语义分割

Semantic Segmentation 简单说就是像素级别的目标检测

简述

summaries

历史进程

1. FCN
2. SegNet
3. Dilated Convolutions
4. DeepLab (v1 & v2)
5. RefineNet
6. PSPNet
7. Large Kernel Matters
8. DeepLab v3

9. Auto DeepLab

评价标准

FCN 论文 Results 部分总结了以下四个评价标准。定义 n_{ij} 为属于种类 i 的所有像素中，被预测为 种类 j 的数量；定义 t_i 为种类 i 的所有像素被分类的数量， $t_i = \sum_j n_{ij}$ 。

- pixel accuracy，像素精度，预测正确的像素点数除以所有类别的像素点：

$\sum_i n_{ii} / \sum_i t_i$

- mean accuraccy，平均精度，计算每一类的精度，对所有类取平均：

$(1/n_{cl}) \sum_i n_{ii}/t_i$

- mean IU，平均IU，每一类预测正确的数量除以其他类预测该类的数量与该类预测错了数量之和，对所有类取平均：

$(1/n_{cl}) \sum_i n_{ii}/(t_i + \sum_j n_{ji} - n_{ii})$

- frequency weighted IU，频率加权IU，在mean IU的基础上，乘以每一类的数量：

$(\sum_k t_k)^{-1} \sum_i t_i n_{ii}/(t_i + \sum_j n_{ji} - n_{ii})$

FCN

参考

参考博客1

参考博客2

参考博客3

参考博客4 shift and stitch

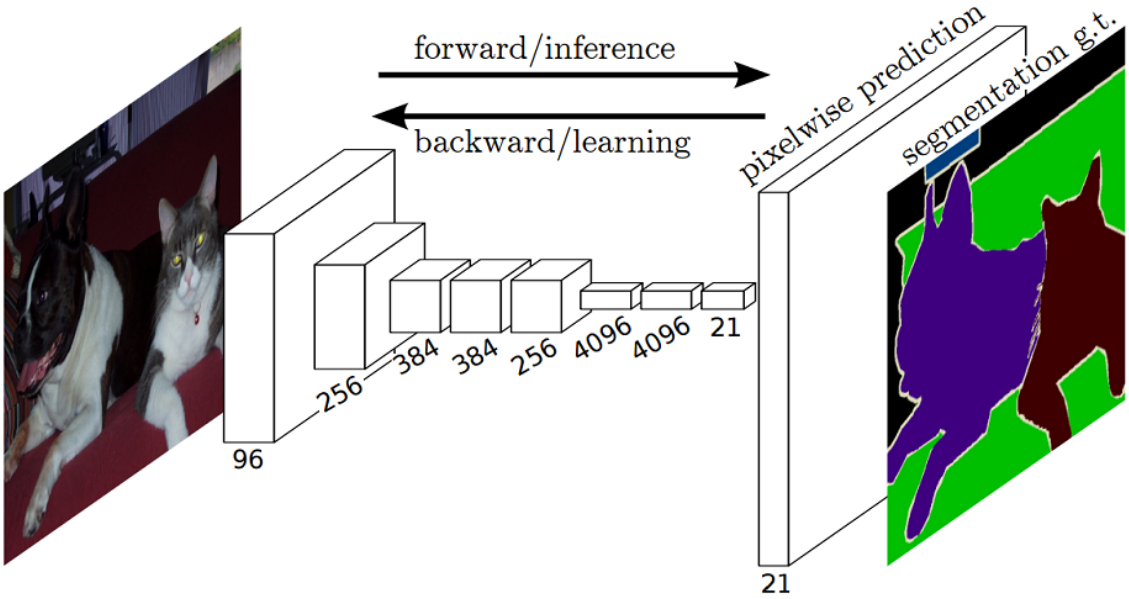
池化层BP推导

CV系列文章 FCN

