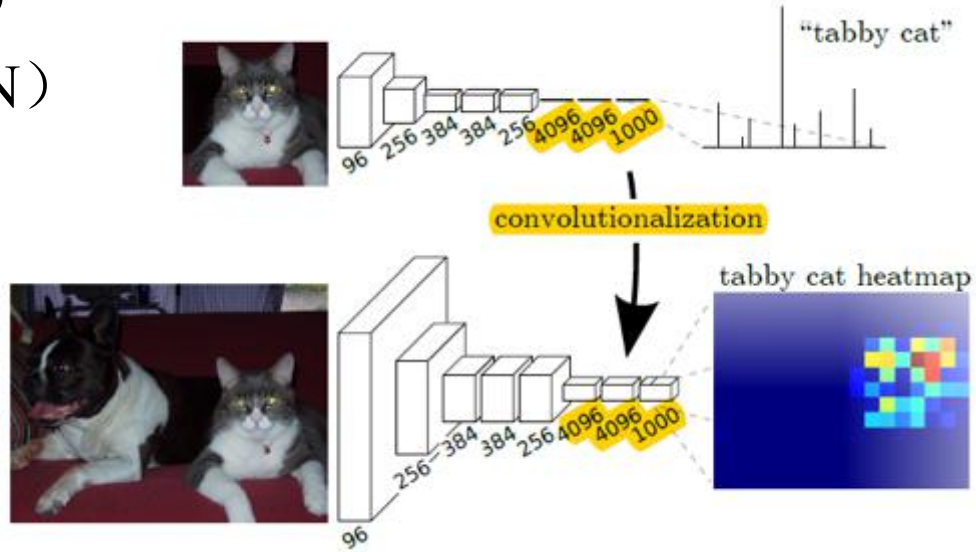
- 全卷积化（Convolutionalization）
  - 将所有全连接层转换成卷积层
  - 适应任意尺寸输入，输出低分辨率分割图片
- 反卷积（Deconvolution）
  - 将低分辨率图片进行上采样，输出同分辨率分割图片
- 跳层结构（Skip-layer）
  - 精化分割图片



语义分割（Semantic Segmentation）

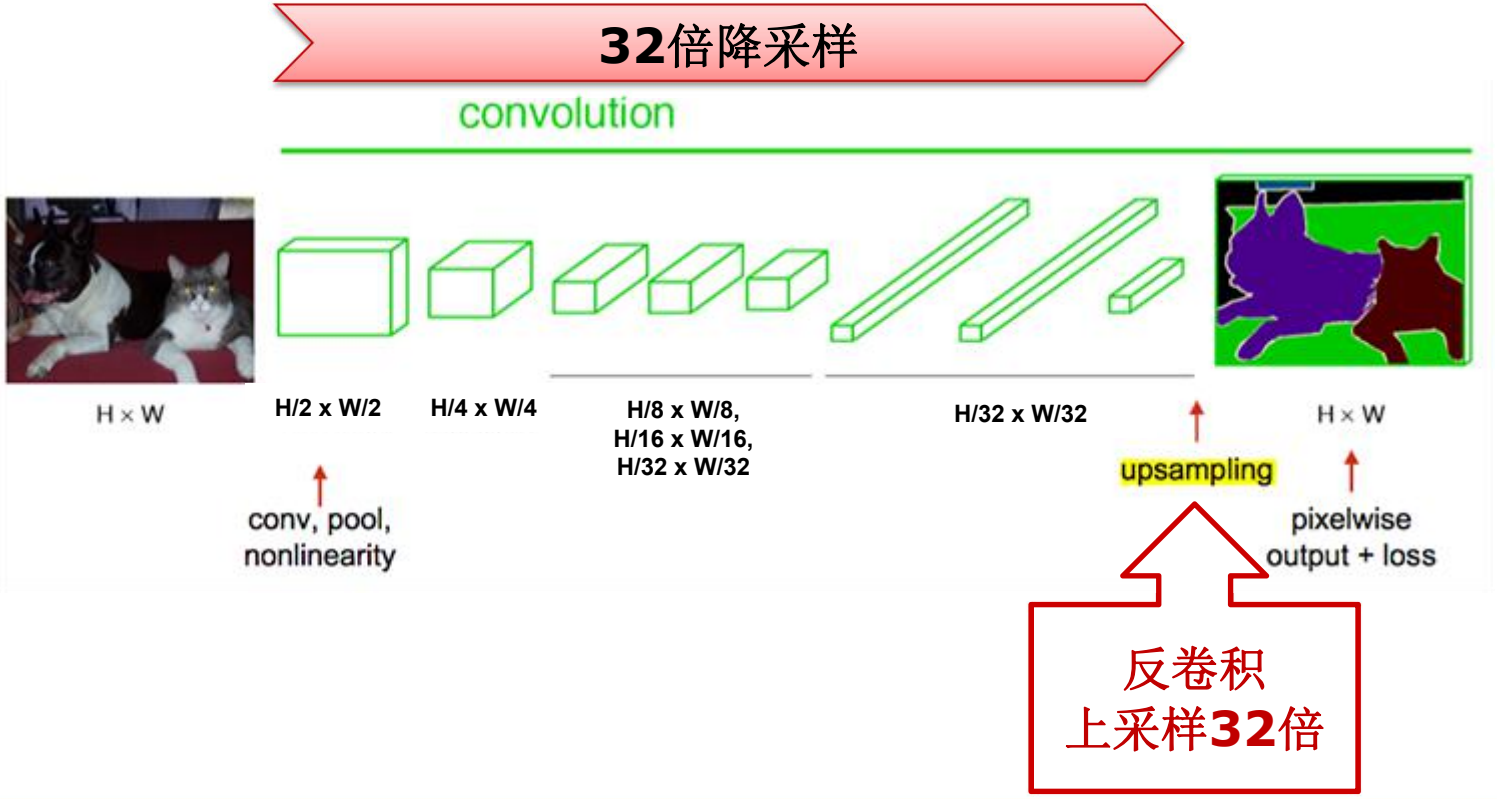
FCN-卷积化（Convolutionalization）

- 基础CNN网络: AlexNet, VGG16, GoogLeNet
- 卷积化后的核尺寸（通道数，宽，高）
  - FC6→（1x1, 4096）
  - FC7→（1x1, 4096）
  - FC8→（1x1, 类别N）
- 分辨率降低32倍
  - 5个卷积层
  - 每层降2倍



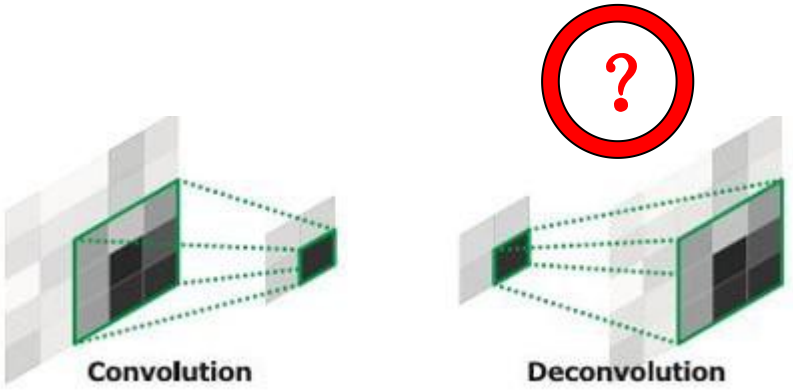
语义分割（Semantic Segmentation）

FCN-卷积化的降维问题



FCN-反卷积

- 卷积的逆操作（？）
  - 小数步长1/f
  - 卷积核尺寸不变

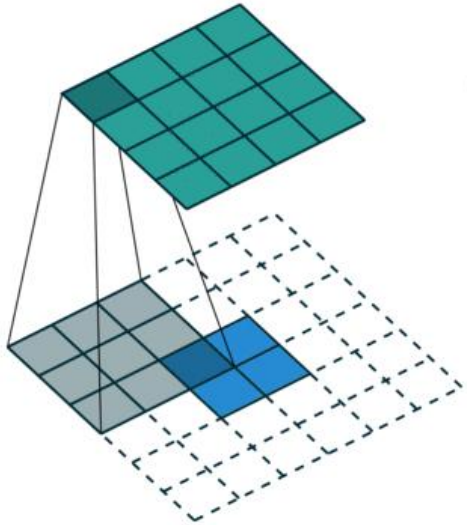


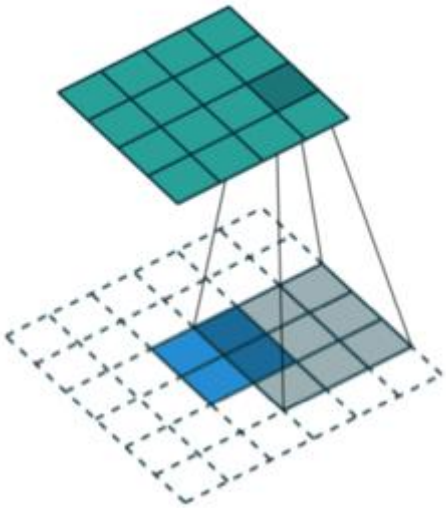
- 前向和后向传播
  - 对应于卷积操作的后向和前向传播，优化上做颠倒
  - 反卷积核是卷积核的转置，学习率为0
- 也叫转置卷积（Transposed convolution）
- 可以拟合出双线性插值

反卷积

反卷积（Deconvolution）

- 外围全补零（Full padding）反卷积
- 输入：2x2
- 输出：4x4
- 参数设置
  - 卷积核尺寸：3x3
  - 步长：1
  - Padding：2
- Skip-layer使用

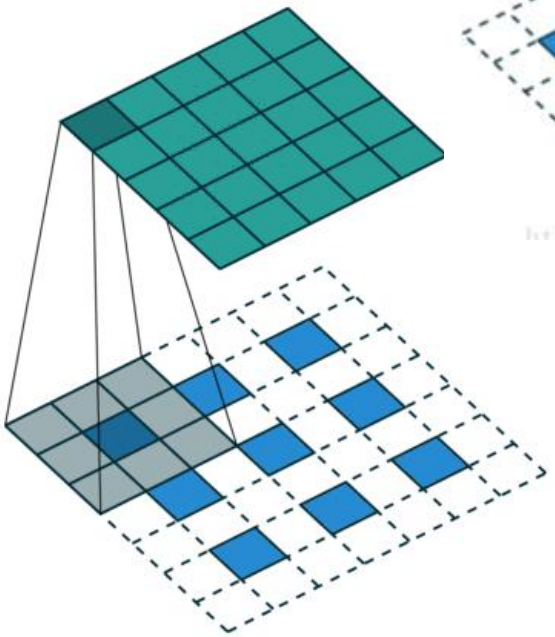


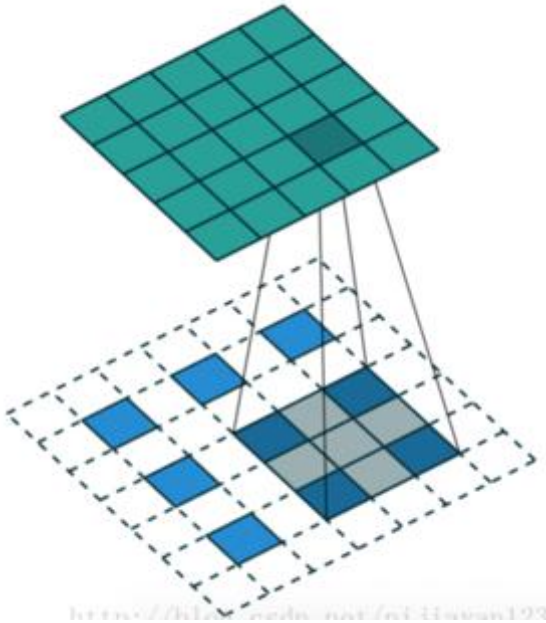


反卷积

FCN-反卷积（Deconvolution）

- 插零分数步长反卷积
- 输入：3x3
- 输出：5x5
- 参数设置
  - 卷积核尺寸：3x3
  - 步长：2
  - Padding：1



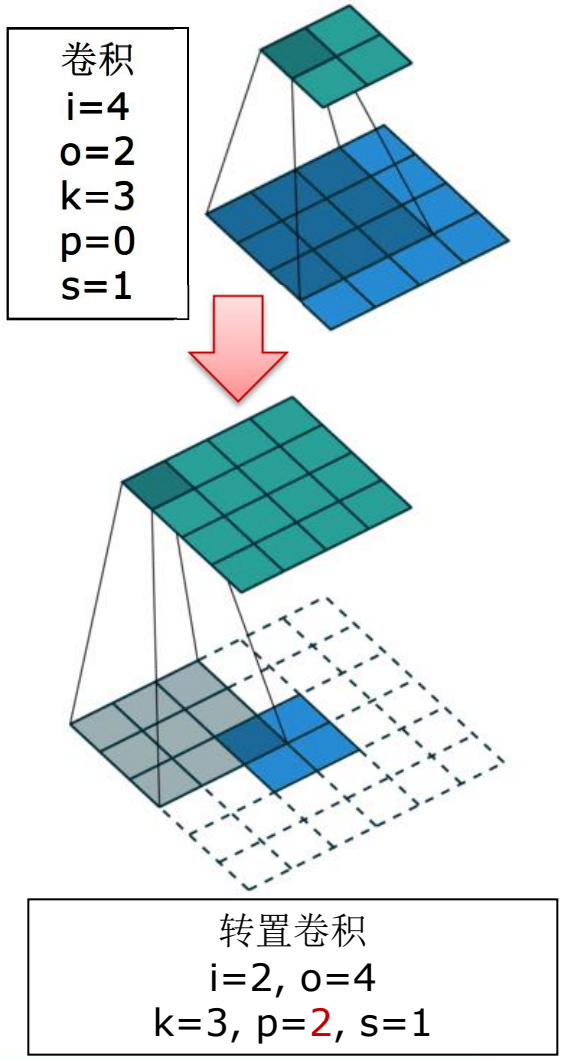


反卷积

FCN-卷积/转置卷积的参数关系

- 步长：1
- padding：0

	卷积	转置卷积
核尺寸	k	k’ = k
步长	s = 1	s’ = s
Padding	p = 0	p’ = k - 1
输入	i	i’ = o
输出	o	o’ = i’ + (k-1)

