# 总述：

这次周报主要为周五至周天的学习内容，研究了大部分较经典的语义分割模型，后续任务是DeepLab V3的总结。然后会再花1天左右时间，进行图像分割的实验（目前只做了一部分，还有些模型因为环境还没配好，所以实验没做，GitHub上没更新相应的实验结果）。下一周的计划开始迁移学习的survey阅读并做总结，理想情况为做完总结的基础上能将实验也做完。初步时间安排是，周中看完综述，周末做实验与报告的完善，尽量多做一些东西。

# 语义分割：

什么是语义分割? 我们知道视觉任务有三大类：图像分割、目标检测、目标定位。图像分割包含：语义分割、实例分割等等，语义分割其实是在像素级别上的分类，属于同一类的像素都要被归为一类，因此语义分割是从像素级别来理解图像的。

而语义分割不同于实例分割，举例来说，如果一张照片中有多个人，对于语义分割来说，只要将所由人的像素都归为一类，但是实例分割还要将不同人的像素归为不同的类。也就是说实例分割比语义分割更进一步。如下图所示：



原图



语义分割



实例分割

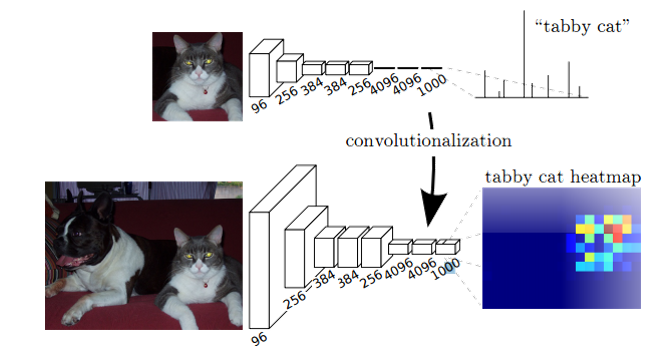
下面具体阐述一些语义分割模型及一点小思考

1. FCN

顾名思义，FCN是fully convolutional networks，全卷积神经网络。

1. **全卷积化：**

在普通卷积神经网络中，因为全连接层的存在，导致网络的输入必须是固定的。而FCN用卷积代替了全连接层,因此使任意图像大小的输入都变成可能。



这幅图显示了卷积化的过程,图中显示的是AlexNet的结构，简单来说卷积化就是将其最后三层全连接层全部替换成卷积层这意味着，我们可以不像传统方法一样，一个patch一个patch进行预测，而是对输入的任意大小的图像进行预测，并且输出图像大小与输入相对应。

但是这样也带来了一些问题：因为卷积层之后的pooling操作，会使得分辨率降低，虽然提取了更高维度信息，但是削弱了空间信息，而语义分割中需要score map和原图对齐，因此需要丰富的位置信息。

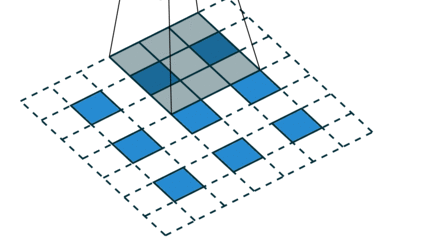
1. **反卷积(deconvolution)** ：

也可以称为，上采样。上采样也就是对应于上图中最后生成heatmap的过程。在解释上采样之前，先说说我们熟识的下采样。在一般的CNN结构中,如Alex Net, VGGNet均是使用池化层来缩小输出图片的size，例如VGG16,五次池化后，图片被缩小了32倍; 而我们需要一个与原图像size相同的分割图，因此我们需要对最后一层进行上采样，其实也被称为反卷积(Deconvolution)

