# Fully Convolutional Networks for Semantic Segmentation

## FCN

参考链接：<https://blog.csdn.net/shuqiaos/article/details/88854807>

卷积网络（Convnet）中的每个层都是一个尺寸为h×w×d h\times w\times dh×w×d的3维array，其中h hh和w ww都是空间尺寸，d dd是特征维度或通道数。第一层是图像，尺寸为h×w h\times wh×w，颜色通道数为d dd。高维层中的位置对应回与其直接相关的图像的位置，叫做感受野（receptive fields）。

Writing xij for the data vector at location (i，j) in a particular layer, and yij for the followinglayer, these functions compute outputs yij by

1572863397(1)

where k is called the kernel size, s is the stride or subsampling factor, and fks determines the layer type

This functional form is maintained under composition,with kernel size and stride obeying the transformation rule：

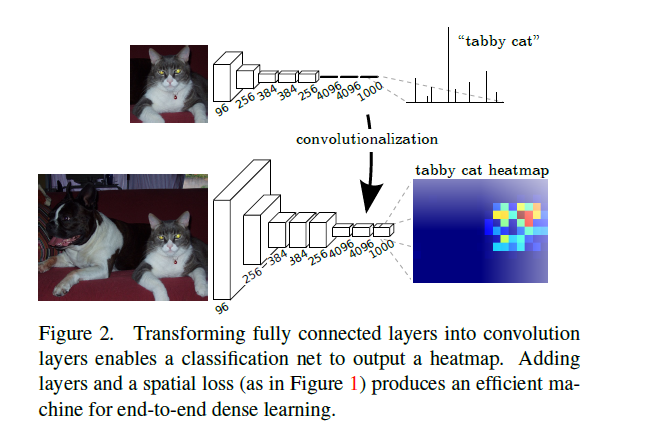
1572864252(1)

When these receptive fields overlap significantly, both feedforward computation and backpropagation are much more efficient when computed layer-by-layer over an entire image instead of independently patch-by-patch.

通常cnn网络在卷积之后会接上若干个全连接层，将卷积层产生的特征图（feature map）映射成为一个固定长度的特征向量。一般的CNN结构适用于图像级别的分类和回归任务，因为它们最后都期望得到输入图像的分类的概率，如ALexNet网络最后输出一个1000维的向量表示输入图像属于每一类的概率。

FCN对图像进行像素级的分类，从而解决了语义级别的图像分割问题。与经典的CNN在卷积层使用全连接层得到固定长度的特征向量进行分类不同，FCN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基层的特征图（feature map）进行上采样，使它恢复到输入图像相同的尺寸，从而可以对每一个像素都产生一个预测，同时保留了原始输入图像中的空间信息，最后奇偶在上采样的特征图进行像素的分类。

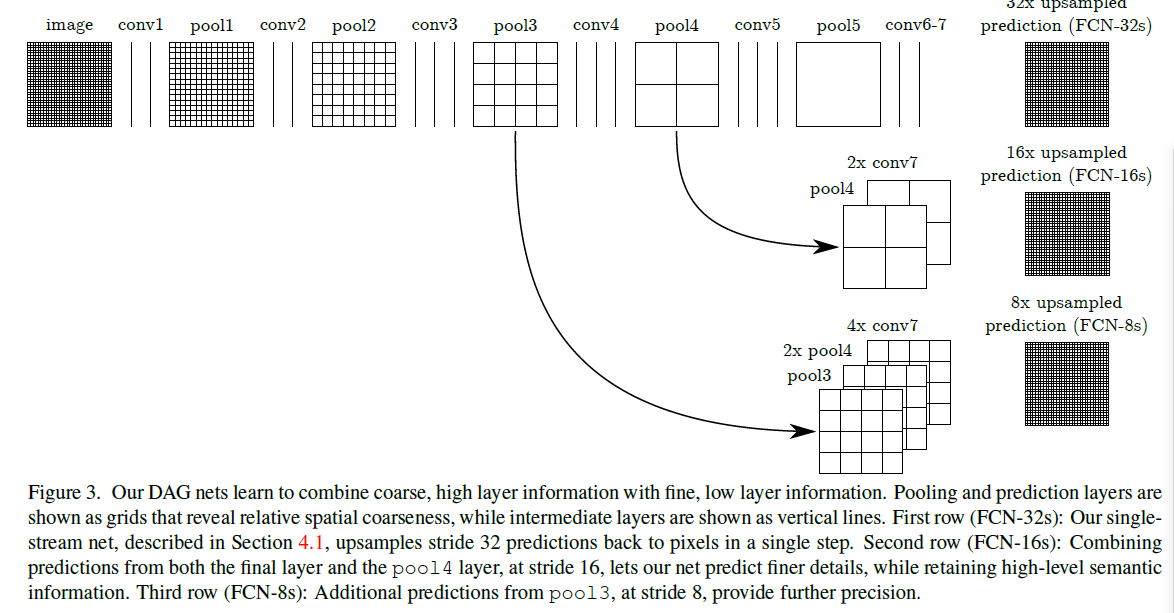
FCN将传统CNN中的全连接层转化成一个个的卷积层。如下图所示，在传统的CNN结构中，前5层是卷积层，第6层和第7层分别是一个长度为4096的一维向量，第8层是长度为1000的一维向量，分别对应1000个类别的概率。FCN将这3层表示为卷积层，卷积核的大小(通道数，宽，高)分别为（4096,1,1）、（4096,1,1）、（1000,1,1）。所有的层都是卷积层，故称为全卷积网络。



下面参考<https://blog.csdn.net/qq_36269513/article/details/80420363>

如下图所示：对原图进行卷积conv1、pool1后图像缩小为1/2；对图像进行第二次卷积conv2、pool2后图像缩小为1/4；对图像进行第三次卷积conv3、pool3后图像缩小为1/8，此时保留pool3的featuremap；对图像进行第四次卷积conv4、pool4后图像缩小为1/16，此时保留pool4的featuremap；对图像进行第五次卷积conv5、pool5后图像缩小为1/32，然后把原来CNN操作过程中的全连接编程卷积操作的conv6、conv7，图像的featuremap的大小依然为原图的1/32,此时图像不再叫featuremap而是叫heatmap。

其实直接使用前两种结构就已经可以得到结果了，这个上采样是通过反卷积（deconvolution）实现的，对第五层的输出（32倍放大）反卷积到原图大小。但是得到的结果还上不不够精确，一些细节无法恢复。于是将第四层的输出和第三层的输出也依次反卷积，分别需要16倍和8倍上采样，结果过也更精细一些了。这种做法的好处是兼顾了local和global信息。



上图中，32x即为扩大32倍。Pool5扩大32倍就可以得到原来图像大小了。

Pool5扩大2倍与Pool4融合得到，再扩大16倍也可以得到原来图像大小了。

扩大2倍与Pool3融合再扩大8倍也可以得到原来图像大小了。

再将我觉得理解最好的放上来https://zhuanlan.zhihu.com/p/30195134?utm\_source=qq&utm\_medium=social&utm\_oi=1014976979178848256