反卷积

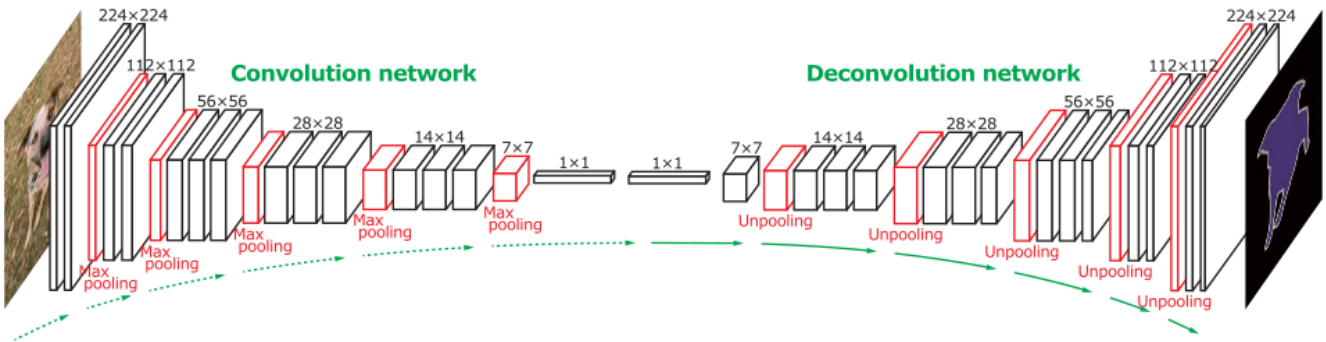
Jonathan Long 等人的《Fully Convolutional Networks for Semantic Segmentation》是深度学习在语义分割上的开山之作。FCN对图像进行像素级的分类，从而解决了语义分割。与经典的CNN在卷积层之后使用全连接层得到固定长度的特征向量进行分类（全联接层 + softmax输出）不同，FCN（FCN似乎是这篇文章先提出的：OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks）可以接受任意尺寸的输入图像，采用反卷积层对最后一个卷积层的 feature map 进行上采样， 使它恢复到输入图像相同的尺寸，从而可以对每个像素都产生了一个预测，同时保留了原始输入图像中的空间信息，最后在上采样的特征图上进行逐像素分类。主要贡献如下：

- Popularize the use of end to end convolutional networks for semantic segmentation
- Re-purpose imagenet pretrained networks for segmentation
- Upsample using deconvolutional layers
- Introduce skip connections to improve over the coarseness of upsampling



Why FCN

CNN在语义分割中有一个问题：“网络的输出不匹配”，即比如说416 * 416的输入，经过池化层，全连接层的处理，输出肯定小于416 * 416。所以我们需要先卷积（conv），然后反卷积（deconv，现在一般不称其为反卷积，transposed convolution更为贴切）