那么究竟什么是反卷积，网上有一个解释我觉得很棒：反卷积是一种特殊的正向卷积，先按照一定的比例通过补 0 来扩大输入图像的尺寸，接着旋转卷积核，再进行正向卷积。

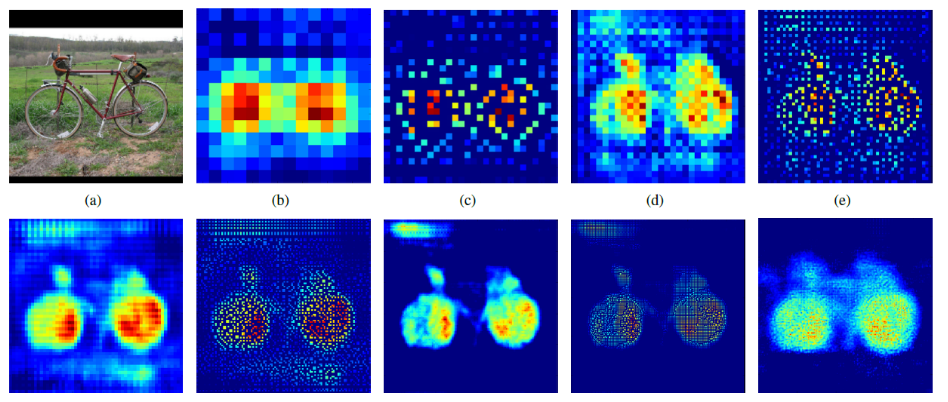
具体过程如下：

假设原图是3X3，首先使用上采样让图像变成7X7（通过填充0），如下图所示：可以看到图像多了很多空白的像素点。



使用一个3X3的卷积核对图像进行滑动步长为1的valid卷积，得到一个5X5的图像，我们知道的是使用上采样扩大图片，使用反卷积填充图像内容，使得图像内容变得丰富，这也是CNN输出end to end结果的一种方法。这样图像就从3\*3，变成了5\*5.

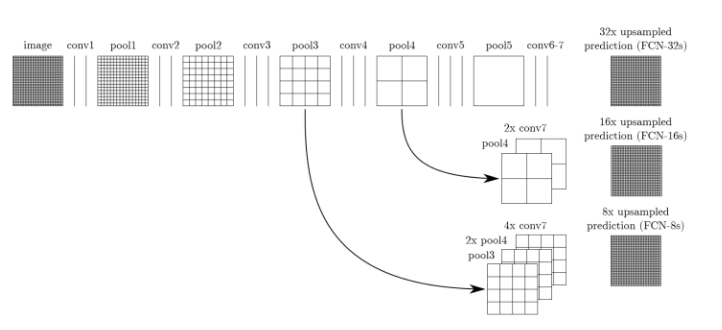
那么，这么做是否合理呢？韩国作者Hyeonwoo Noh使用VGG16层CNN网络后面加上对称的16层反卷积与上采样网络实现end to end 输出，其不同层上采样与反卷积变化效果如下



可以看到，这样做的效果实际上是很好的，得到的结果图经过上采用成功和原图尺寸一致，并且物体位置大致是相对应的。

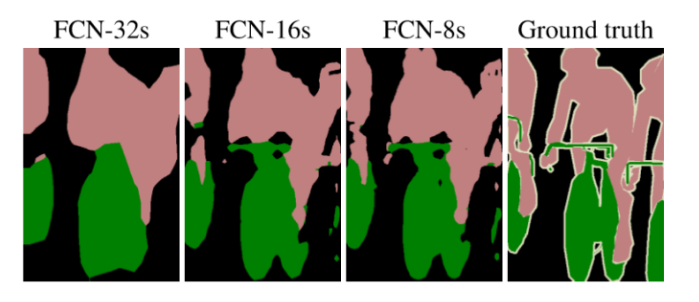
1. **跳跃结构(skip architecture)**：

其实直接使用前两种结构就已经可以得到结果了，但是直接将全卷积后的结果上采样后得到的结果通常是很粗糙的。所以这一结构主要是用来优化最终结果的，思路就是将不同池化层的结果进行上采样，然后结合这些结果来优化输出，具体结构如下:



这种跨层之间的连接操作比较经典，融合了网络浅层的细(fine-grain)粒度信息以及深层的粗糙(coarse)信息。

而不同的结构产生的结果对比如下:



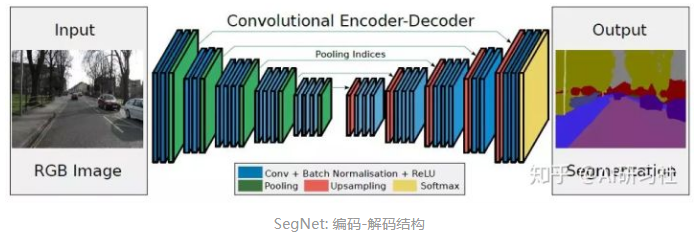
不同结构对比图

**2. Segnet**

FCN的上卷积层和一些跳跃连接产生的分割图比较粗糙，因此SegNet增加了更多的跳跃链接。不过，SegNet并不是直接将encoder的特征进行直接复制，而是对maxpooling的指数进行复制（相当于引入了池化索引），这使得SegNet的效率更高(内存更方便调用)。