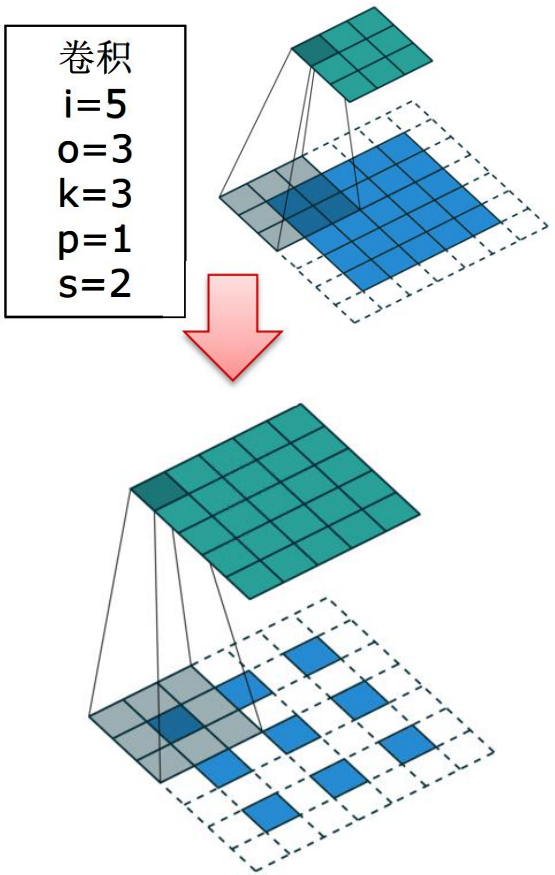


反卷积

FCN-卷积/转置卷积的参数关系

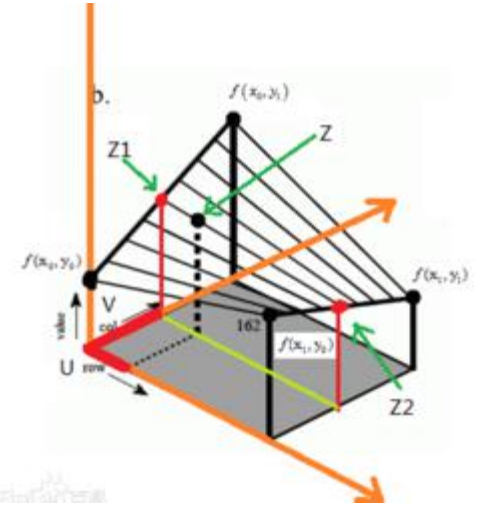
- 步长：> 1
- padding：> 0
- a = i + 2p - k 整除 s

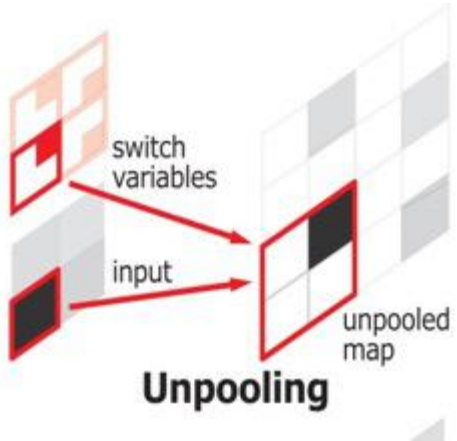
	卷积	转置卷积
核尺寸	k	k' = k
步长	s > 1	s' = 1
Padding	p	p' = k - p - 1
输入	i	i' = o 插零输入：i''=i'+(s-1)(i'-1)
输出	o	o' = s(i'-1) + k - 2p



上采样的三种实现

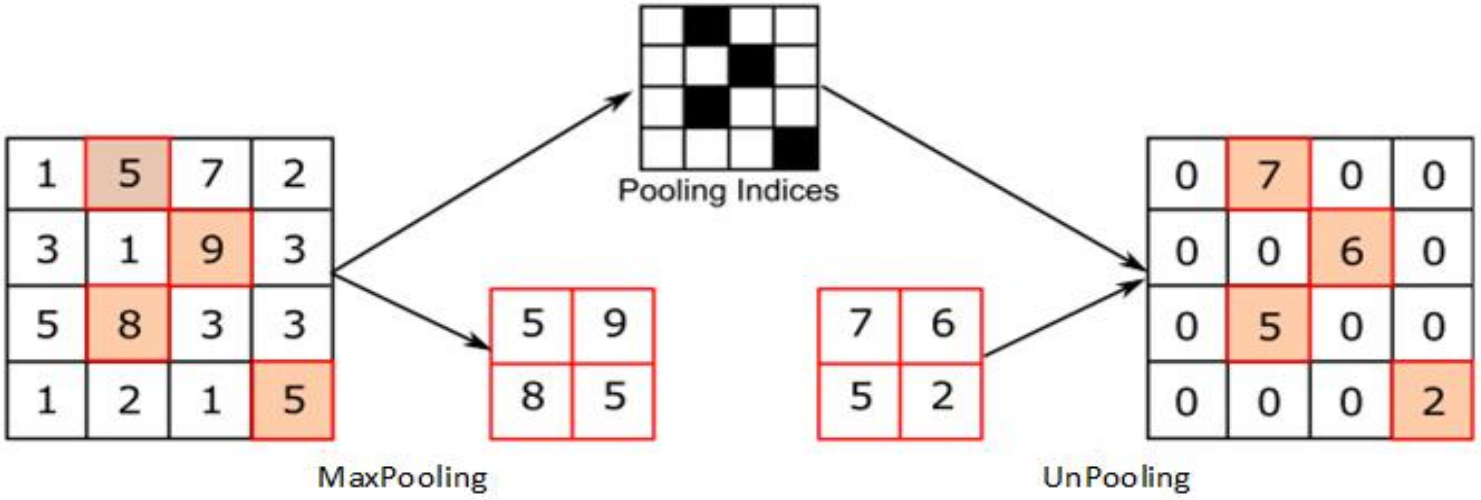
- 双线性插值：
  - 特点是不需要进行学习，运行速度快，操作简单。
- 反卷积：
  - 是为了还原原有特征图，类似消除原有卷积的某种效果，所以叫反卷积
- 反池化：
  - 在池化过程中，记录下池化后元素在对应kernel中的坐标，作为反池化的索引。



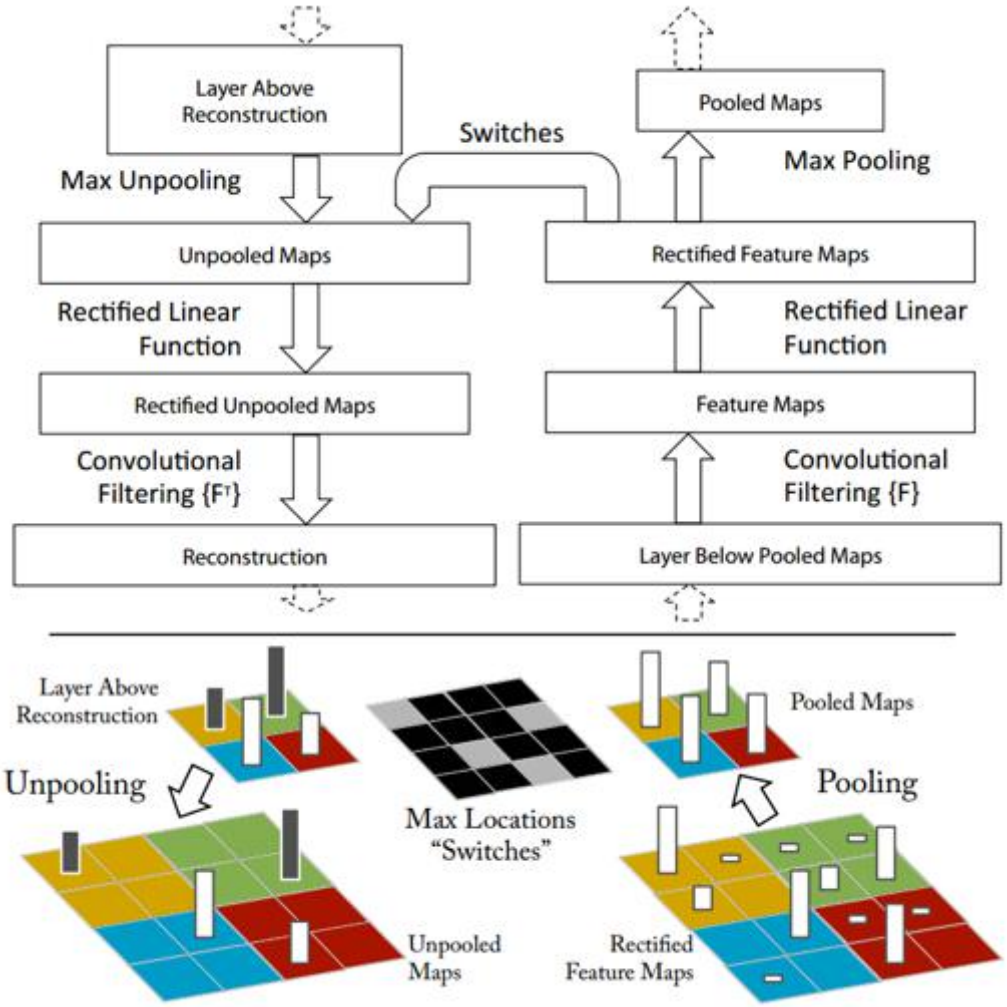


反池化

- 反池化操作（Unpooling）
  - 记录池化时的位置
  - 形成"池化索引"
  - 将输入特征按记录位置摆放回去

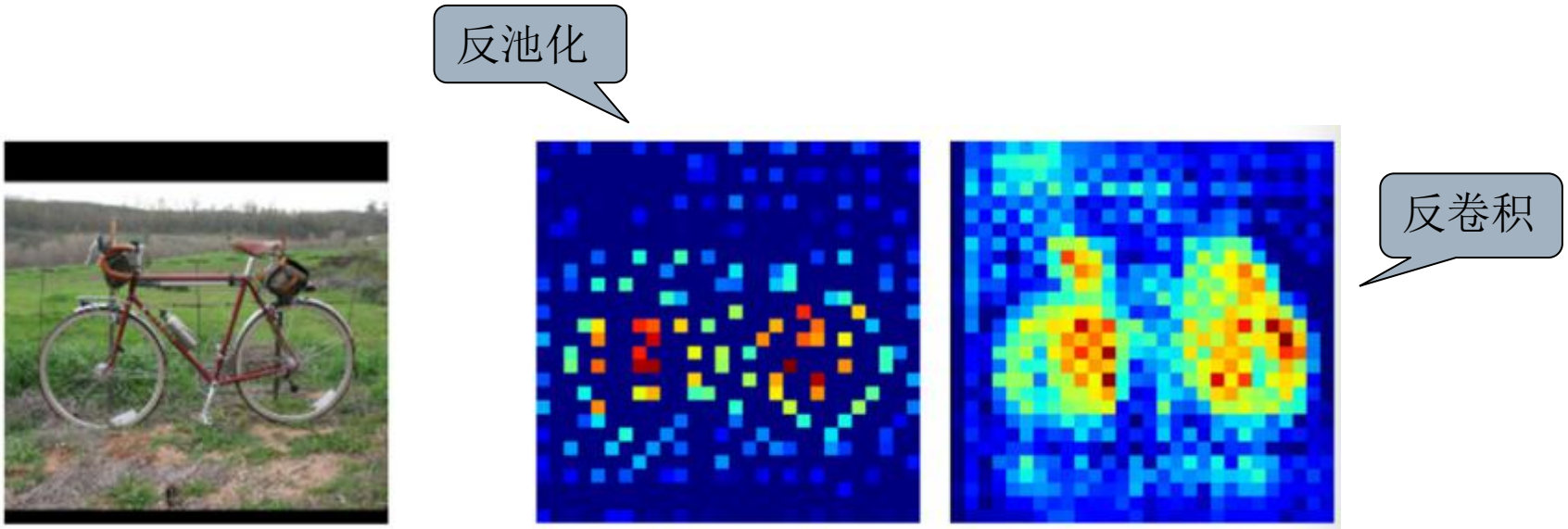


反池化


Layer Above Reconstruction
Max Unpooling
Unpooled Maps
Rectified Linear Function
Rectified Unpooled Maps
Convolutional Filtering {Fᵀ}
Reconstruction
Switches
Pooled Maps
Max Pooling
Rectified Feature Maps
Rectified Linear Function
Feature Maps
Convolutional Filtering {F}
Layer Below Pooled Maps
Layer Above Reconstruction
Unpooling
Unpooled Maps
Max Locations "Switches"
Pooled Maps
Pooling
Rectified Feature Maps

