训练结果：



心得体会：

X8的模型结构和提供的代码里x16的较为类似，根据论文的图改了一下参数即可。

虽然代码简单，但实际运行的时候却遇到了两个麻烦：

NotFoundError (see above for traceback): Tensor name "fcn/conv\_pool3/biases" not found in checkpoint files ./vgg\_16.ckpt

读取ckpt的时候会报错，一开始谷歌寻找答案，主要的说法是保存模型的TensorFlow版本与读取的不一致，但很快就排除了这个问题。

最后发现是由于上采样时的name='pool5/t\_conv\_x2'这个参数设置问题。

tensorflow.python.framework.errors\_impl.OutOfRangeError: RandomShuffleQueue '\_1\_shuffle\_batch/random\_shuffle\_queue' is closed and has insufficient elements (requested 16, current size 0)

是由于图片过大导致的，原因在于处理数据时，一开始我没有使用下面的判断

if height < vgg\_16.default\_image\_size or width < vgg\_16.default\_image\_size:

return None

对于8x的理解：

由于直接对最终结果做32x上采样的话，由于池化层过多，会损失非常多的信息，导致最后的预测结果十分平滑。但如果不用多层池化，对资源的消耗也会非常大。

因而改用利用第三、四、五层池化后的结果进行上采样、叠加的方式，让最后的结果只需要8倍上采样即可恢复到原图大小。

这个过程中，3、4层池化后的结果直接进行预测，虽然准确度可能会比logits低，但蕴含更多信息，将三层的结果组合后边缘轮廓会更具体一些。

但有一点不理解，3、4层的结果是直接进行一次1x1的卷积作为该层的logits，并与其它层相加，在最后也有计算cross\_entropy\_loss，但似乎并没有进行反向传播，权重也都是用0初始化，那么这个logits是如何生效的？