1. 编码-解码器架构

其实这个架构在之前的FCN就出现了，其意义也是一样，这里不再多余阐述，用一个图来清晰表达下结构。



如图所示，左边为编码器网络，右边为解码器网络，最后用Softmax进行分类。

总结：FCN和SegNet都是encoder-decoder架构。但是在benchmark上面表现还不如FCN，不过速度效率更高。

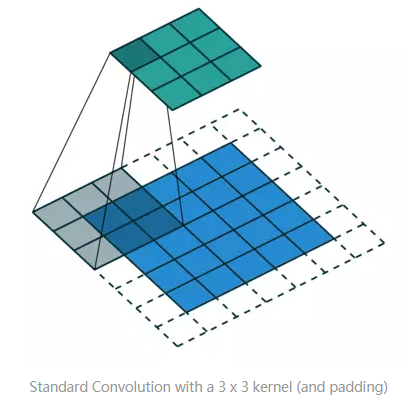
## DeepLab(v1,v2)

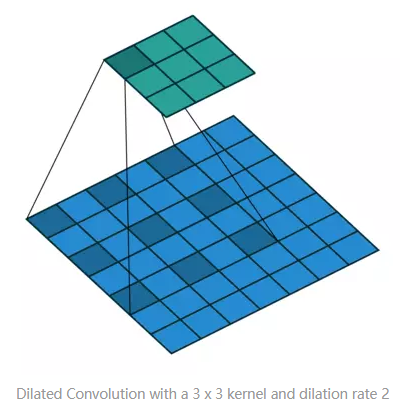
## DeepLab是Google团队一系列semantic image segmentation的paper.其主要创新点有三个，所以下面具体讲解创新点

1. 使用了空洞卷积

下面介绍一下空洞卷积：

Dilated/Atrous Convolution 或者是 Convolution with holes 从字面上就很好理解，是在标准的 convolution map 里注入空洞，以此来增加 reception field。相比正常的卷积操作，它额外多了一个参数：dilation rate，也就是核间隔的数量。



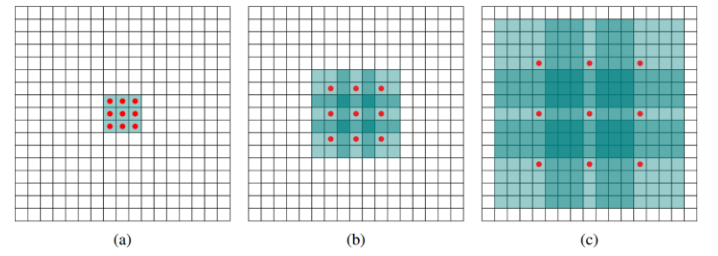


理解如何做很好理解，难点在于理解为什么有效。

还是从传统的卷积入手，卷积过程中的下采样/池化操作，是不可以学习的，在确立网络结构初期就已经确定。然而这个过程会使得很多信息丢失，比如小物体信息无法重建（假如有4个pooling layer，则任何小于 2^4 = 16 pixel 的物体信息将理论上无法重建。）；内部数据结构丢失；空间层级化信息丢失等等问题。这也制约了语义分割的精度。

这里回想起了初期人们在对神经网络框架搭建时，普遍都喜欢用大卷积，因为大卷积receptive field（感受野）越大，看到的图片信息越多，因此获得的特征越好。而空洞卷积正是从这方面入手，使得卷积能够看到更多的信息，也就保留了更多的内部信息，从而增加语义分割的精度。

标准的3×3卷积核只能看到对应区域3×3的大小，但是为了能让卷积核看到更大的范围，dilated conv使其成为了可能。dilated conv原论文中的结构如图所示：



如图b，卷积核大小依然是3×3，但是每个卷积点之间有1个空洞，也就是在绿色7×7区域里面，只有9个红色点位置作了卷积处理，其余点权重为0。这样即使卷积核大小不变，但它看到的区域变得更大了。

但是，这其中也有一些问题值得注意，但不是这里主要研究的，后续关于空洞卷积的改进可以参考Hybrid Dilated Convolution (HDC)。这里就留一个坑了

1. 使用了全连接条件随机场

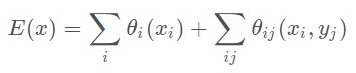
要了解全连接条件随机场，需要先了解随机场、再到马尔可夫随机场、再到条件随机场、再到线性条件随机场、再到全连接条件随机场。这里我就不写这么多了，假设都知道前面的情况下，说说我对全连接条件随机场的看法：

首先，将条件随机场作用于图像分割。我们都知道，一张图片由多个像素点构成，如果每个像素点都是一个无向图的顶点，那么一个随机场便构成了。图像分割的过程，就是将每个顶点赋予不同的label目标或背景），即，将无向图中的边在边界处正确地切分开。