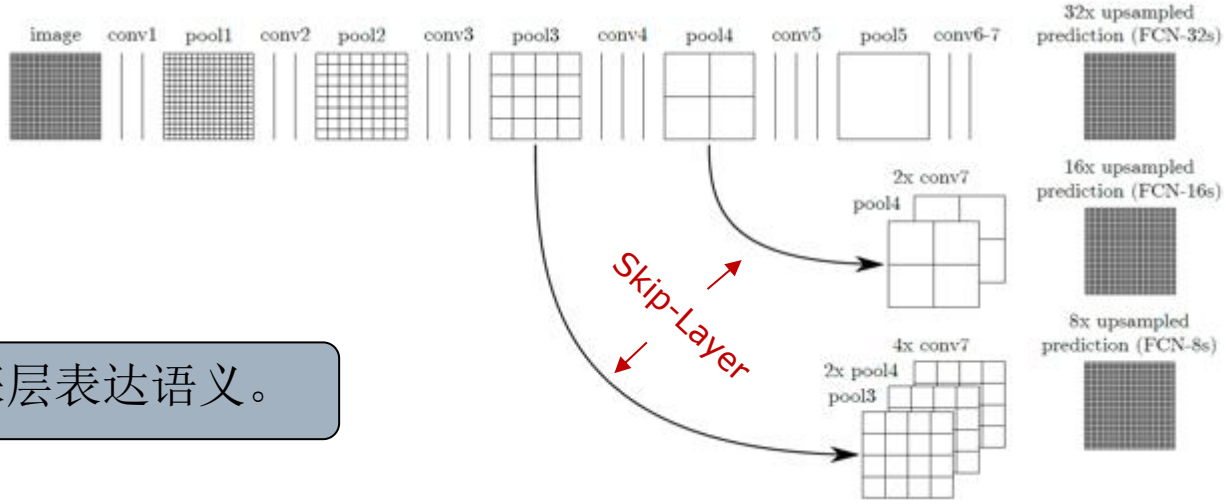
反卷积与反池化

- 反卷积与反池化之间最大的区别在于反卷积过程是有参数要进行学习的。
- 理论上反卷积可以实现反池化(UnPooling)，只要卷积核的参数设置的合理。



FCN-跳层结构（Skip-layer）

- 原因：直接使用32倍反卷积得到的分割结果粗糙
- 使用前2个卷积层的输出做融合
  - 跳层：Pool4和Pool3后会增加一个1x1卷积层做预测
  - 较浅网络的结果精细，较深网络的结果鲁棒

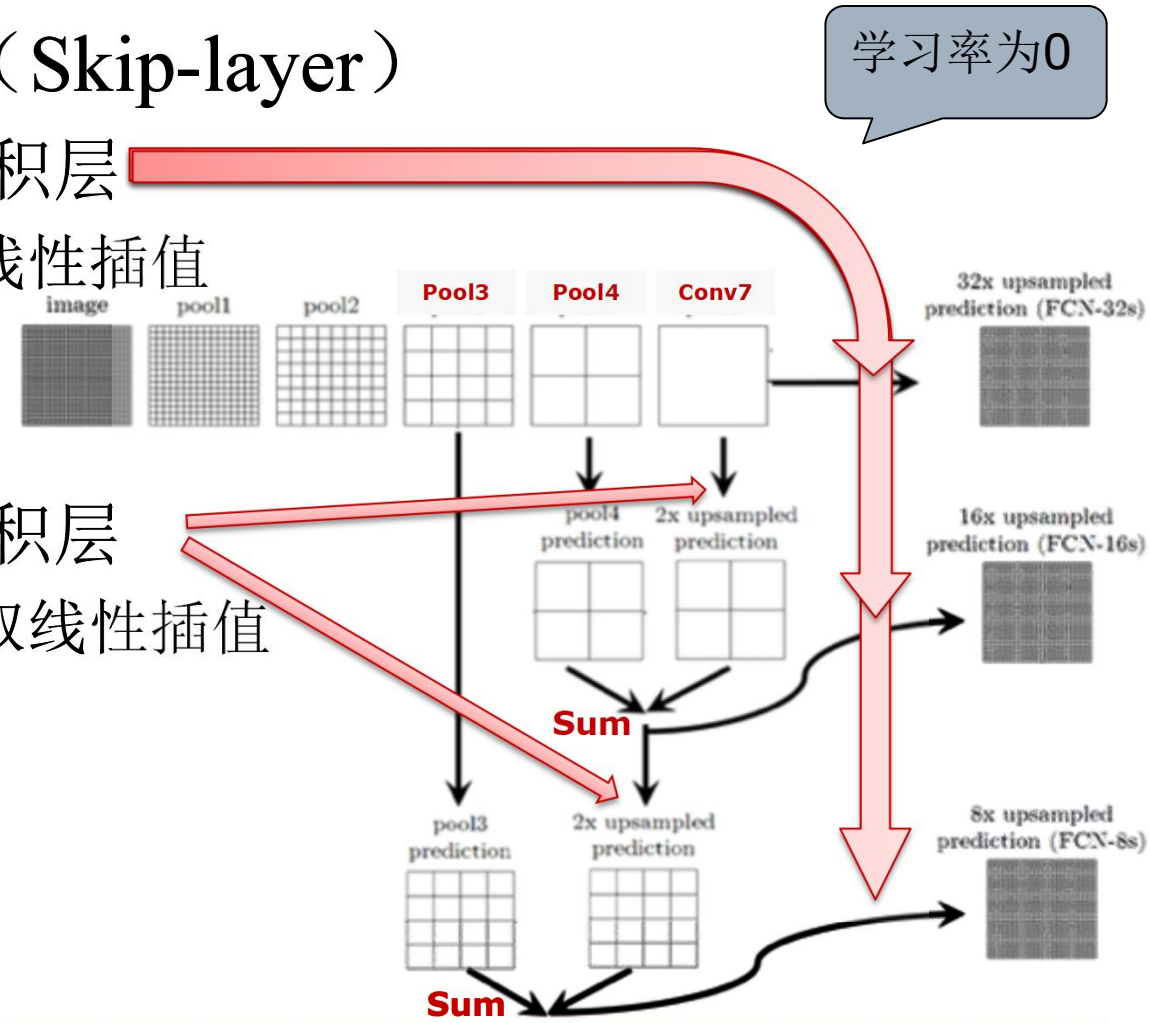


浅层体现细节；深层表达语义。

FCN-跳层结构（Skip-layer）

FCN-跳层结构（Skip-layer）

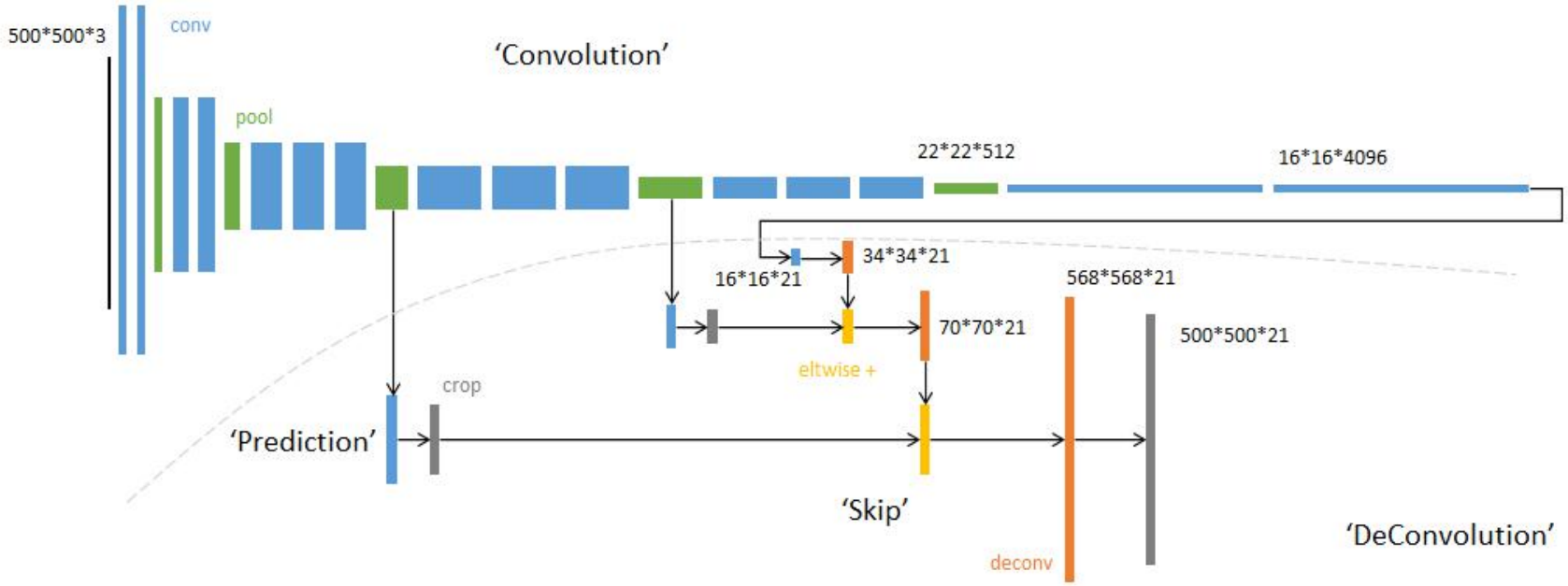
- 最后的反卷积层
  - 固定为双线性插值
  - 不学习
- 中间的反卷积层
  - 初始化为双线性插值
  - 需要学习



构建FCN

FCN构架图例

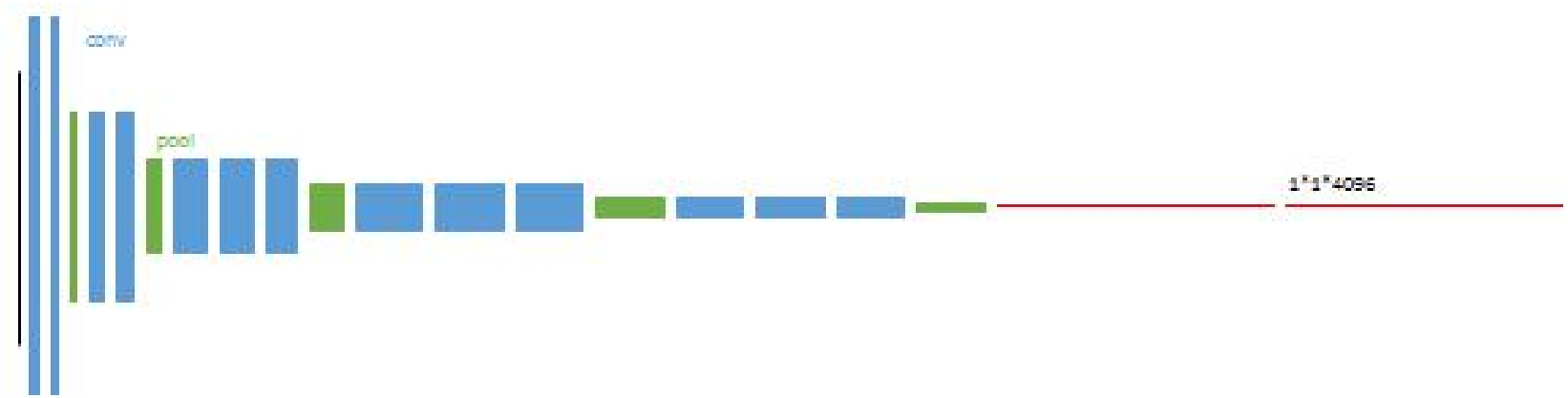
蓝色	绿色	黄色	橙色	灰色
卷积层	池化层	逐像素相加	反卷积层	剪裁层
	获取特征	融合结果	上采样	统一尺寸



