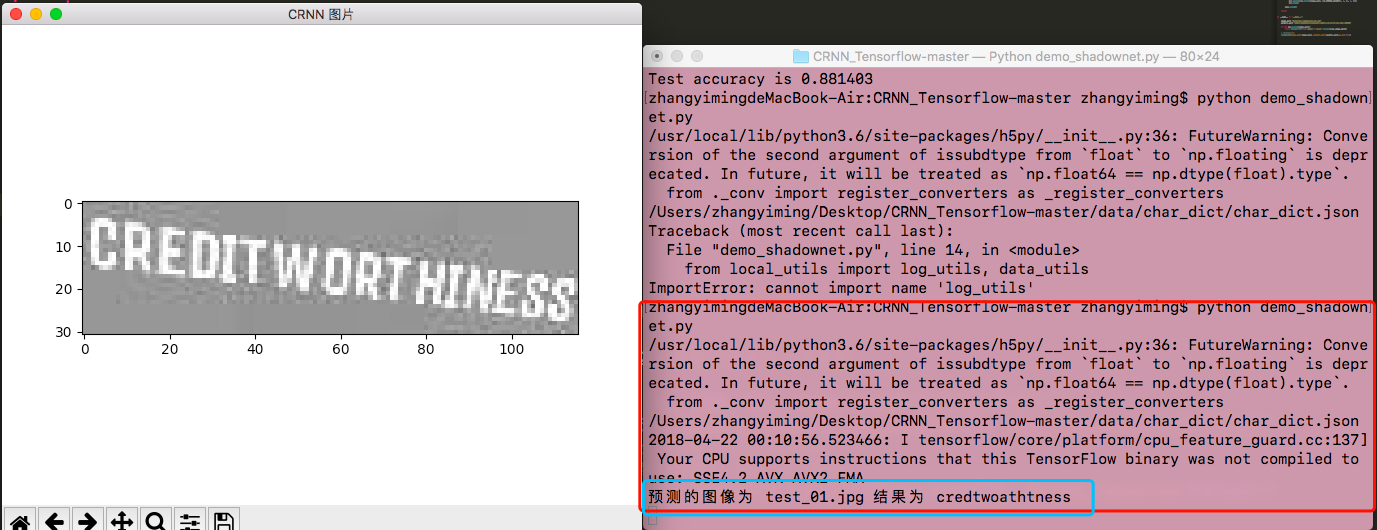
这个项目的论文地址：<http://arxiv.org/abs/1507.05717>

CRNN\_Tensorflow-master文件夹下的四个py文件：

1)demo\_shadownet：一个小demo，执行它会测试在CRNN\_Tensorflow-master/data/test\_image下的图片，如果你想更改测试的图片，在demo\_shadownet.py中修改文件名：

57行 ：image\_path='data/test\_images/test\_01.jpg'

weights\_path='model/shadownet/shadownet\_2018-4-21-11-47-46.ckpt-199999'这是训练好的模型路径，如果你有新的模型，更改它就好了。



2) test\_shadownet：这是测试批量的数据，获得这个模型的准确率。

dataset\_dir='data/test\_feature.tfrecords'#这个.tfrecords文件批量存储了测试数据包括图像和对应的label

weights\_path='model/shadownet/shadownet\_2018-4-21-11-47-46.ckpt-199999'

运行结果：



3) train\_shadownet：这是训练模型的文件，

dataset\_dir='data/sample/train\_feature.tfrecords'#这是训练的数据地址

weights\_path='model/shadownet/shadownet\_2018-4-21-11-47-46.ckpt-199999'#这个是训练好的模型。如果你在：

train\_shadownet(dataset\_dir, weights\_path)这个函数中输入weights\_path，那么会在这个模型的基础上继续训练，就是将原来保存的网络权重载入，继续训练调整权重。

训练的数据太大了，我没办法传给你，而且我训练好了，你也不用训练了吧。

这里就不贴图了。

4) write\_text\_features：这是将你的训练或测试数据写入.tfrecords文件，方便网络载入数据。

dataset\_dir=path+'data/sample'#读取test和train图像和注释的目录

save\_dir=path+’data/sample’#要保存的.tfrecords文件目录

以上四个文件中你可能需要改动的只会在最下面的main函数中。

执行这四个文件记得要在终端中cd到CRNN\_Tensorflow-master文件夹下！！

各个文件夹的 说明：

crnn\_model：这里创建了网络

cnn\_basenet.py这个文件定义了一些基本的卷积操作，其实他利用现有的库就可以很好的完成，不知道为什么作者非要重新定义一遍，这个项目里好多地方都是，明明有很好的库，非要自己写，晦涩难懂，感觉很没必要。

crnn\_model这个文件设计里完整的网络结构，其实并不复杂，在程序中需要注意的我都在注释中说明了。具体的模型设计，你需要看论文。

data文件夹：所有的数据都在这里

char\_dict

sample:训练的样本

test\_images：demo\_shadownet.py文件使用的测试图片

test\_feature.tfrecords：上面说了干嘛的

global\_configuration:模型训练时使用的一些参数，在这里定义他们，方便更改

local\_utils：对.tfrecords文件数据的读写操作

model:保存的训练好的模型在这里