在图片分割当中，具有相似位置和颜色特征的两个像素，其被赋上相同label的概率大，则被分割的可能性小，这就对应了条件随机场中的概率模型。

除此以外，条件随机场可以对所有特征进行全局[归一化](https://link.zhihu.com/?target=http%3A//baike.baidu.com/item/%25E5%25BD%2592%25E4%25B8%2580%25E5%258C%2596)，能够求得全局的[最优解](https://link.zhihu.com/?target=http%3A//baike.baidu.com/item/%25E6%259C%2580%25E4%25BC%2598%25E8%25A7%25A3)。这就对应到能量函数求最小值。也就是说，在各种连接当中，寻找一个最优的分类方式，本质就是一个最优化问题。

而能力函数定义如下：

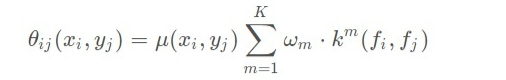


这个能量函数计算的是所有像素点的能量和：

前半部分是一个一元函数：

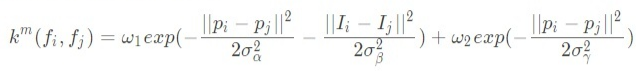


而核心在于后半部分的函数：



其中：

，label compatibility：约束力的传导方向，只有label相同的能力才可以互相传传导。例如，“a像素是飞机的一部分”、“b像素是飞机的一部分”、“c像素是人的一部分”，显然a、b才可以互相传导，而c则不能。

，代表了不同像素间的亲密度：

其中，前半部分成为appearance kernel，其中，p表示像素的位置，I表示像素的值[对应RGB三维]，而他们前面都有一个 – 号，所以像素距离越近，像素值更接近，结果就越强，反之则越弱；并且分母，表示方差，越大则能量越难强起来。

而后半部分则仅仅是一个平滑作用。

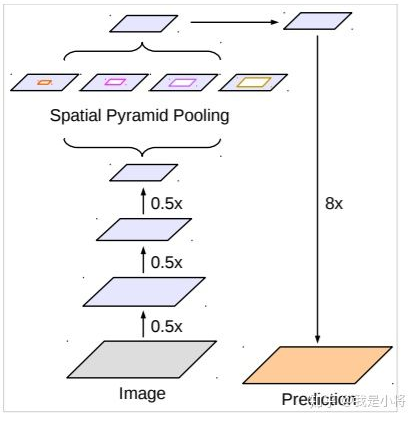
这样一个全连接条件随机场便解释清楚了，主要是通过全连接条件随机场实现结构化预测，需将条件随机场的训练和微调单独作为一个后期处理步骤，以改善分割效果。

1. 在空间维度上实现金字塔型的空洞池化（ASPP）

此处的金字塔型空间池化有点像PSPNet里面的金字塔池化版块，只是作用不一样，具体后面讲述PSPNet会提到。

在DeepLab中，采用空间金字塔池化模块来进一步提取多尺度信息，这里是采用不同rate的空洞卷积来实现这一点。

ASPP模块主要包含以下几个部分：



**RefineNet**

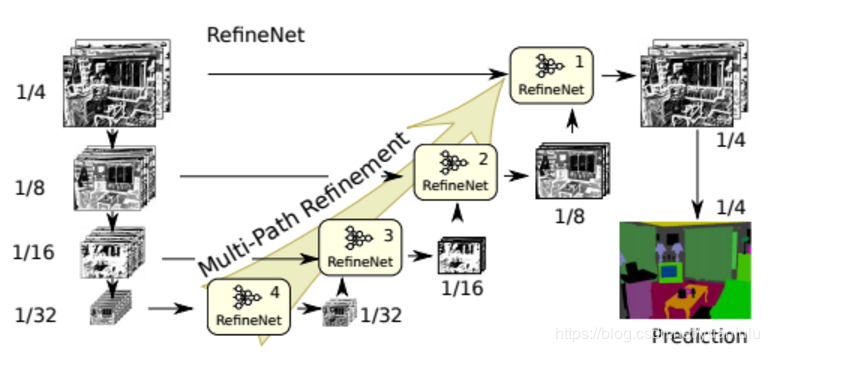
在总结这篇文章时，个人觉得自deeplab V1 V2之后的文章，创新点没有之前文章那么意义重大（基本算是奠定了语义分割基础，如：FCN, U-net , 空洞卷积 ）。

下面开始介绍RefineNet，前面文章的deconvolution操作或者空洞卷积的操作其实都是为了解决一个问题。由于pooling和卷积步长的存在，feature map会越来越小，导致损失一些细粒度的信息（低层feature map有较丰富的细粒度信息，高层feature map则拥有更抽象，粗粒度的信息）。这个现象对于分类问题是完全可以接受的，但是对于逐像素的分割问题，除了需要强语义信息之外，还需要高空间分辨率。