构建FCN

使用AlexNet构建FCN

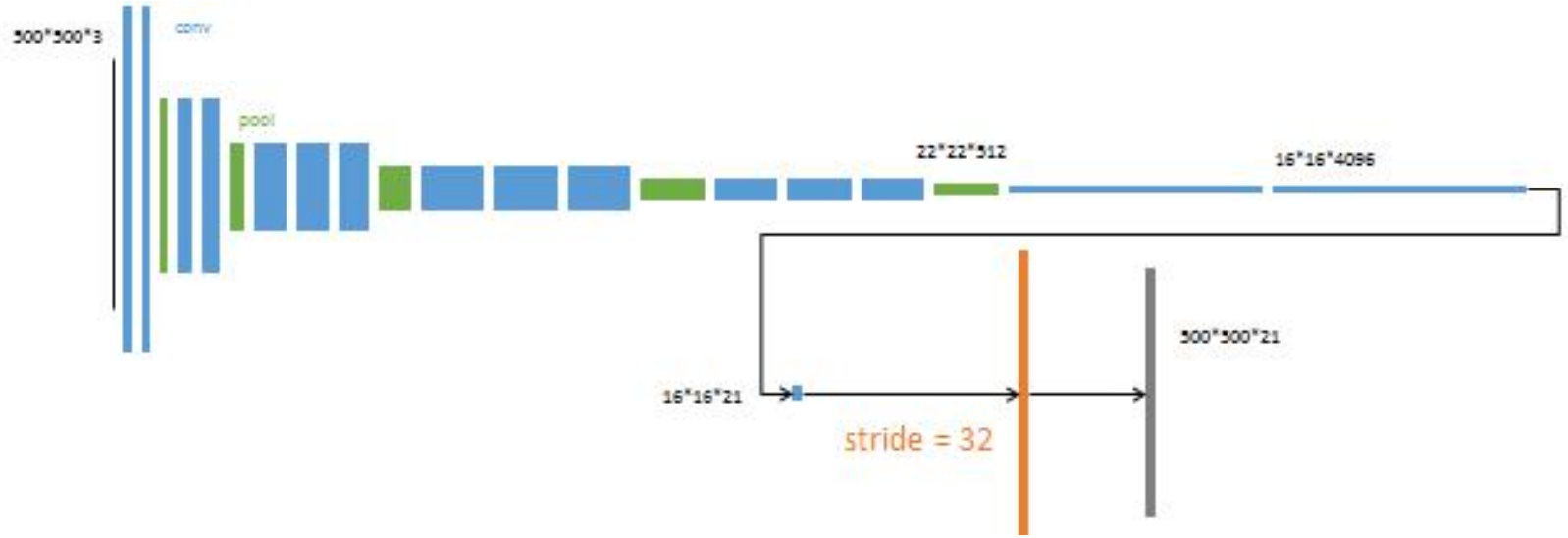
- 第1步
  - 使用AlexNet作为初始网络，保留参数
  - 舍弃全连接层



构建FCN

使用AlexNet构建FCN

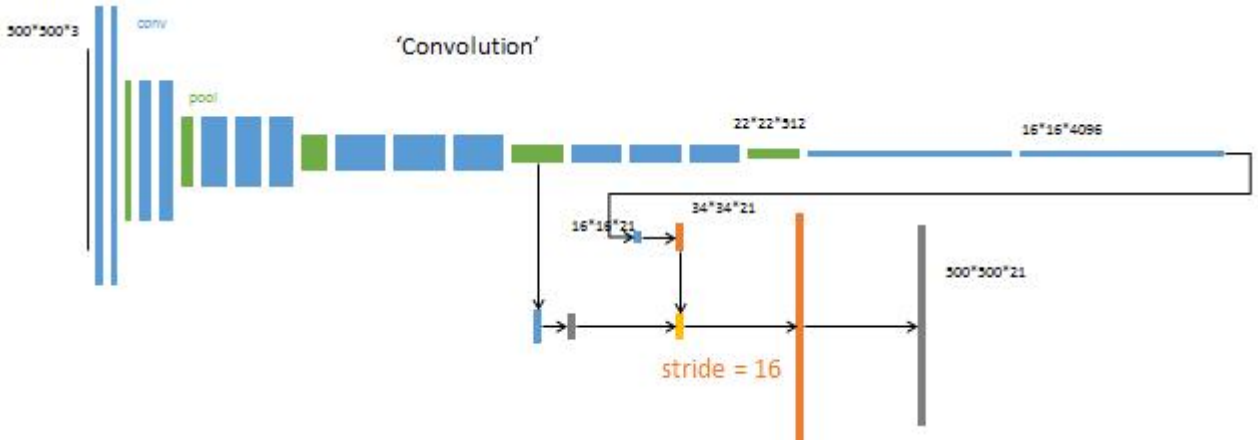
- 第2步（FCN-32s网络）
  - 替换为两个同深度的卷积层（4096,1,1）→16x16x4096
  - 追加一个预测卷积层（21,1,1）→16x16x21
  - 追加一个步长为32的双线性插值反卷积层→500x500x21



构建FCN

使用AlexNet构建FCN

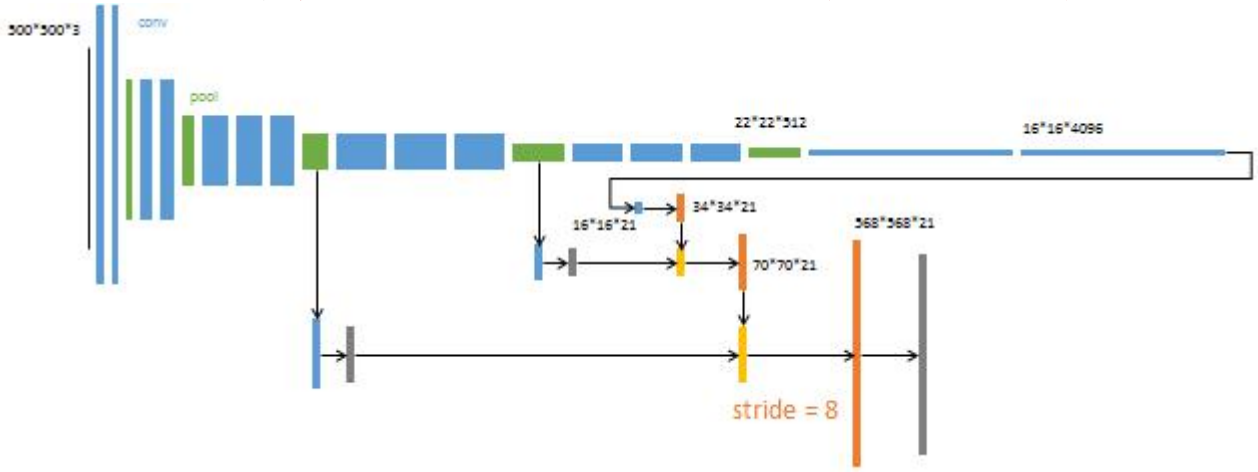
- 第3步（FCN-16s网络）
  - 对最终层Conv7结果2倍上采样→34x34x21
  - 提取Pool4输出，追加预测卷积层（21,1,1）→34x34x21
  - 相加融合→34x34x21
  - 追加一个步长为16的双线性插值反卷积层→500x500x21



构建FCN

使用AlexNet构建FCN

- 第4步（FCN-8s网络）
  - 对上次融合结果2倍上采样→70x70x21
  - 提取Pool3输出，追加预测卷积层（21,1,1）→70x70x21
  - 相加融合→70x70x21
  - 追加一个步长为8的双线性插值反卷积层→500x500x21



FCN训练

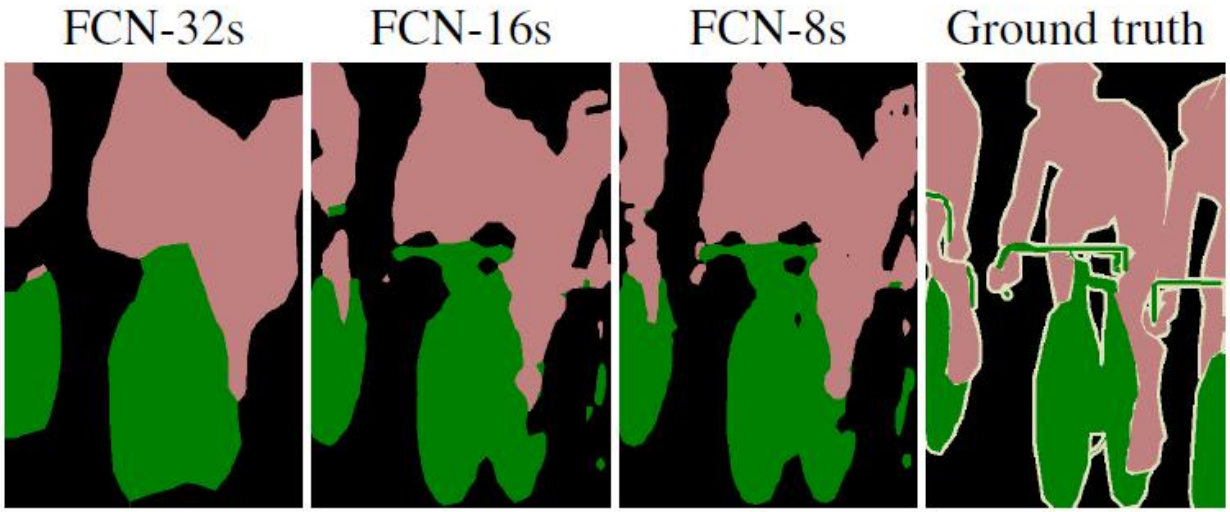
- SGD with momentum (0.9)
  - Learning rate
    - 0.001(AlexNet)，0.0001(VGG16)，0.00001(GoogLeNet)
  - Minibatch：20
- 初始化
  - 卷积层
    - 前5个卷积层使用初始CNN网络的参数
    - 剩余第6和7卷积层初始化为0
  - 反卷积层
    - 最后一层反卷积层固定为双线性插值，不做学习
    - 剩余反卷积层初始化为双线性插值，做学习

FCN的性能

FCN跳层结构的性能

- FCN-8s最优

	pixel acc.	mean acc.	mean IU	f.w. IU
FCN-32s-fixed	83.0	59.7	45.4	72.0
FCN-32s	89.1	73.3	59.4	81.4
FCN-16s	90.0	75.7	62.4	83.0
FCN-8s	**90.3**	**75.9**	**62.7**	**83.2**



FCN的性能

FCN的基础网络性能

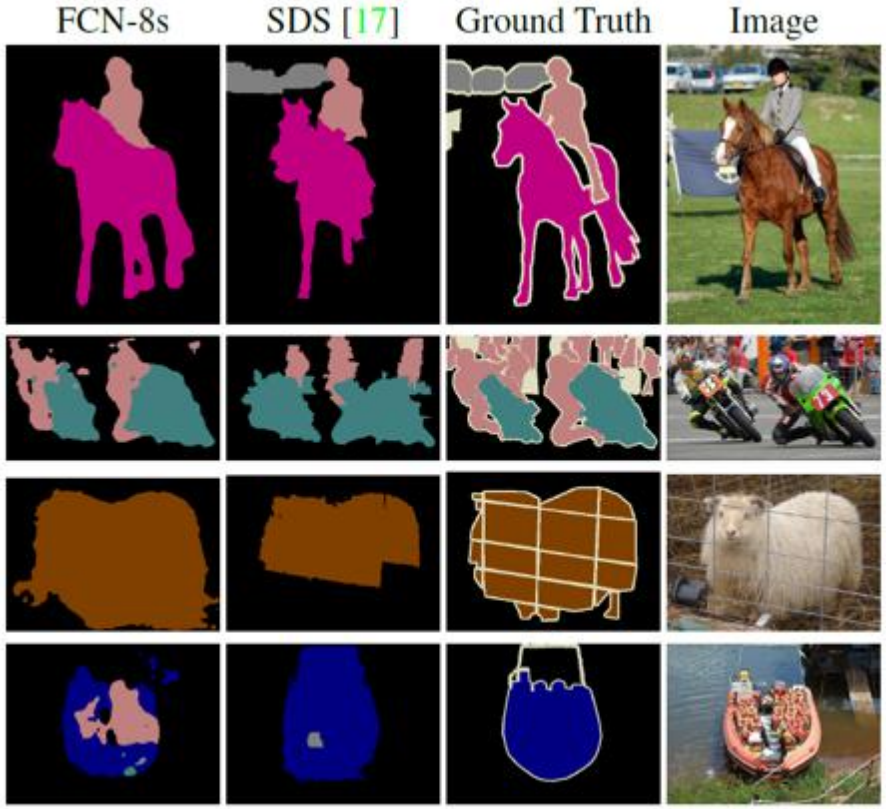
- FCN-VGG16最优

	FCN-AlexNet	FCN-VGG16	FCN-GoogLeNet[4]
mean IU	39.8	**56.0**	42.5
forward time	50 ms	210 ms	59 ms
conv. layers	8	16	22
parameters	57M	134M	6M
rf size	355	404	907
max stride	32	32	32

FCN的性能

FCN-8s的Pascal VOC竞赛结果

- 边缘准确性比较差
  - 第1个卷积层大量补零
  - 之后做裁剪
  - 保证输出分辨率
  - 带来噪声



	mean IU VOC2011 test	mean IU VOC2012 test	inference time
R-CNN [12]	47.9	-	-
SDS [17]	52.6	51.6	~ 50 s
FCN-8s	**62.7**	**62.2**	**~ 175 ms**

