## 本地

### 1.下载数据集

<http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html#devkit>

### 2.训练数据准备

**convert\_fcn\_dataset.py**

python convert\_fcn\_dataset.py --data\_dir=F:\data\VOCdevkit\VOC2012 --output\_dir=./

Q1：ModuleNotFoundError: No module named 'cv2'

A1：查看博客<https://blog.csdn.net/zstarwalker/article/details/72855781>解决了该问题

Q2：Traceback (most recent call last):

File "convert\_fcn\_dataset.py", line 112, in <module>

tf.app.run()

File "F:\Anaconda3\envs\tensorflow\lib\site-packages\tensorflow\python\platform\app.py", line 126, in run

\_sys.exit(main(argv))

File "convert\_fcn\_dataset.py", line 107, in main

create\_tf\_record(train\_output\_path, train\_files)

File "convert\_fcn\_dataset.py", line 77, in create\_tf\_record

tf\_example = dict\_to\_tf\_example(img\_path,label\_path)

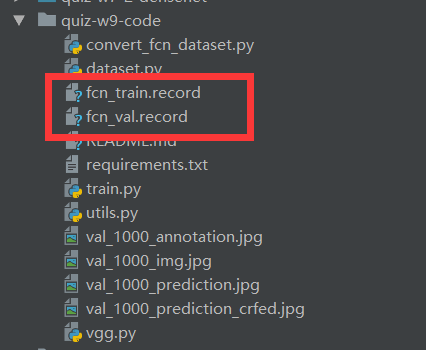
File "convert\_fcn\_dataset.py", line 64, in dict\_to\_tf\_example

'image/filename': bytes\_feature(data['filename'].encode('utf8')),

TypeError: string indices must be integers

A2：filename逻辑处理有问题

运行成功后，在同级目录下生成**fcn\_train.record**,**fcn\_val.record**



### 3.预训练模型

<https://github.com/tensorflow/models/tree/master/research/slim>

下载VGG16 vgg\_16\_2016\_08\_28.tar

解压：

tar -xvf vgg\_16\_2016\_08\_28.tar

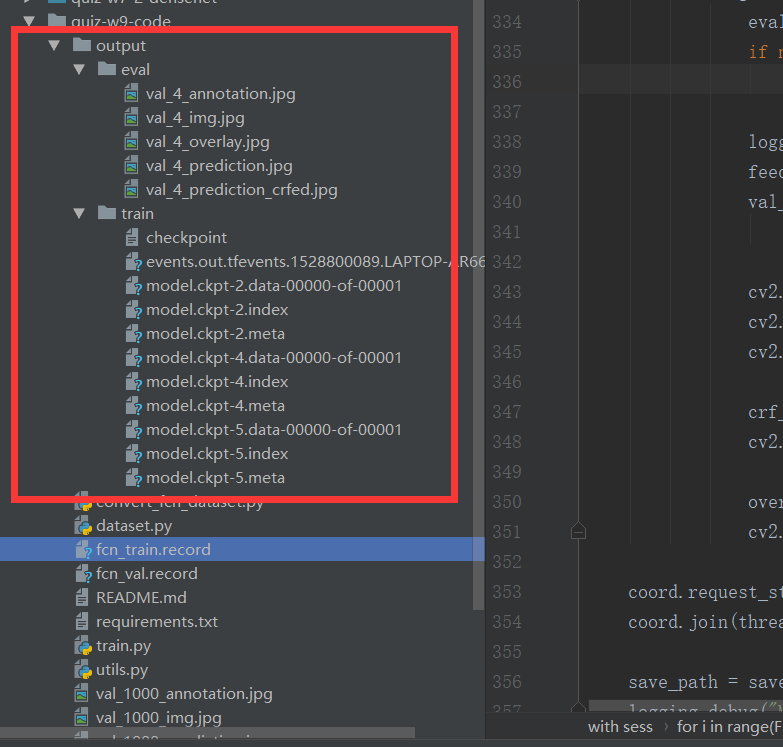
### 4.训练

python train.py --checkpoint\_path ./vgg\_16.ckpt --output\_dir F:\code\CNN\quiz-w9-code\output --dataset\_train ./fcn\_train.record --dataset\_val ./fcn\_val.record --batch\_size 16 --max\_steps 2000