作者认为高级语义特征可以更好地进行分类识别，而低级别视觉特征有助于生成清晰、详细的边界。所以作者基于另一种方法-基于skip connection的方式提出了RefineNet：

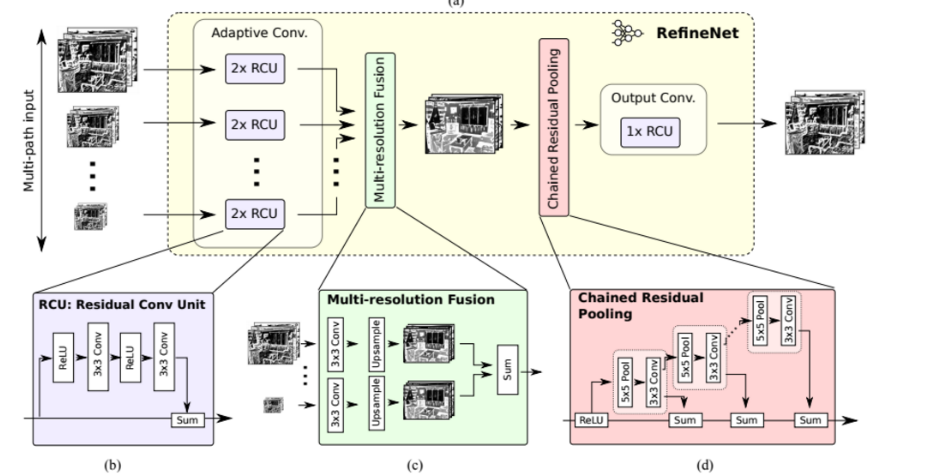
1. 提出一种多路径refinement网络，称为RefineNet。这种网络可以使用各个层级的features，使得语义分割更为精准。
2. RefineNet中所有部分都利用residual connections（identity mappings），使得梯度更容易短向或者长向前传，使段端对端的训练变得更加容易和高效。
3. 提出了一种叫做chained residual pooling的模块，它可以从一个大的图像区域捕捉背景上下文信息。

我觉得RefineNet还是一个加强版的U-Net，效果好是因为参数和计算量上来了（RefineNet部分参数变多了），之所以还能端到端训练是因为内部使用了大量跳层连接减少了梯度弥散。



可以看到decoder顺次往上融合更高阶段的特征，最后做softmax分类，然后4倍线性插值恢复到原图大小。

具体结构则如下：



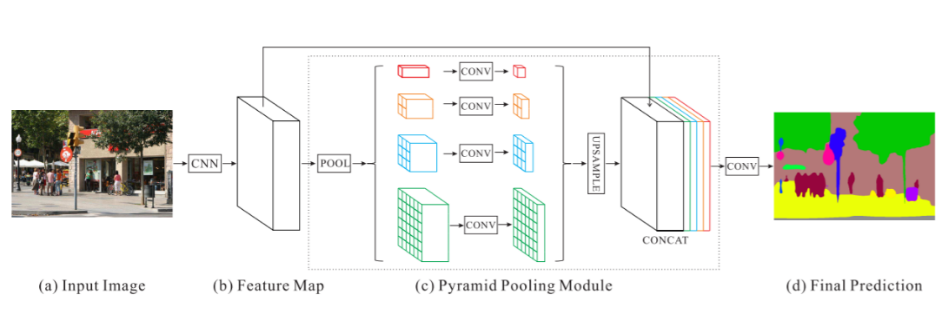
# PSPNet

提出的金字塔池化模块( pyramid pooling module)能够**聚合不同区域的上下文信息**,从而提高获取全局信息的能力。

在一般CNN中感受野可以粗略的认为是使用上下文信息的大小，论文指出在许多网络中没有充分的获取全局信息，所以效果不好。要解决这一问题，常用的方法是：

1. 用全局平均池化处理。但这在某些数据集上，可能会失去空间关系并导致模糊。
2. 由金字塔池化产生不同层次的特征最后被平滑的连接成一个FC层做分类。这样可以去除CNN固定大小的图像分类约束，减少不同区域之间的信息损失。

论文提出了一个具有层次全局优先级，包含不同子区域之间的不同尺度的信息，称之为pyramid pooling module：



模块融合了4种不同金字塔尺度的特征，第一行红色是最粗糙的特征–全局池化生成单个bin输出，后面三行是不同尺度的池化特征。为了保证全局特征的权重，如果金字塔共有N个级别，则在每个级别后使用 1×1 的卷积将对于级别通道降为原本的1/N。再通过双线性插值获得未池化前的大小，最终concat到一起。