DeepLab V1

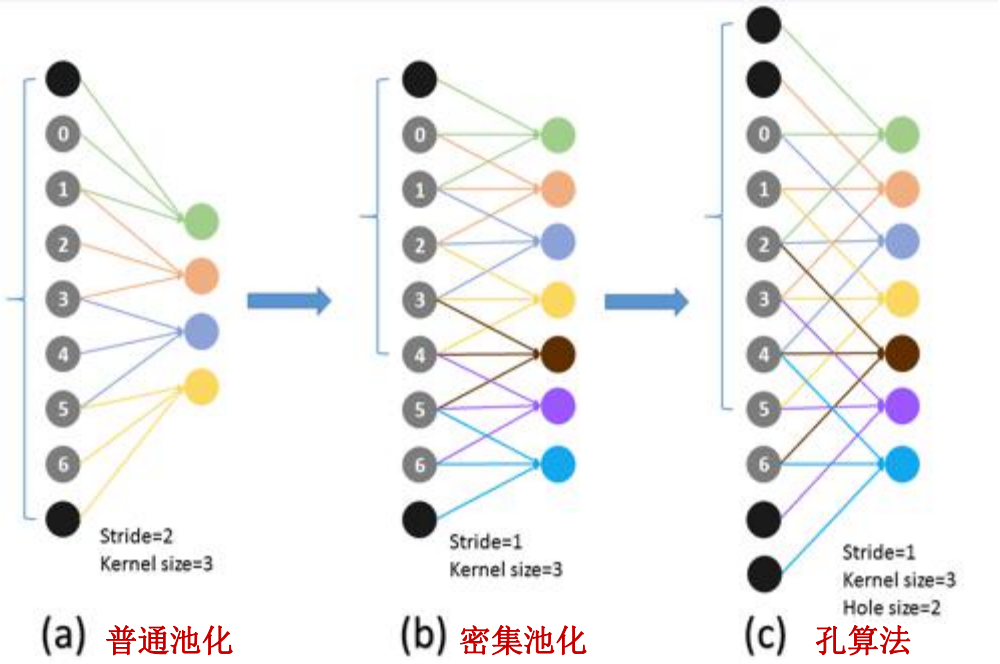
- 基本结构
  - 优化后的CNN+传统的CRF图模型
- 新的上采样卷积方案
  - 带孔（hole）结构的空洞卷积（Atrous/Dilated convolution）
- 边界分割的优化
  - 使用全连接条件随机场CRF进行迭代优化
- 两个步骤
  - CNN输出粗糙的分割结果
  - 全连接CRF精化分割结果

DeepLab V1

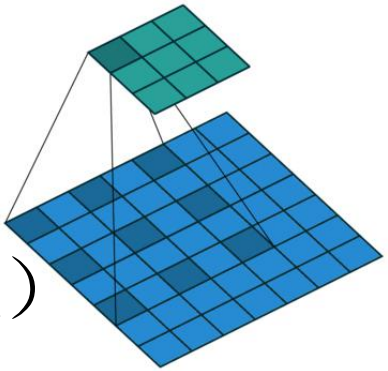
- 孔（Hole）算法
  - 解决原始FCN网络的输出低分辨问题
  - 降低池化层的降采样倍数
    - VGG16网络Pool4和Pool5层的步长：2→1
    - 减小降采样倍数：32→8
    - 后续卷积核的感受野（Field-Of-View）会变小
  - 更改卷积核的结构→加孔（Hole）
    - 无上采样功能
    - 恢复感受野，可以用来fine-tune
    - 保证了网络最终的高分辨输出（仅8倍降采样）

DeepLab V1

- 孔（Hole）算法
  - 卷积核结构
    - 尺寸不变（3x3），元素间距变大（1→2）
    - 步长不变（1）
  - 优势
    - 参数数量不变
    - 计算量不变
    - 高分辨输出
  - 采用层
    - Conv5：孔尺寸2
    - Conv6：孔尺寸4



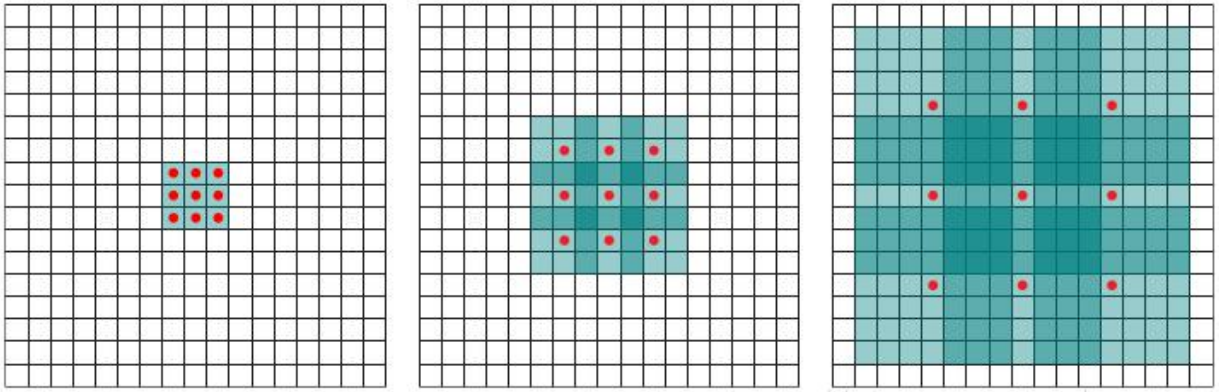
DeepLab V1



- 空洞卷积（Atrous/Dilated convolution）
  - 孔算法的正式名称
  - 与降低池化层步长配对使用，以取代上采样反卷积
  - 孔尺寸→Rate
    - Rate越大，感受野越大

