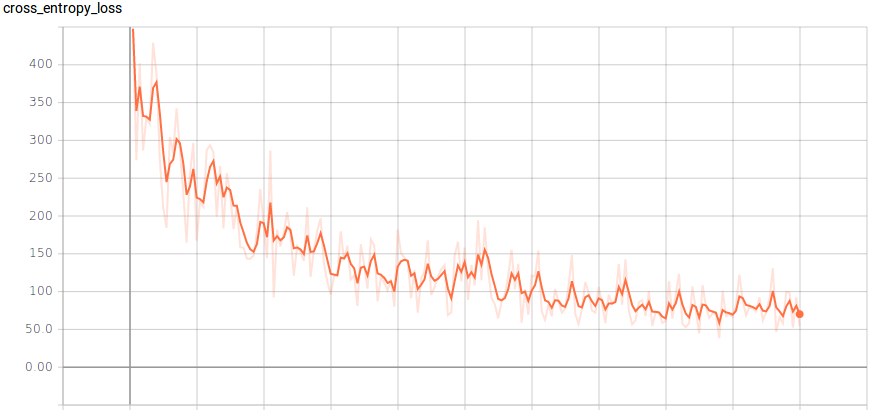
使用卷积神经网络进行图像分割

1.使用fcn\_8s进行图像分割

learning rate=0.0001，loss=53.75，损失变化如图。

图1

训练集上效果

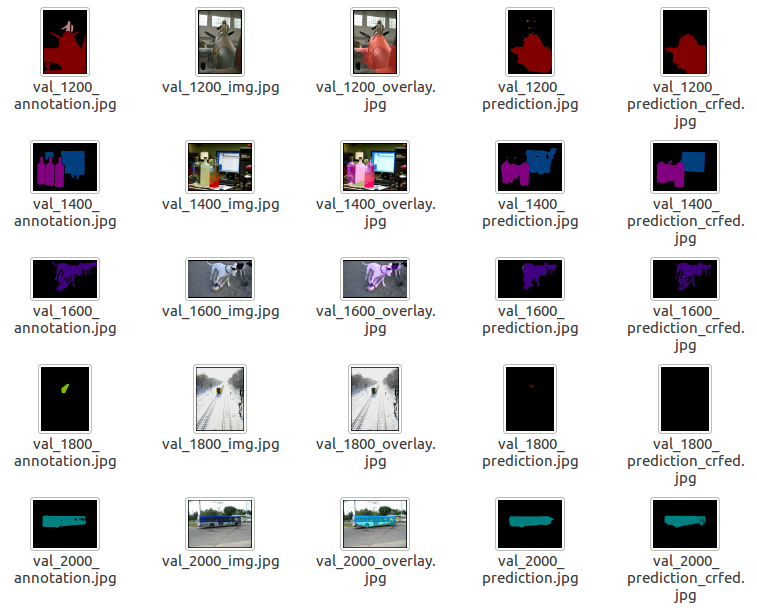
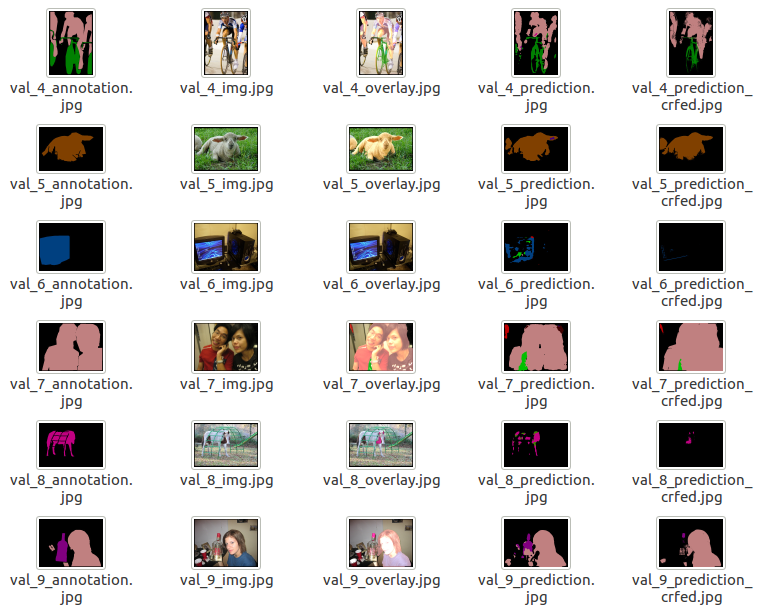


图2

经过2000个step后，测试集上效果

图3

训练到2000个step时，loss下降趋于平缓，但是仍然较大，尝试了前800个step使用lr=0.0003，之后600个step使用lr=0.0001，之后400个step使用lr=0.0005，之后200个step使用0.0001。loss变化如图