这样，FCN论文可以说定义了一种通用语义分割结构。

为什么采用FCN

- 输出不匹配
- 任意输入大小
- end to end

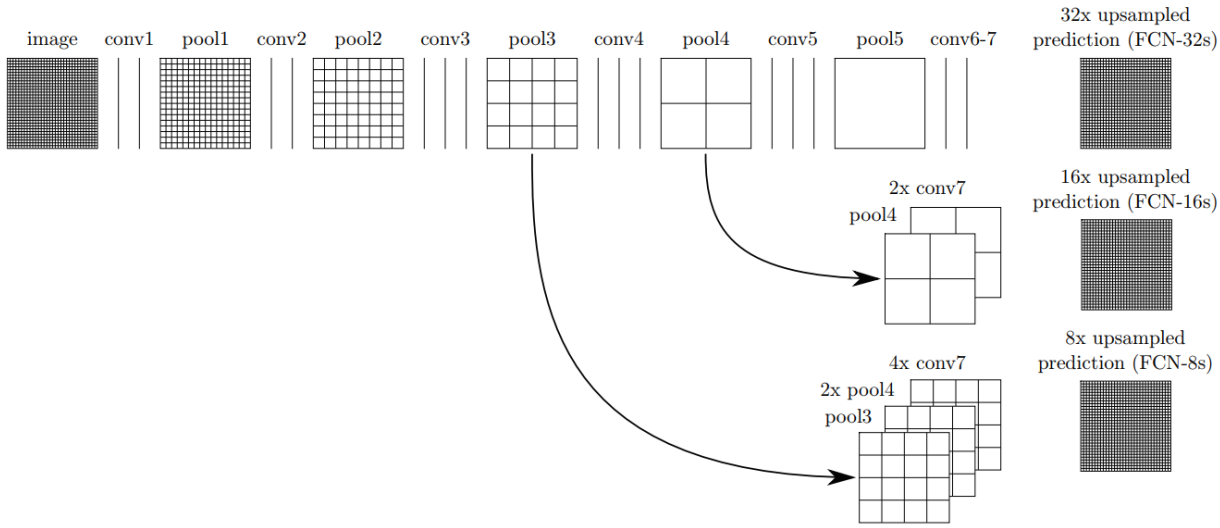
  得到语义分割通用结构 “卷积-反卷积” 的FCN

deconvolution

- **上采样层** 比较了几种方法/技巧，最终采用 deconvolution，即翻转卷积层的前向传播和反向传播，使其可以增加输出大小，同时，该层参数可学习。
- 训练 CNN 一般将整张图片作为网络的输入进行训练，即 whole image train。而语义分割需要将图像中的所有像素分类，故认为输入整张图片会有很多冗余，所以采用 patchwise train，即对一个像素，以它为中心取一个 patch（相当于裁剪后的小图片），这个 patch 作为网络的输入，网络的输出为该像素的标签。该方法提高了训练了速度，而且通过适当的 sampling 方法来选取 patches 作为训练集，可以平衡类别和解决空间相关性等问题。 本文通过实验，认为通过 weighting the loss and loss sampling，whole image train也可以达到平衡类别，解决空间相关性和加快训练速度，故 FCN 采用 whole image trian.

skip connections

卷积的过程，一直在减小输出的尺寸，最后一层输出的特征图信息丢失了很多，作者利用多个尺寸的特征图，叠加后上采样得到预测结果。以 FCN-16s 为例，将con7得到的1 * 1的特征图上采样为 2 * 2 之后和 pool4 之后的 2 * 2 的特征图组合，得到 16 倍上采样得到最终结果（即与输入相同的尺寸的输出）。



tricks on training

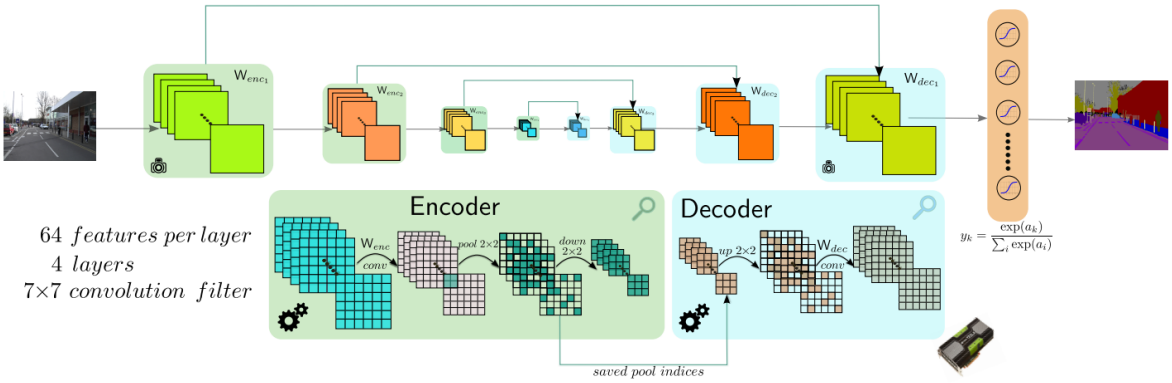
SegNet

参考

SegNet简介 条件随机场 CRF SegNet与其他论文的比较

SegNet架构

SegNet总体思想也是先conv，之后deconv，但是，它具体结构与FCN不同，它把第一层卷积的结构作为最后一层反卷积的输入，如图所示。直观看就是 skip connect 方式不同。