插入rate-1个0，周边也是

卷积核尺寸3x3



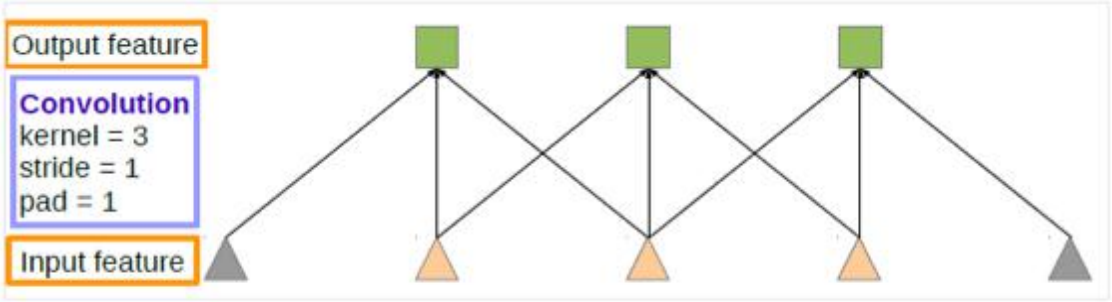
Rate = 1
无插零

Rate = 2
插1个零

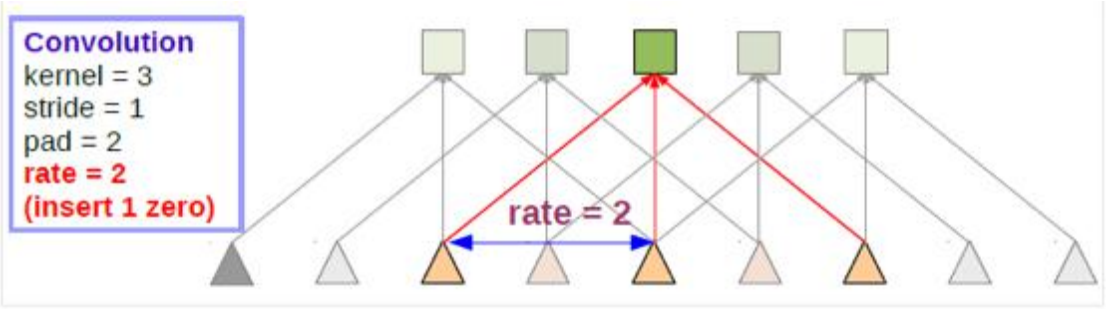
Rate = 4
插3个零

DeepLab-DCNN

- 空洞卷积的参数：



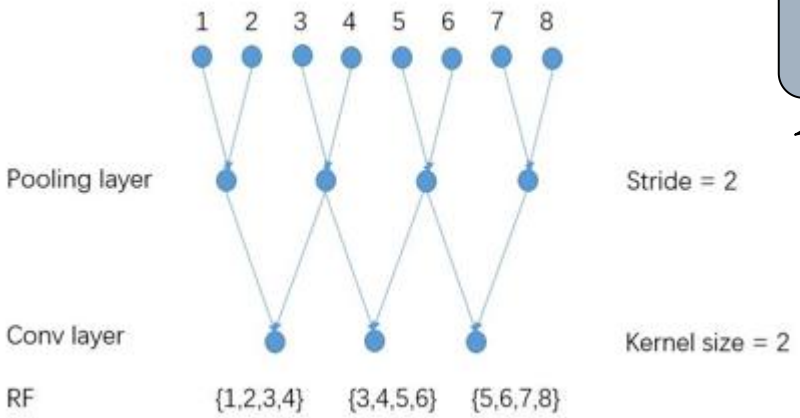
(a) Sparse feature extraction



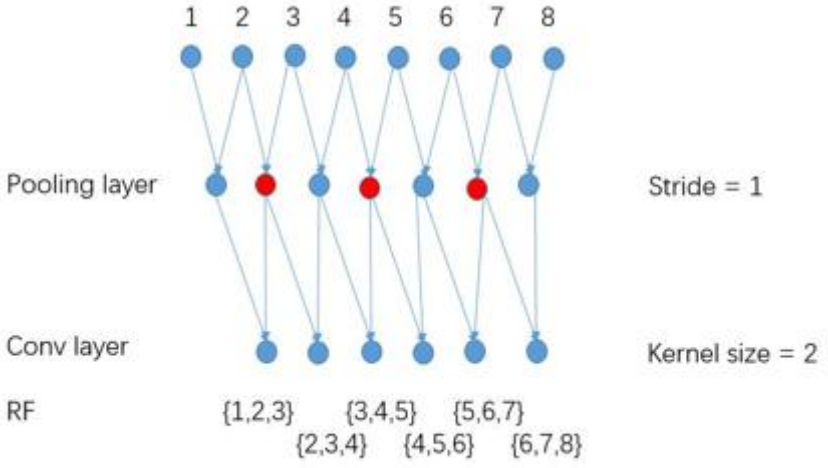
(b) Dense feature extraction

DeepLab V1

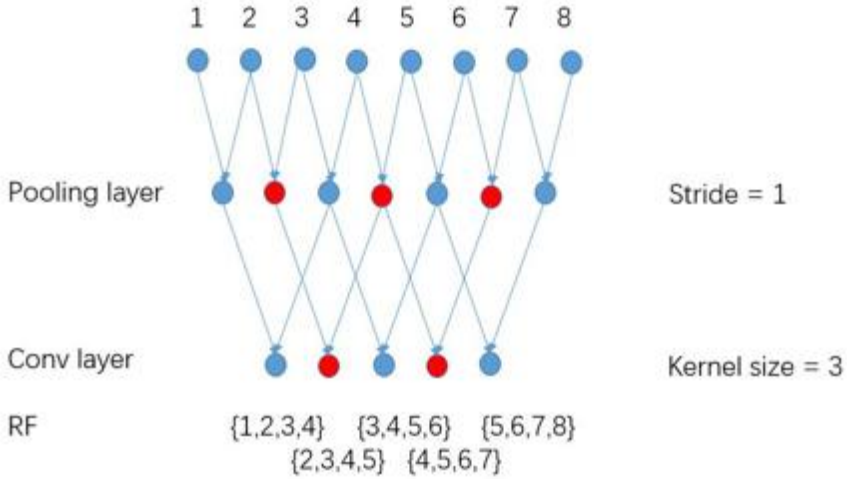
传统的池化+卷积：
feature map变稀疏



感受野较小

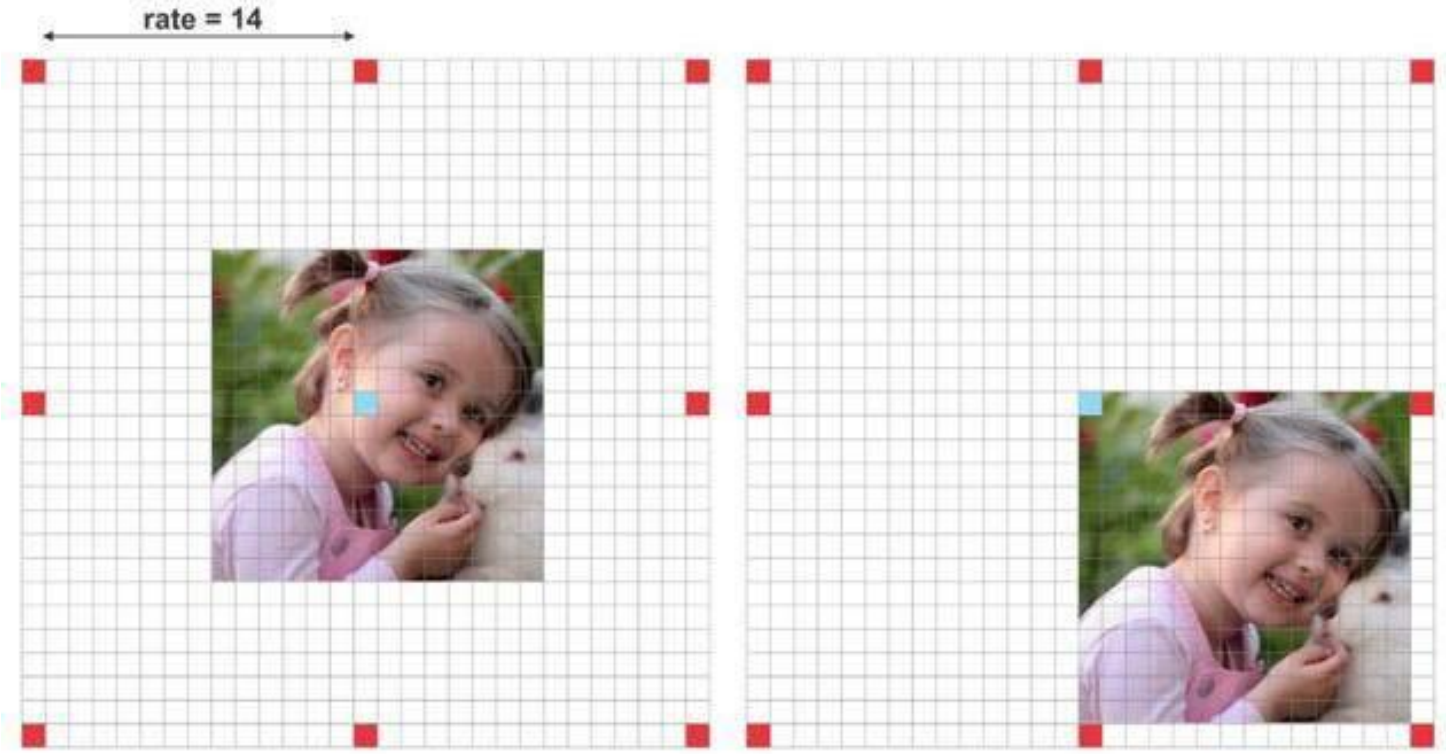


不增加计算量的条件下：扩大了感受野



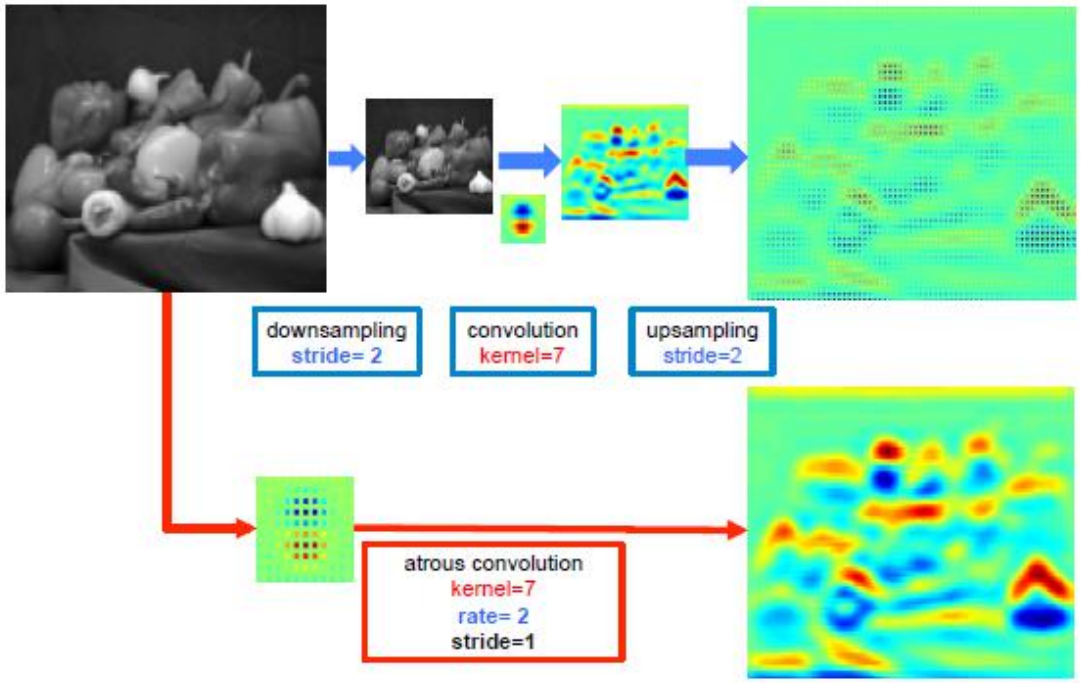
DeepLab V1

- 显然，扩张率过大也不合适。给小图设置大的扩张率的副作用：对于 14×14 的输入图像，使用扩张率为 15 的 3×3 卷积，其结果和常规的 1×1 卷积类似。



DeepLab V1

- 空洞卷积效果
  - 稀疏特征提取：x2降采样→7x7卷积→x2上采样
  - 稠密特征提取：7x7空洞卷积

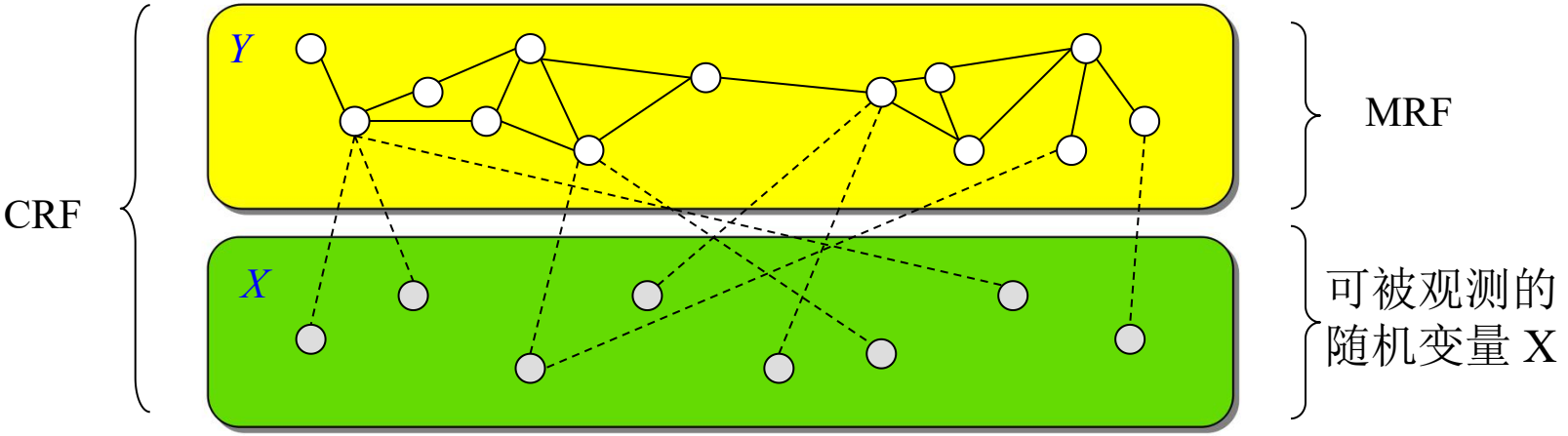


DeepLab V1

- 图像在CNN中逐层传递，是一个逐步抽象的过程，原来的位置信息会随着深度而减少甚至消失。
- CRF(条件随机场)的作用是，能在决定一个位置的像素值时，会考虑周围邻居的像素值（label）。
- short range的CRF可用于消除噪音，但是通过CNN得到的feature map在一定程度上已经足够平滑了。
- 因此采用fully connected CRF，这样考虑的就是全局的信息了。

条件随机场（CRF）

- 随机场可以看成是一组"对应同一个样本空间的"随机变量的集合。当给每一个位置按照某种分布随机赋予一个值之后，其全体就叫做随机场。
- 马尔科夫随机场（MRF）对应一个无向图。此无向图上的每一个节点对应一个随机变量，节点之间的边表示节点对应的随机变量之间有概率依赖关系。



条件随机场（CRF）

- 有向图模型中，每一个随机变量都有自己的条件概率分布（CPD）；无向图模型中类似CPD的“关系”称作Factor ——“亲密关系（affinities）”。
- 求解无向图的联合概率方法：那把所有的Factor像有向图模型的贝叶斯网络那样都连乘起来，再做归一化。

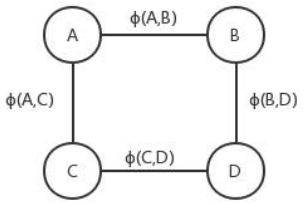
$$
\begin{aligned}
&P(A=1,B=1,C=1,D=1)\\
&=\frac{1}{Z}\phi(A=1,B=1)\times\phi(A=1,C=1)\times\phi(B=1,D=1)\times\phi(C=1,D=1)\\
&=\frac{1}{Z}\times 60\times 25\times 60\times 150\\
&=\frac{1}{Z}\times 60\times 25\times 60\times 150\\
&=\frac{1}{Z}\times 13500000\\
&=0.1128
\end{aligned}
$$

ϕ(A,B)

A=0	B=0	10
A=0	B=1	70
A=1	B=0	1
A=1	B=1	60

ϕ(A,C)

A=0	C=0	300
A=0	C=1	1
A=1	C=0	5
A=1	C=1	25



ϕ(B,D)

B=0	D=0	10
B=0	D=1	70
B=1	D=0	1
B=1	D=1	60

ϕ(C,D)

C=0	D=0	20
C=0	D=1	70
C=1	D=0	1
C=1	D=1	150

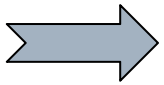
条件随机场（CRF）

- 这些联合概率实际上代表了无向图模型的概率分布，称为Gibbs分布 。
- Gibbs分布就是利用Factor表示的无向图模型的概率分布，Gibbs分布的表示形式如下所示：

$$P(X_1, X_2, \ldots X_n) = \frac{1}{Z(X)} P(X_1, \tilde{X_2}, \ldots X_n)$$

$$P(X_1, \tilde{X_2}, \ldots X_n) = \prod_{i=k}^{m} \phi_i(X)$$

$$Z(X) = \sum \prod \phi_i(X)$$



CRF

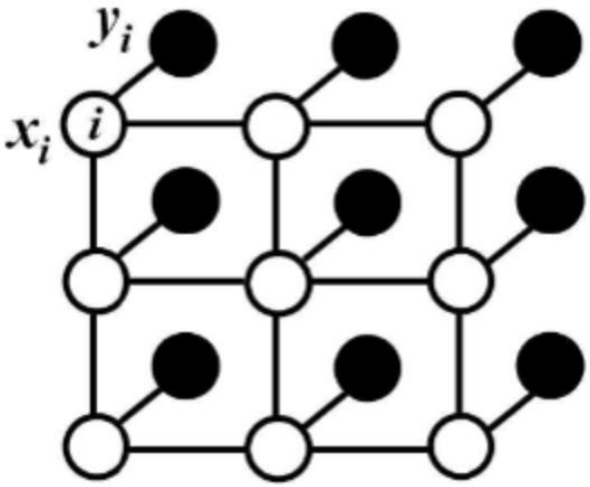
$$P(Y|X) = \frac{1}{Z(X)} \tilde{P}(Y, X)$$

$$\tilde{P}(Y, X) = exp(\sum_i w_i * f_i(Y, X))$$

$$Z(X) = \sum_Y exp(\sum_i w_i * f_i(Y, X))$$

全连接条件随机场(FC-CRF)

- 对于每个像素i具有类别标签x_i还有对应的观测值y_i，这样每个像素点作为节点，像素与像素间的关系作为边，即构成了一个条件随机场。
- 而且我们通过观测变量y_i来推测像素i对应的类别标签x_i。条件随机场如下图：



$$P(X = x \mid I) = \frac{1}{Z(I)} \exp(-E(x \mid I))$$

在deepLab中，整个模型的能量函数

$$E(x) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j).$$

条件随机场（FC-CRF）

- x是对全局pixels的概率预测分布，x_i是其中一个pixel的概率预测分布，
- ψ_u是一元势函数，这里是$\psi_u(x_i) = -\log P(x_i)$；
- 二元势函数为：

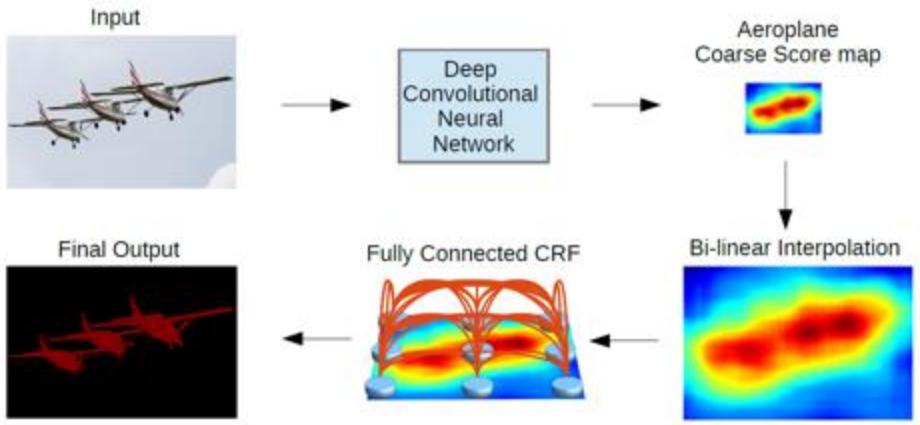
$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \underbrace{\sum_{m=1}^{K} w^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j)}_{k(\mathbf{f}_i, \mathbf{f}_j)}$$

$$k(\mathbf{f}_i, \mathbf{f}_j) = w^{(1)} \underbrace{\exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\right)}_{\text{appearance kernel}} + w^{(2)} \underbrace{\exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}\right)}_{\text{smoothness kernel}}$$

- 其中，K为Kernel数量，w为权重，在这里采用高斯核，并且任意两个像素点都有此项，故称为fully connected CRFs.