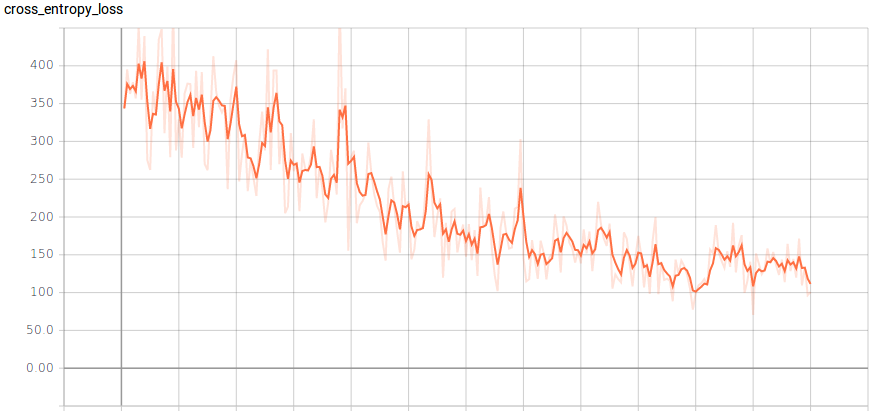


图4

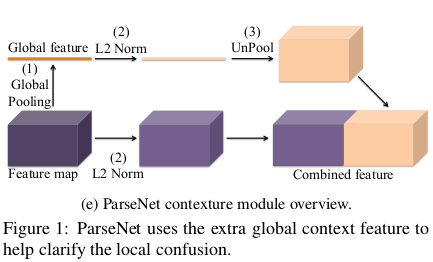
相较于2000个step都使用学习率为0.0001，虽然前期loss较小（350 vs 400+)，但是变化趋势明显更为缓慢，最终稳定在100左右。作为一次失败的尝试，希望助教能教教怎么调参。

从图片效果上看，FCN对小物体的分割效果比较差，细节上恢复的不够好，如图2中val\_1800。CRF作为后端处理旨在恢复物体边缘，有时候还没有预测的结果好，比如图3中，val\_8\_prediction还有椅子腿，val\_8\_predicition\_crfed就被处理得几乎无法辨认椅子的形状了；但有时又确实能恢复物体边缘，如图2中val\_2000，经过CRF后分割结果包括了车轮。

2.语义分割简述

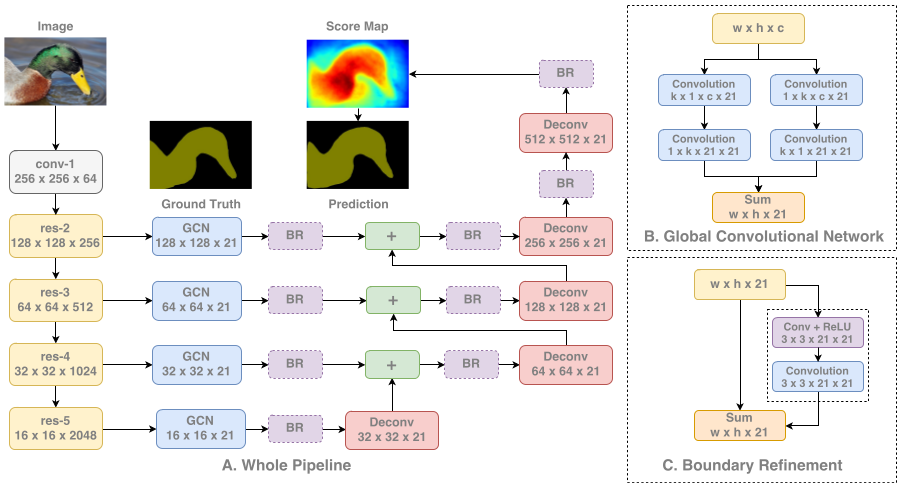
2014年11月FCN作为使用卷积神经网络进行语义分割的开山之作被提出。通过将经典网络，如VGG等，的全连接层改为卷积层，使得原本只能接受固定大小的图片作为输入的卷积神经网络，可以接受任意大小的图片，并输出相应大小（对于VGG-16，w和h是输入图片大小的1/32，channel不变）的特征。然后通过双线性插值(通过transposed\_convolution实现，卷积核的learning rate=0)，将特征图的长宽放大32倍，得到原图中每一个像素点属于某一类的概率。但是从1/32的输出特征图中预测原图中的每一个像素点属于某一类的概率依旧困难，需要将输出特征的维度变大(dense)。考虑将卷积或池化的步长调小，但是这样原来经3\*3，stride=2的卷积层输出到下一层，若采用stride=1，那么下一层3\*3 kernel的感受野由7\*7变成了5\*5，原先训练的权重就不再适用了；考虑shift-and-stitch(对1\*9的输入做1\*3的池化，首先padding=1，输入变成1\*11，对1~9做池化，得到1\*3的输出；对2~10做池化，得到1\*3的输出；对3~11做池化，得到1\*3输出，合计1\*9输出，即通过shift-and-stitch使得池化不改变输入维度)，本质上是将卷积核做了rarefaction(将滤波器从[1 1 1]变成了[1 0 1 0 1]或[0 1 0 1 0 1])，这样做虽然感受野没变小，但是检测物体的尺度变大了(the filters are prohibited from accessing information at a finer scale than their original design)。最终提出了skip-architecture,把fc7的输出通过双线性插值放大2倍，然后与经1\*1卷积(0初始化，使得与FCN-32s的预测结果相同)获得分类结果的pool4相加(FCN-16s)；把相加的结果通过双线性插值放大2倍(transposed\_convolution, filter can be learned)，然后与pool3相加(FCN-8s)；最后放大8倍得到与原图等大的结果。