unpooling

SegNet还有一个特点：encoder 部分的池化层结果保存对应坐标，用于decode还原。如图，池化后的6对应一个坐标(1,1)，8对应(1, 3)，在对应的decoder层，将池化结果 n * n 还原为 2n * 2n，每个数字还原到对应的位置，其他位置置0（靠反卷积层学习这些位置的数值）

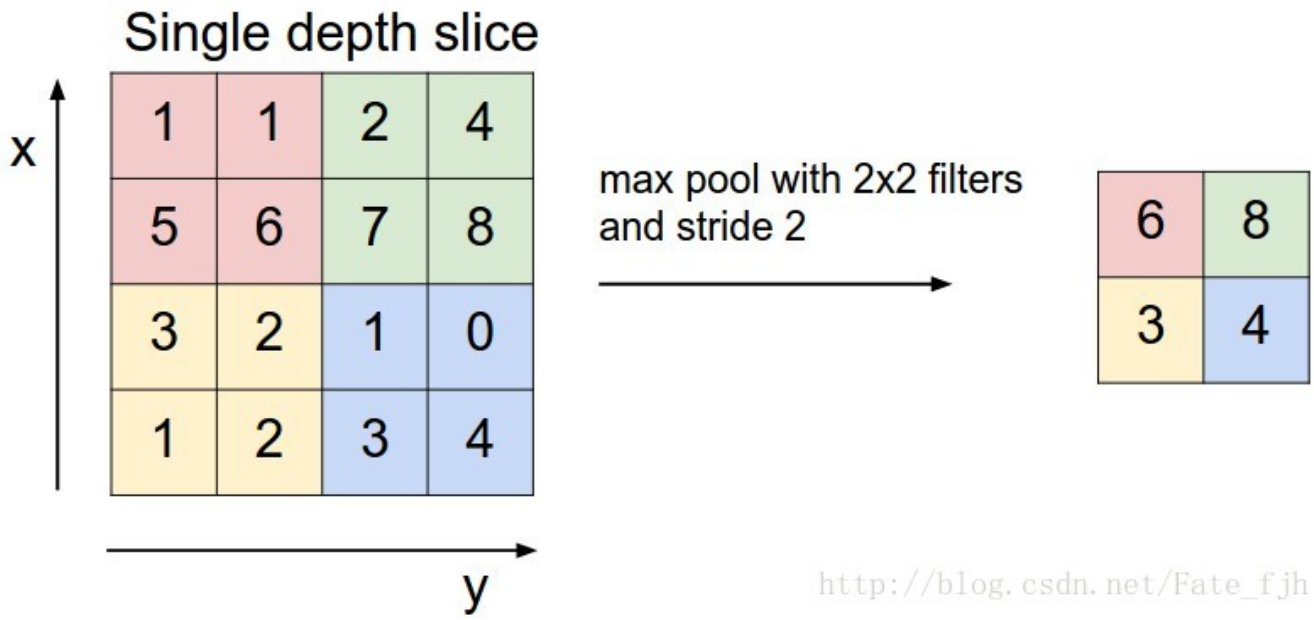
encoder pooling

index	0	1	2	3
0	1	1	2	4
1	5	6	7	8
2	3	2	1	0
3	1	2	3	4

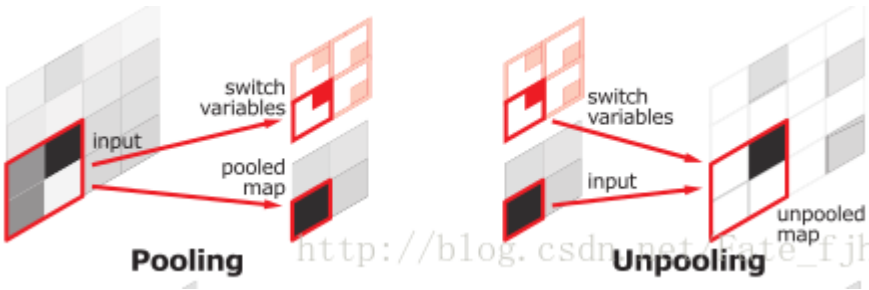
decoder unpooling

index	0	1	2	3
0	0	0	0	0
1	0	6	0	8
2	3	0	0	0
3	0	0	0	4

pooling



pooling to unpooling with location index



trick

实验证明，more 激活层，more work。

Dilated Convolutions