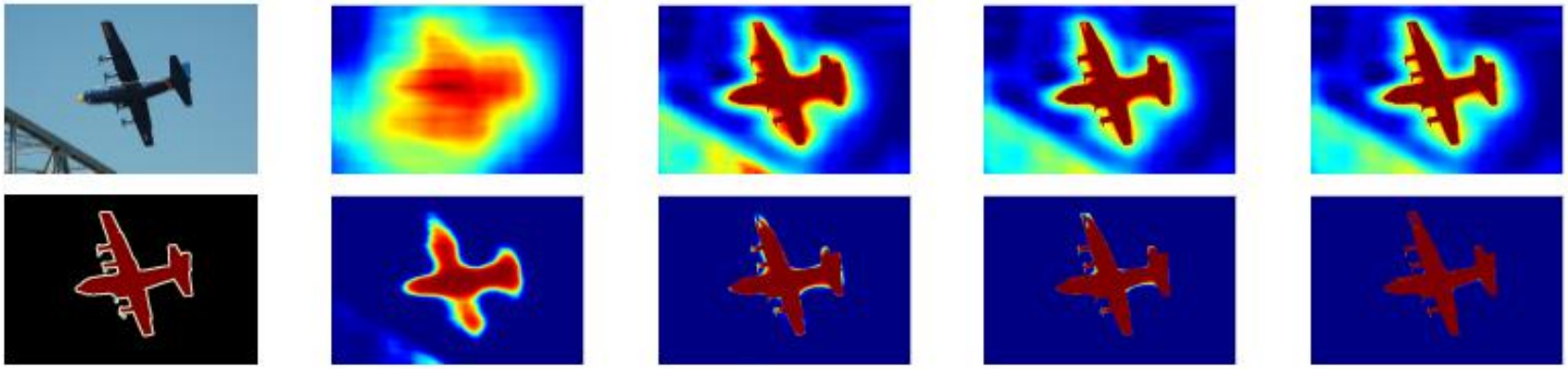
DeepLab V1

- 作用：通过迭代精化分割结果（恢复精确边界）
- 输入
  - 首次：FCN网络输出结果的8倍双线性插值
  - 非首次：上一轮迭代结果
- 能量计算基于图片RGB像素值



DeepLab V1

- CRF作用：通过迭代精化分割结果（恢复精确边界）
- 第一行：飞机类别的分值（softmax之前）
- 第二行：飞机类别的概率值（softmax之后）



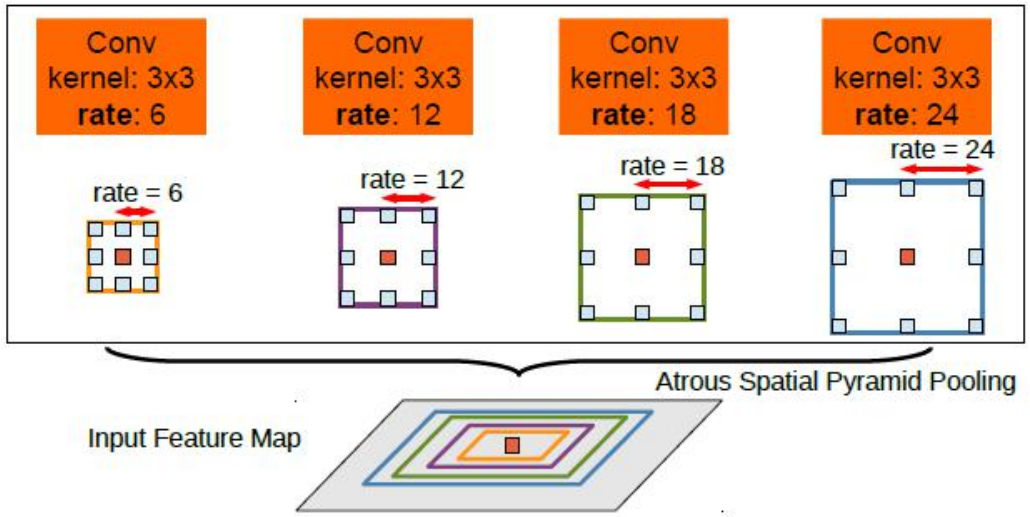
DeepLab v2

- 我们知道"多尺度"技术对性能提升很大，如果有多个感受野，就相当于一种"多尺度"。

- DeepLab v2有两个基础网络结构：
  - 一个是基于vgg16的
  - 另外一个是基于resnet101的
- DeepLab v2通过多尺度对图片进行表达
  - Atrous空间金字塔池化（Atrous Spatial Pyramid Pooling）

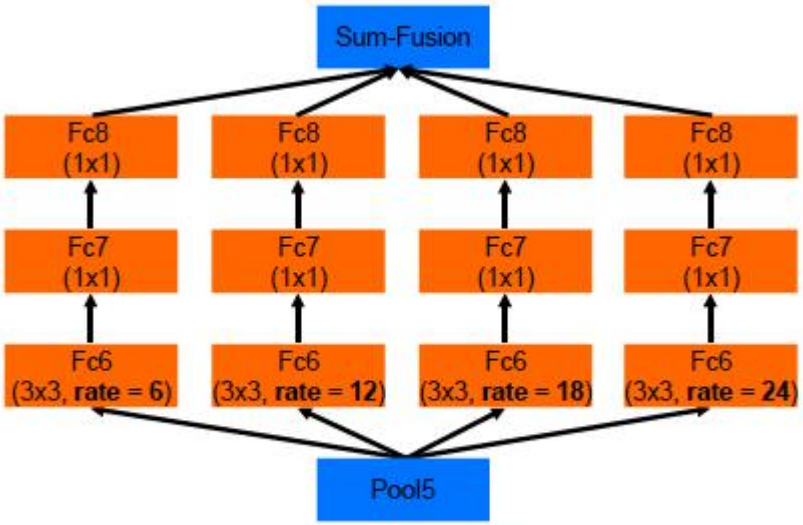
DeepLab v2

- Atrous空间金字塔池化
  - 不同感受野（rate）捕捉不同尺度上的特征
  - 在Conv6层引入4个并行空洞卷积
    - Rate: 6, 12, 18, 24



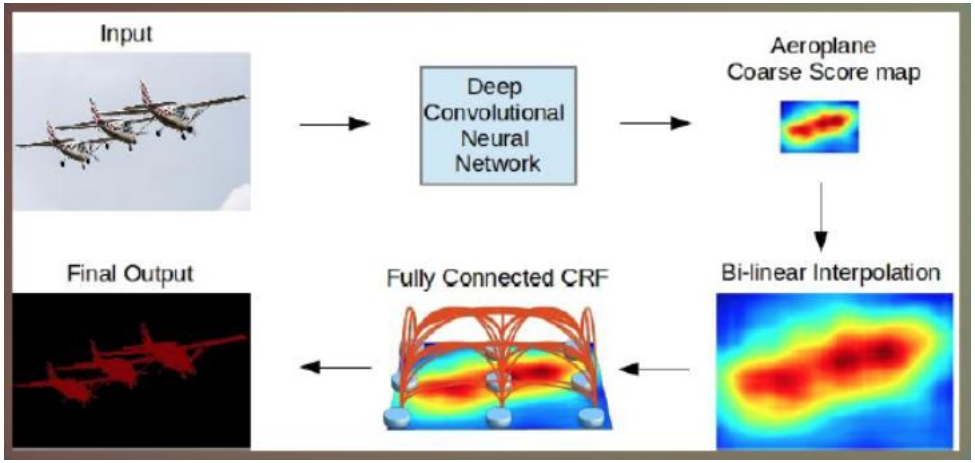
DeepLab v2

- Atrous空间金字塔池化
  - 4个并行膨胀卷积
    - 感受野：13x13, 25x25, 37x37, 49x49
  - Fc6→Fc7→Fc8
    - 深度：4096→2014→类别数量
    - 卷积核：3x3→1x1→1x1
  - 融合：概率相加