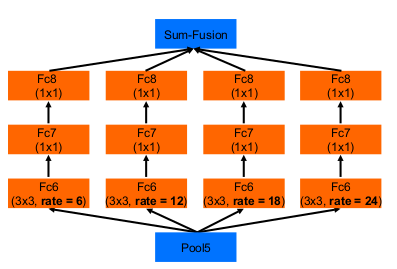
2014年12月DeepLab v1在VOC2012上mIOU=71.6%。使用Titan GPU，测试速度8frames/sec，训练速度3frames/sec。提出了使用CRF(conditional random fields)来获得物体的准确边界。除此之外，通过对VGG-16最后两个pooling+conv使用多孔卷积(A’trous conv/Dilated conv)，使特征图大小×4后，使用双线性插值使大小×8，解决了如何得到dense的特征图的问题。相较于FCN通过额外训练分类器，再做上采样卷积将特征图映射到原图大小的方式，不必训练上采样卷积核，显著减少了模型复杂度，将训练时间由几天缩短到10个小时。通过将第1个全连接层使用的7\*7\*channel\*4096的卷积核，简单抽取变成4\*4 (or 3\*3)\*channel\*4096，使得感受野从224\*224变成128\*128，该层的计算时间加速到原来的2~3倍。把全卷积层的channel从4096降到1024，在未牺牲性能的情况下，显著降低了计算时间和内存消耗。

2015年6月ParseNet提出为了将每一个像素进行准确分类，在像素点处获取全局信息（整张图片的信息，大感受野）非常重要。通过添加图像特征，即global average pooling+ norm\_scale（对feature map的每个通道，除L2 norm，乘常数，如10，之后通过反向传播优化）+unpooling，可以显著提升FCN的性能（5% better ，64.96 vs. 59.80）。

2016年6月DeepLab v2在VOC20P12上将mIOU刷到了79.7%。为了准确分割不同尺度的物体，首先考虑了使用同一网络对不同缩放比例的图片进行处理，然后使用双线性插值获得与原图等大的多张特征图，通过取最大值获得某一像素是某一类别的概率。这样虽然提高了检测精度，但是增大了计算复杂度。继而提出了ASPP(Atrous Spatial Pyramid Pooling)，通过把pool5后的全连接层改为采用不同rate（6,12,18,24）的3\*3卷积核进行多孔卷积获得4张7\*7\*1024概率图，在分类结果处按channel叠加得到7\*7\*4096。

2017年3月GCN在VOC2012上将mIOU刷到了82.2%。在大家普遍使用3\*3或1\*1的小卷积核时，提出Large Kernel在语义分割领域具有更好的性能。通过使用Large Kernel来增强物体中心像素点的分类能力，BF来增强物体边缘像素点的定位能力，消除了“分类需要全局信息，分割需要局部细节”的矛盾。对于16\*16的特征图，使用15\*15的卷积核进行卷积得到的效果比在这之下的小卷积核更好。由于大卷积核带来的时间和内存消耗较大，因此对于k\*k的卷积使用k\*1-1\*k+1\*k-k\*1共4次卷积来代替。相较于通常的k\*k卷积，虽然在k比较小的时候性能差了一点，但是在k比较大的情况下，通常的k\*k卷积用到的参数多容易过拟合出现了性能下降，而4k卷积的得分一直在上升。另一个k\*k卷积的替代方案是连着使用多个3\*3的卷积，但是效果不够好。



2017年7月Deep lab v3在VOC2012上将mIOU刷到了85.7%。在检测不同尺度目标的问题上，总结了4种方法：图像金字塔虽然能够带来性能上的提升，但是由于GPU显存有限，如果网络较大较深，通常只在预测的阶段使用；编码器-解码器（FCN使用）首先通过卷积和池化减小特征图的尺寸，获得大范围上的类别信息，然后通过转置卷积恢复物体的细节，区分哪一个像素是属于哪一类的；多孔卷积通过使用不同的atrous rates来获得long-range information；空间金字塔池化（SPP)通过使用不同尺度的池化来获得不同尺度上的信息。相较于Deep lab v2，使用ResNet作为基础网络；在ASPP部分，增加了1\*1的卷积（也许是在Large Kernel的启发下，1\*1卷积与多孔卷积在感受野为整张特征图的情况等价），增加了图像特征（ParseNet提出），使用了Batch normalization；不再使用CRF作为后端处理。