参考

空洞卷积 知乎

空洞卷积 知乎 （借鉴极多）

之前的问题

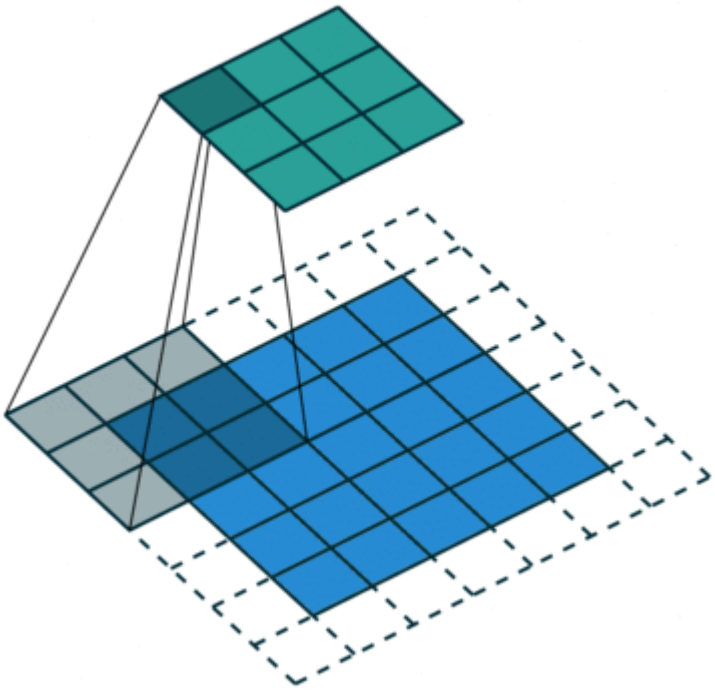
我们输送给神经网络的一般是多通道的矩阵，一个数字代表一个像素，但是单独的像素点信息太少，网络需要需要做到全局信息，或者说需要知道较大的区域的信息。我们把这区域叫做感知野。如何扩大感知野呢，现在有两个方法，池化层和带有步长的卷积层。之前的语义分割，比如FCN，他们主要依靠池化层进行下采样，降低图像的像素并且增大感知野，之后利用反卷积的上采样方法将图像恢复到原来的尺寸，并且将卷积过程中的不同尺寸的特征图结合提高性能；SegNet整理了整个过程，将网络分为编码和解码（encoder and decoder）两部分，并且将编码特征图与指定的解码部分连接，并且记录编码部分池化层保留的数字位置以增强位置信息利用率。简言之，就是两个手段：一个是池化（pooling）减少图像尺寸增大感受野，随后上采样扩大图像尺寸。二是提供图像的多个重新缩放版本（multiple rescaled verions of the image）作为网络的输入。