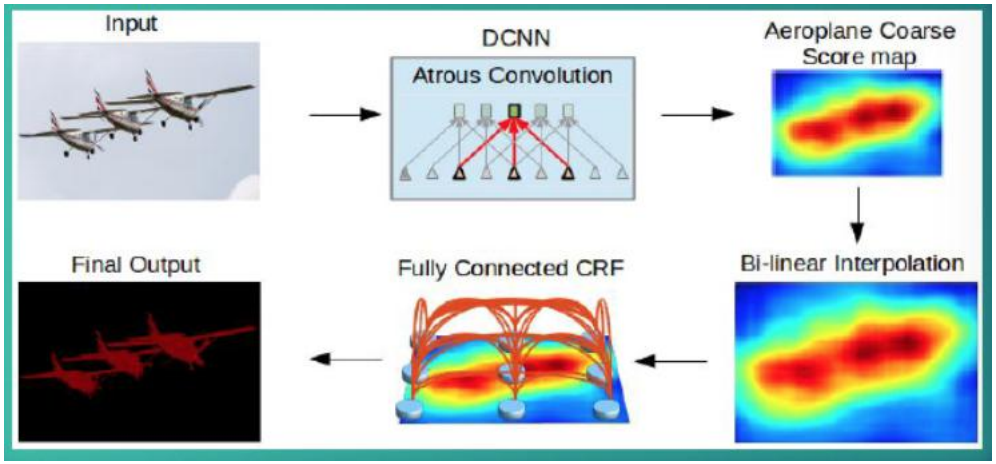


(b) DeepLab-ASPP

DeepLab v2

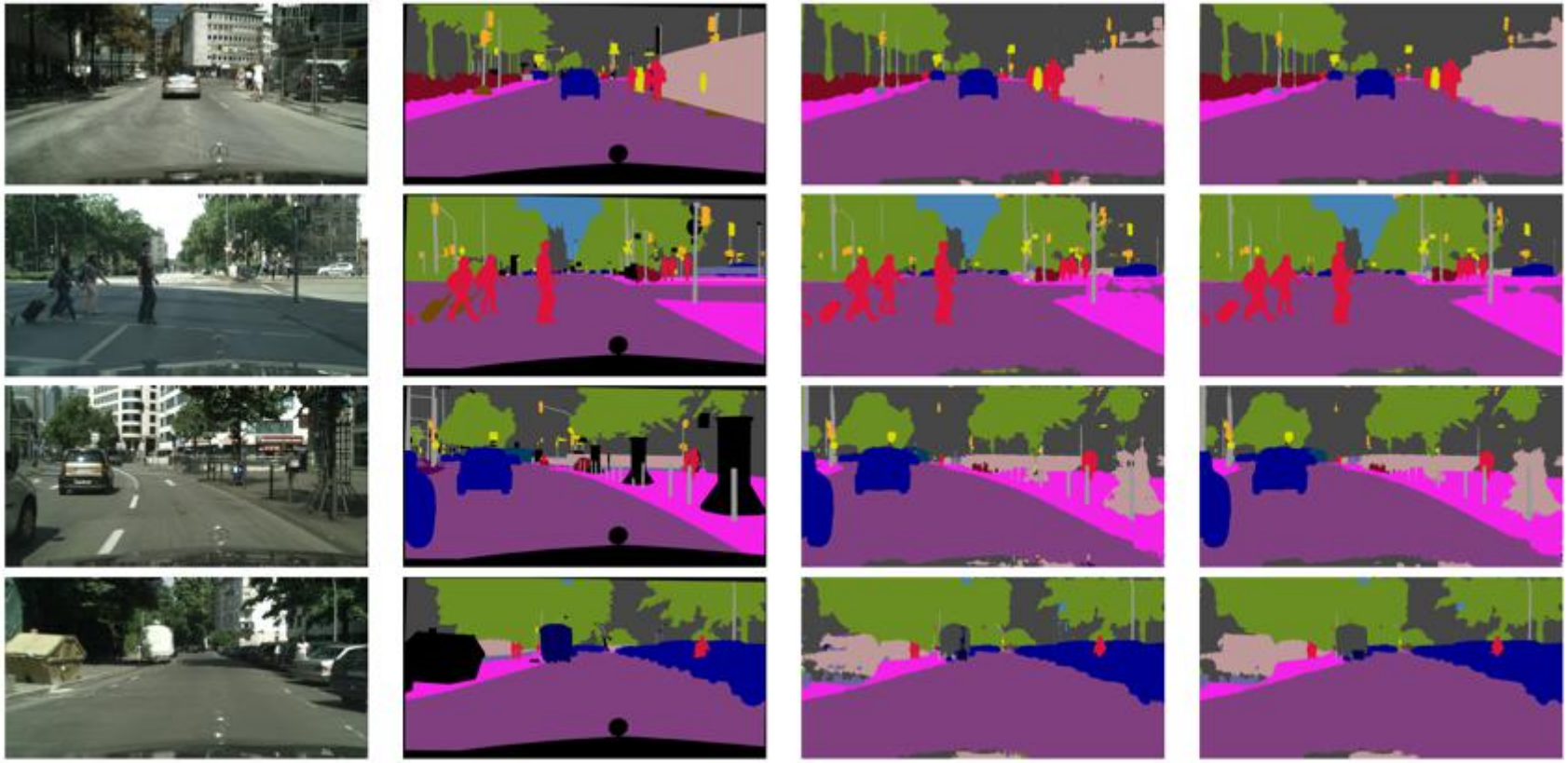


- 按说ASPP架构才是DeepLab V2的特点
- 但人家官网上的图就是这么画的……



DeepLab v2的效果

Cityscapes数据集分割效果



(a) Image (b) G.T. (c) Before CRF (d) After CRF

DeepLab v2的效果

Cityscapes数据集性能

- ResNet-101 优于 VGG16

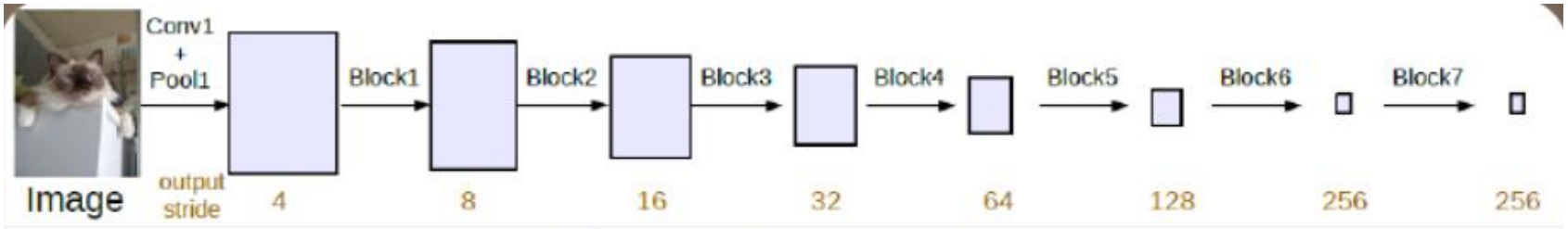
Full	Aug	LargeFOV	ASPP	CRF	mIOU
VGG-16					
		✓			62.97
		✓		✓	64.18
✓		✓			64.89
✓		✓		✓	65.94
ResNet-101					
✓					66.6
✓		✓			69.2
✓			✓		70.4
✓	✓		✓		71.0
✓	✓		✓	✓	71.4

DeepLab v3

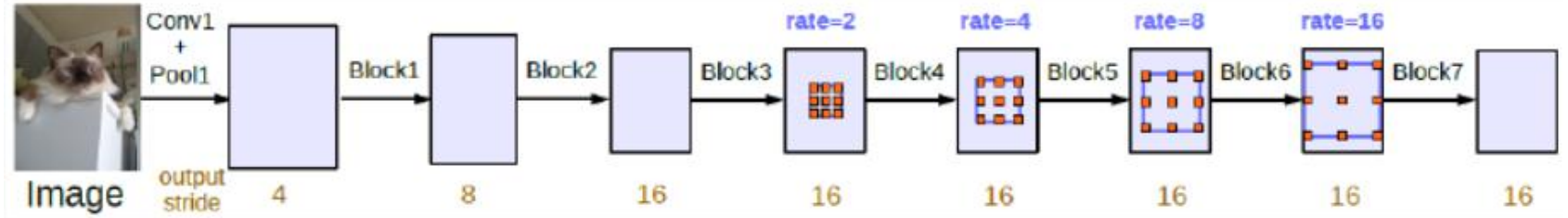
- 提出了更通用的框架，适用于任何网络
- 复制了ResNet最后的block，并级联起来
  - block5-7是block4的副本
- 每个block中包含3个卷积层
- 在ASPP中使用BN层
- 最后一个特征图，采用全局平均池化
- **没有**使用CRF

DeepLab v3

- DeepLab V3 采用了多种空洞率(atrous rate)的空洞卷积级联和并联的方式有效的提取多尺度的语境信息。



(a) Going deeper without atrous convolution.

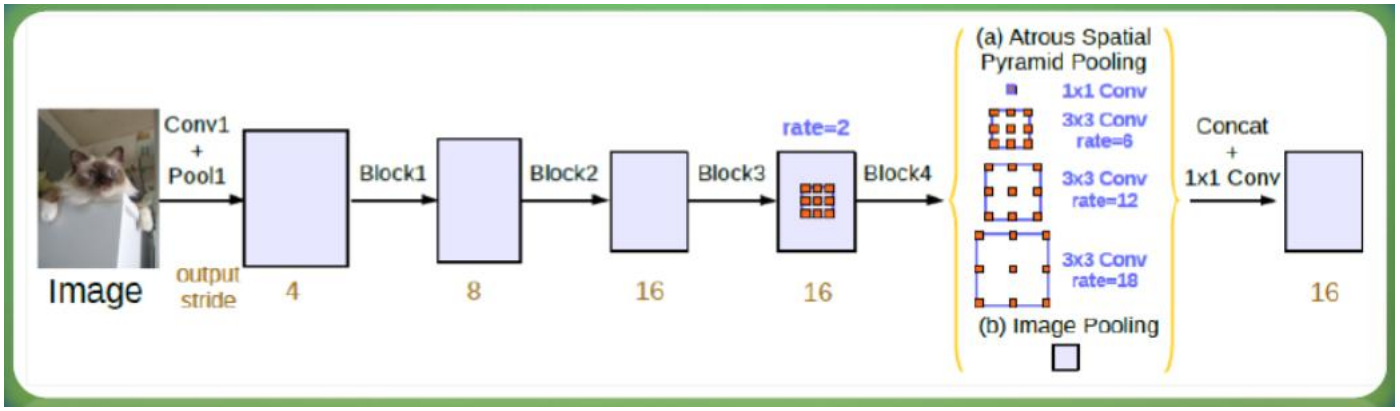


(b) Going deeper with atrous convolution. Atrous convolution with $rate > 1$ is applied after block3 when $output_stride = 16$.

Figure 3. Cascaded modules without and with atrous convolution.

DeepLab v3

- 并联的方式
- 新的ASPP模块包括：
  - 一个1×1卷积和3个3×3的空洞卷积(采样率为(6,12,18))，每个卷积核都有256个且都有BN层
  - 包含图像级特征(即全局平均池化)



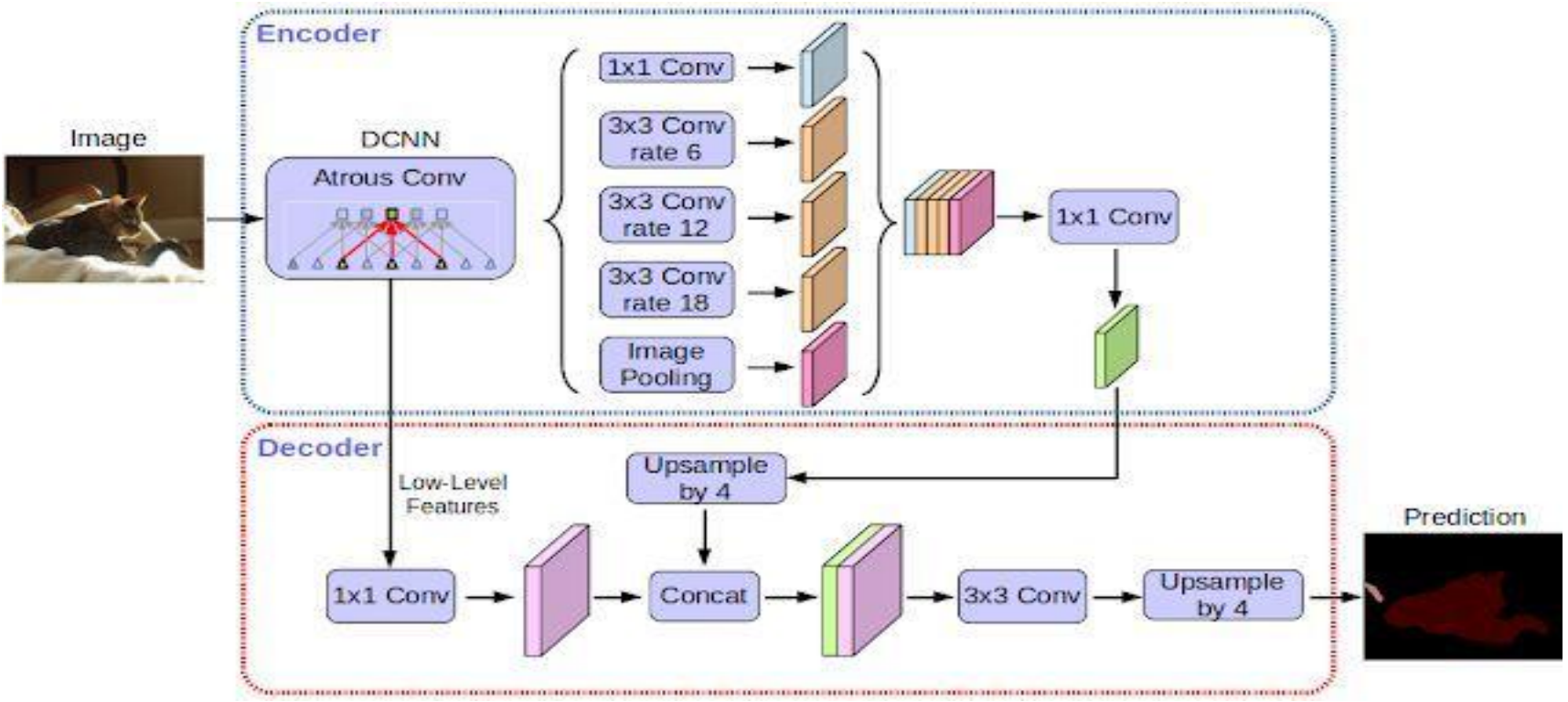
DeepLab v3+

- DeepLab-v3+模型是DeepLab系列模型最新的改进方案。

- 研究人员通过添加一个简单的解码器模型扩展DeepLab-v3，用于细化分割结果，在物体边界的划分上效果甚好。

DeepLab v3+

- V3+基本上可以理解成在v3的基础上增加了encoder-decoder模块，进一步保护物体的边缘细节信息。



数据集

Pascal VOC - 2012

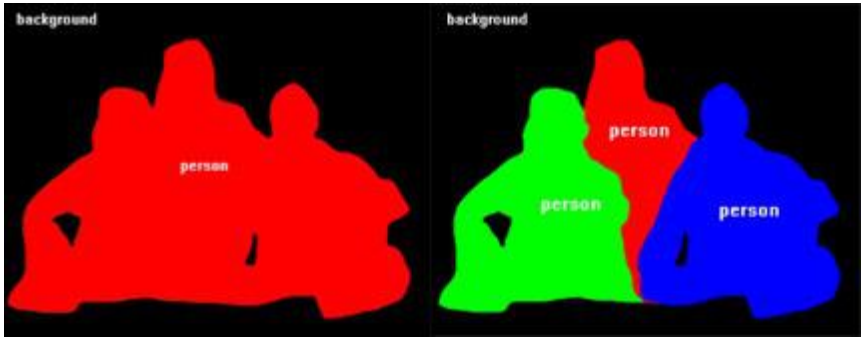
- 20个物体类别
  - 人类
  - 动物（鸟、猫、牛、狗、马、羊）
  - 交通工具（飞机、自行车、船、公共汽车、小轿车、摩托车、火车）
  - 室内（瓶子、椅子、餐桌、盆栽植物、沙发、电视）
- 像素级标签9,993张图片

数据集



MSCOCO

- 80个类别
- COCO-stuff扩展集： 172类别
  - Object: 80
  - Stuff: 91
  - Unknown: 1



- 主要用于：
  - 实例级别的分割（Instance-level）
  - 图片描述（Image Captioning）
- http://mscoco.org/

数据集

Cityscapes

- 30个类别
- 标注：
  - 5,000张像素标注（pixel level）
  - 20,000张多边形标注（instance level）
- 辅助/自动驾驶中的语义场景理解
- 采集于50个城市
- https://www.cityscapes-dataset.com

参考资料

- http://www.cnblogs.com/hjlweilong/p/6119822.html
- https://blog.csdn.net/xg123321123/article/details/53092154
- https://blog.csdn.net/c_row/article/details/52161394
- https://so.csdn.net/so/search/s.do?q=deeplab&t=blog&u=jyzhang_cvml
- https://blog.csdn.net/roslei/article/details/53404560

问答互动

在所报课的课程页面，

1、点击“全部问题”显示本课程所有学员提问的问题。

2、点击“提问”即可向该课程的老师和助教提问问题。



联系我们

小象学院：互联网新技术在线教育领航者

– 微信公众号：**小象学院**