**Q3**：ModuleNotFoundError: No module named 'pydensecrf'

**A3**: conda install pydensecrf

先在本地跑5个step，确定代码正确，会在该目录output下生成trian和eval文件夹



### 5.本地效果如下：原图



**标签**



**预测**



**CRF之后的预测**

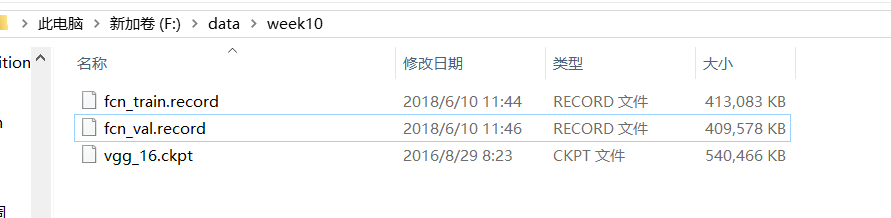


运行只有5个step，还是可以看出有点效果，有基本轮廓，现用tinymind运行1500step

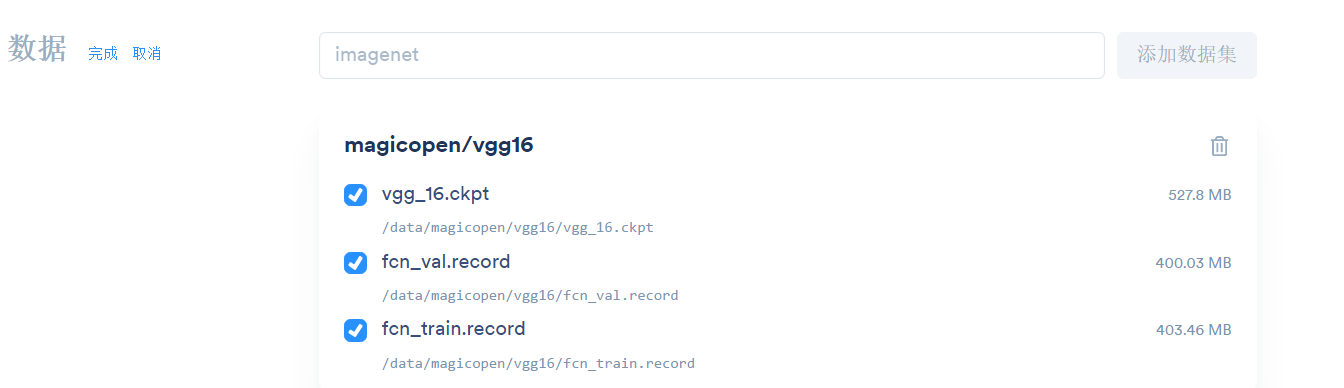
## tinymind

### 1.上传数据集

本地已准备好vgg\_16.ckpt,fcn\_train.record,fcn\_val.record



但tinymind上传400M的文件几个小时了，还没上传成功，没办法只能用别人上传好的数据集

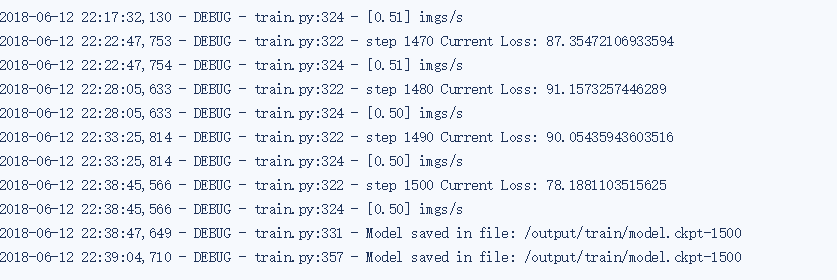


### 2.新建模型关联github，新增参数，新建运行

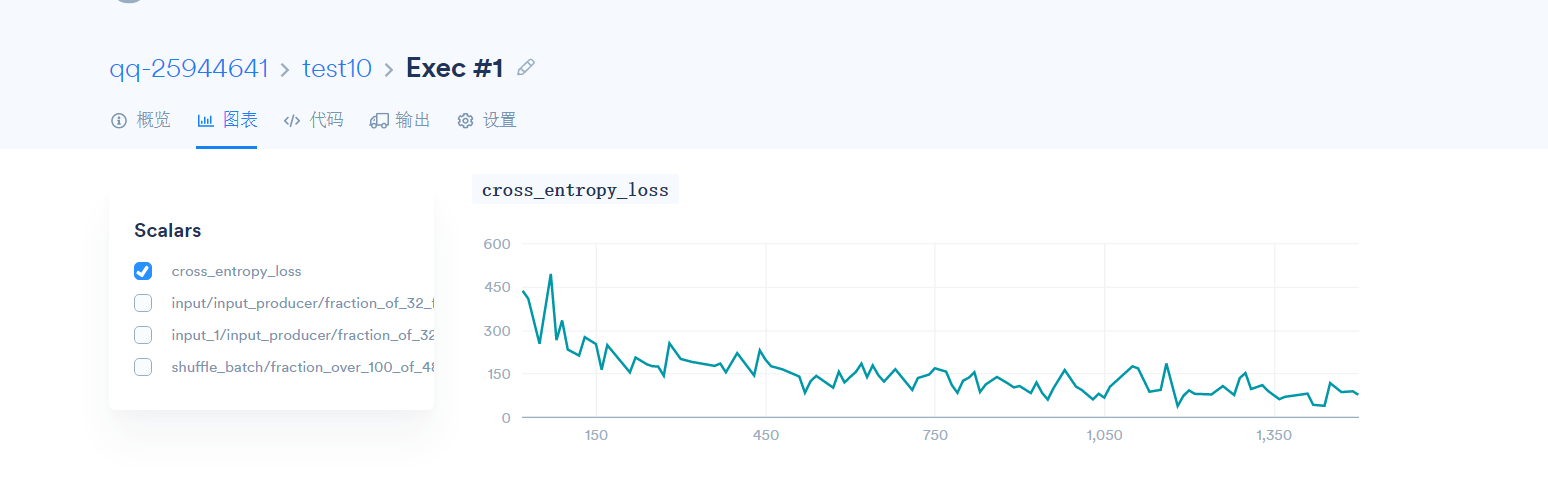
运行地址：

<https://www.tinymind.com/executions/rspt34v8>

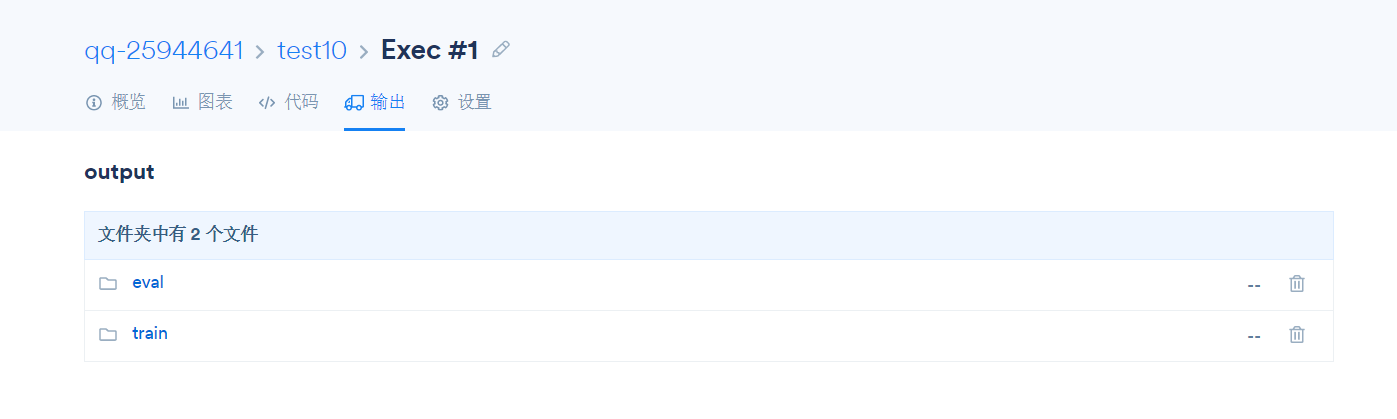
训练1500次截图



图表



输出结果



训练1400次验证结果

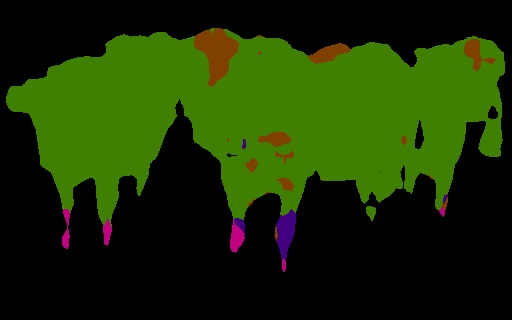
**原图**



**标签**



**预测**



**CRF之后的预测**



效果不是很好，羊的腿没有分割出来，上半身基本训练出来了

**查资料觉得可能原因是FCN有两个缺点**

1.得到的结果还是不够精细。进行8倍上采样虽然比32倍的效果好了很多，

但是上采样的结果还是比较模糊和平滑，对图像中的细节不敏感。

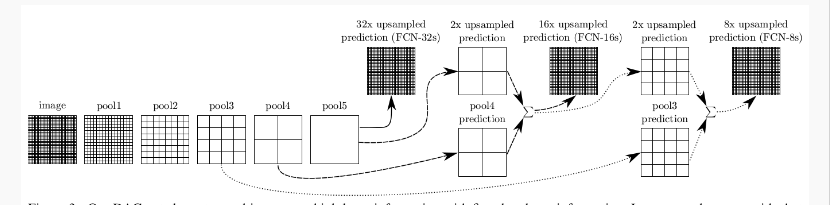