但是，在先减小再增大尺寸的过程中，肯定有信息损失；多个尺度的图像有很多重叠。是否需要下采样层？是否需要对多个重新放缩的图像进行分开分析？

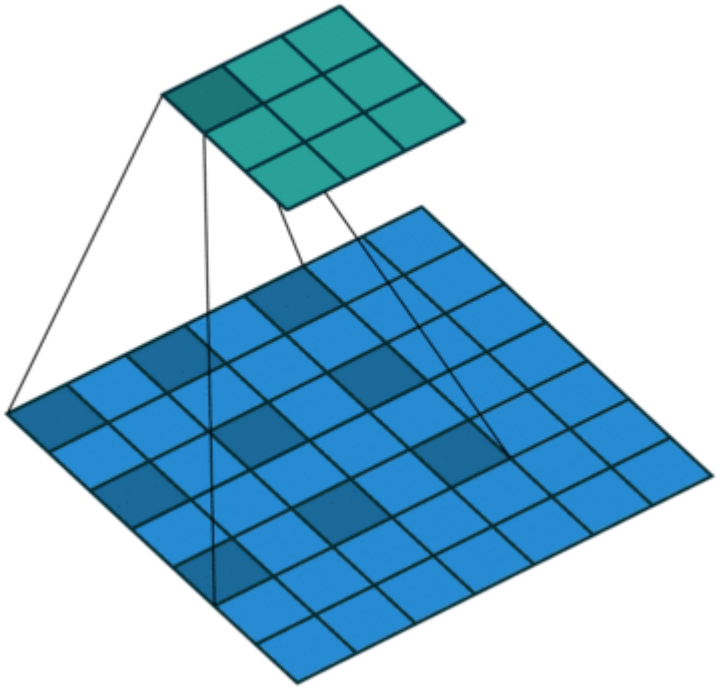
空洞卷积

对于第一个问题，本文的答案是空洞卷积。空洞卷积就是在传统卷积的基础上，增加了“跳步采样”。

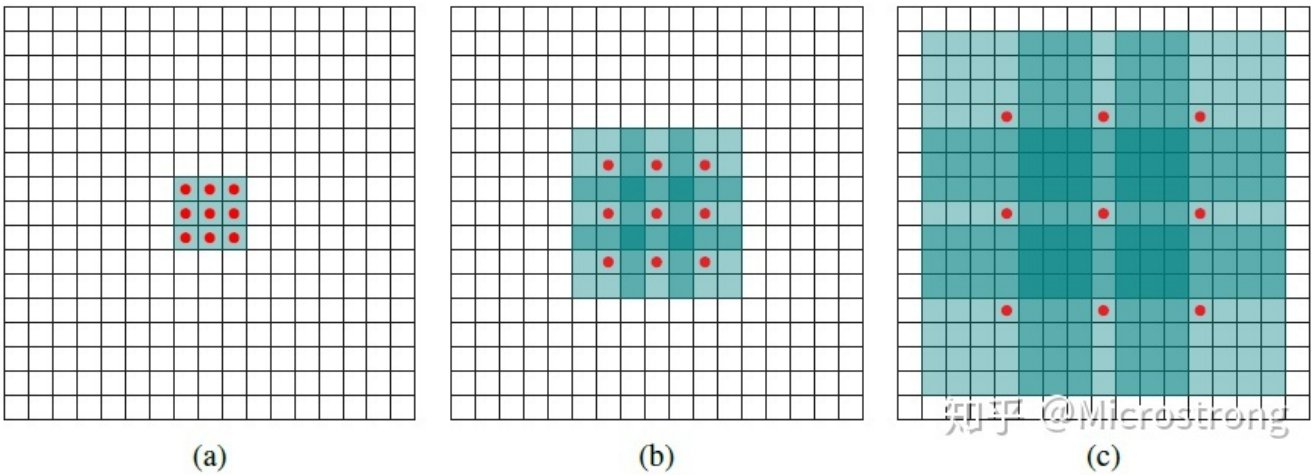
标准卷积：使用卷积核大小为 3 * 3、填充为1，步长为2的卷积操作对输入为 5 * 5 的特征图进行卷积生成3 * 3的特征图。



空洞卷积：使用卷积核大小为 3 * 3、空洞率（dilated rate）为2、卷积步长为1的空洞卷积操作对输入为 7 * 7 的特征图进行卷积生成 3 * 3 的特征图。



思考：为什么空洞卷积能够扩大感受野并且保持分辨率呢？



如图，红色圆点为卷积核对应的输入“像素”，绿色为其在原输入中的感受野。

(a)图对应3x3的扩张率为1的卷积，和普通的卷积操作一样；

(b)图对应3x3的扩张率为2的卷积，实际的卷积核还是3x3，但是空洞率为2，也就是对于一个7x7的图像块，只有9个红色的点也就是3x3的卷积核发生卷积操作，其余的点略过。也可以理解为卷积核的大小为7x7，但是只有图中的9个点的权重不为0，其余都为0。可以看到虽然卷积核的大小只有3x3，但是这个卷积的感受野已经增大到了7x7。如果考虑到这个2-dilated convolution的前一层有一个1-dilated convolution的话，那么每个红点就是1-dilated的卷积输出，感受野为3x3，所以1-dilated和2-dilated合起来就能达到7x7的卷积；

(c)图是4-dilated convolution操作，同理跟在1-dilated和2-dilated convolution的后面，能达到15x15的感受野。

对比传统的卷积操作，3层3x3的卷积加起来，stride为1的话，只能达到(kernel-1) * layer+1=7的感受野，也就是和层数layer成线性关系，而dilated conv的感受野是指数级的增长。

补充：dilated的好处是不做pooling损失信息的情况下，加大了感受野，让每个卷积输出都包含较大范围的信息。在图像需要全局信息或者语音文本需要较长的sequence信息依赖的问题中，都能很好的应用dilated conv，比如图像分割、语音合成WaveNet、机器翻译ByteNet中。

Multi-Scale Context Aggregation

为解决第二个问题，本文的提出了 Multi-Scale Context Aggregation 。

DeepLab v1

DCNN在计算机视觉领域大放光彩。其在高层次的视觉问题，如图像识别、目标检测等任务上取得了很好的成绩，但是在语义分割、姿态估计等底层视觉任务上存在缺陷，精度较低。Google Liang-Chieh Chen等研究人员认为原因是：DCNN的高级特征平移不变性。故提出DeepLab，将深度卷积神经网络（DCNNs）和概率图模型（DenseCRFs）结合，并且在卷积网络上应用空洞卷积扩展感知野，得到的很好的效果。

DCNN在图像标记任务（image labeling tasks）上存在两个技术障碍：信号下采样（signal downsampling）和空间不敏感（ spatial 'insensitivity' ）。第一个问题源于DCNN重复池化和下采样造成的分辨率下降，DeepLab采用空洞卷积解决这个问题；第二个问题采用CRF解决。即将原先纯卷积网络的 encode-decode 结构改为 DCNN-CRFs 结构。

atrous算法

CRFs

CRFs还结合了多尺度预测

DeepLab v2

又过了一年，DeepLab团队抖擞精神，总结经验，提出了DeepLab v2。他们进一步总结了语义分割面临的挑战，归为以下三点：1. reduced feature resolution, 2. existence of objects at multiple scales, and 2. reduced localization accuracy due to DCNN invariance。

第一个挑战，特征减少，是由最大池化和下采样（卷积的步长造成尺寸减小）造成的。这主要归咎于之前的卷积神经网络（如AlexNet，VGG，ResNet）都是面向图像识别任务设计的。解决方案是以空洞卷积取代池化和卷积，也顺带取消了反卷积。

第二个挑战，多尺度问题。传统做法是类似R-CNN，将图片部分扩展为原尺寸。这种做法效率低，本文采用类似SPP-Net的做法，多尺度采样金字塔结构（SPP），并与空洞卷积结合，得到 ASPP。

第三个挑战，CRFs。

ASPP

v2和v1相比，改进不大，主要就是吸取SPP-Net的特点，提出了ASPP。

DeepLab v3

v3 修改网络结构，加深ResNet，

调整/优化 atrous conv

取消CRFs

v3+ 整理提出decoder module，进一步“修饰”边界信息。

将Xception作为基础网络，并做修改

Auto DeepLab