2.是对各个像素进行分类，没有充分考虑像素与像素之间的关系。

忽略了在通常的基于像素分类的分割方法中使用的空间规整（spatial regularization）步骤，缺乏空间一致性。

### 3.FCN 8x代码实现

**FCN原理**

**一句话概括就是**：FCN将传统网络后面的全连接层换成了卷积层，这样网络输出不再是类别而是 [heatmap](http://blog.csdn.net/zizi7/article/details/49506979)；同时为了解决因为卷积和池化对图像尺寸的影响，提出使用上采样的方式恢复



train.py已经实现了FCN16x，这里添加8x的相关实现，具体见train.py

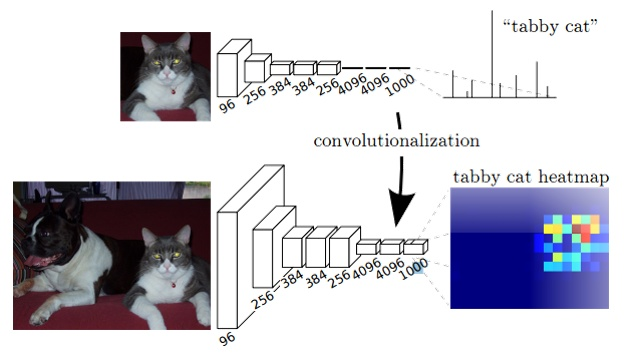
|  |
| --- |
| #从end\_points获取pool3后的输出  pool3\_feature = end\_points['vgg\_16/pool3']  #进行1\*1的卷积  with tf.variable\_scope('vgg\_16/fc8'):  aux\_logits\_8s = slim.conv2d(pool3\_feature, number\_of\_classes, [1, 1],  activation\_fn=None,  weights\_initializer=tf.zeros\_initializer,  scope='conv\_pool3')  #将16x的输出进行2x转置卷积，得到与pool3输出size同样大小  upsampled\_logits = tf.nn.conv2d\_transpose(upsampled\_logits, upsample\_filter\_tensor\_x2,  output\_shape=tf.shape(aux\_logits\_8s),  strides=[1, 2, 2, 1],  padding='SAME')    #16x进行2x转置卷积后和pool3输出相加  upsampled\_logits = upsampled\_logits + aux\_logits\_8s  #获取8x大小  upsample\_filter\_np\_x8 = bilinear\_upsample\_weights(upsample\_factor,  number\_of\_classes)    upsample\_filter\_tensor\_x8 = tf.Variable(upsample\_filter\_np\_x8, name='vgg\_16/fc8/t\_conv\_x8')  #将汇总的pool3进行8x，下面upsample\_factor=8  upsampled\_logits = tf.nn.conv2d\_transpose(upsampled\_logits, upsample\_filter\_tensor\_x8,  output\_shape=upsampled\_logits\_shape,  strides=[1, upsample\_factor, upsample\_factor, 1],  padding='SAME') |

**总结**

图像语义分割，简而言之就是对一张图片上的所有像素点进行分类

FCN实现主要用到了三种技术

**1.全卷积**



分类使用的网络通常会在最后连接全连接层，最后输出一个标量，

FCN将最后三层全连接全部换成卷积层，最后生成heatmap

**2.上采样**

在一般的CNN结构中,如AlexNet,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;而在ResNet中，某些卷积层也参与到缩小图片size的过程。我们需要得到的是一个与原图像size相同的分割图，因此我们需要对最后一层进行上采样，

**3.跳跃结构**

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。

这种结构有点像densenet，输出特性不够好时可以综合前面层的特征来得到更好的结果

**心得体会**

最近几个星期已经学了图像分类，检测，分割，模型训练，验证等知识，对各个算法模型现在也只能简单了解其原理，能够根据作业提示跑跑模型，每个知识点都有一系列的论文，东西太多了想要精通真的好难！！！