TensorFlow 人脸识别网络

写在前面的话

本次文章坑挖的有些大，有些很不好写，想了想其实人脸识别网络大约也是一个简单的前馈神经网络。但是这么说又没有神秘感，要是要用RNN模型又有些高射炮打蚊子。所以准备介绍介绍人脸识别是怎么回事。

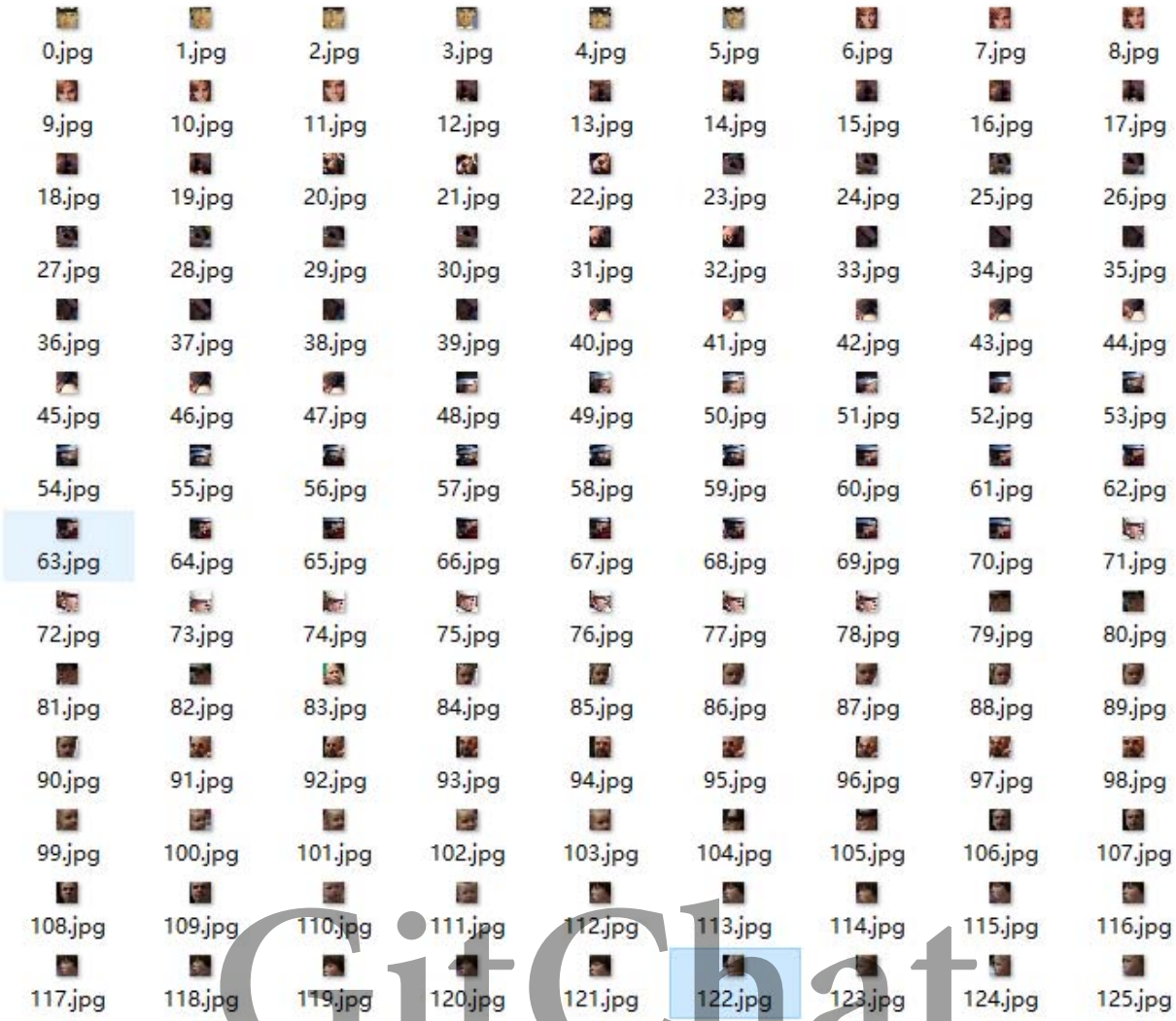
这里挂一漏万，从人脸监测开始，那么什么是人脸检测呢：

人脸识别任务

其实总结起来人脸识别工作可以分为几个部分**人脸检测**（ detection ），**人脸校准**（ aliment ），**人脸识别**（ recognise ）。还记得手机照相过程那个小框框吧，那个就属于人脸检测，有个非常有名的是MTCNN，名字比较唬人，其实就是三个简单的卷积神经网络综合起来的。第一个用于大致确定人脸的位置，第二三个用于精确的识别、定位人脸。这是一种效率与精度兼顾的方式，因为第一层网络参数量少，计算速度快，识别后把人脸截出来需要做精确处理的时候再去多加点参数。

人脸识别recognise的话这个就复杂了，就是通过图片和视频去鉴别不同的人。这个需要更细节的一些东西，相比较而言应用范围也广得多，比如用来识别身份证上不同的人，比如某个AI智障机器人用来提供个性化服务。从人脸监测开始：

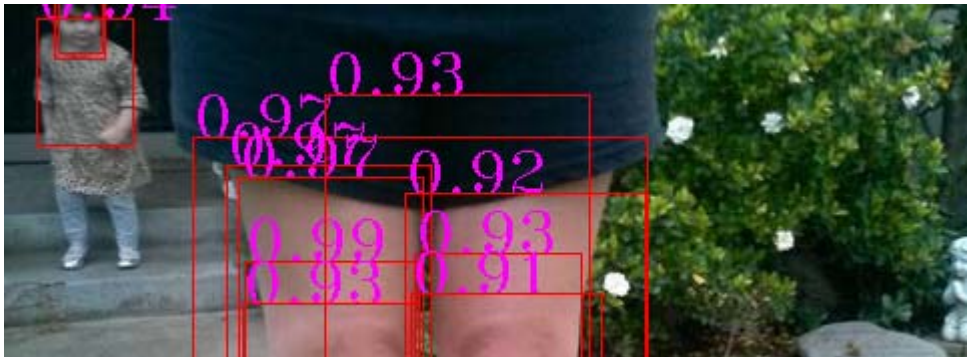
MTCNN



大体上就是一些人脸，是不是感觉图片很小？因为只有12×12像素，当然还有负面样本，就是不是人头像的，用来判别，都说现在的机器学习是在模仿人的认知，但是我作为人，感觉还是看不清，所以机器识别起来也会有困难，但是参数少速度快。

这个loss函数选取平平无奇，对于是否为人脸选择的$loss_1$函数为交叉熵，而对于人脸的box和landmark则用点的距离或者说二范数来表示$loss_2, loss_3$。这三个函数在训练过程中所对应的权值是不同的，PNet、RNet中$loss_3$权值比较低，而最后一个ONet中$loss_3$的权值进行了提高。这个意思是ONet中更侧重于人脸特征点的识别。

可以看到效果很明显，每个人脸都识别出来了，这个是做后续处理的基础，也就是不漏报。你第一层就有几个人脸没有拾取到是不行的，后续只能越来越少。但是出现了一个严重的问题就是膝盖也被识别成了人脸。之于膝盖为什么会被识别成人脸，用下图解释一下：

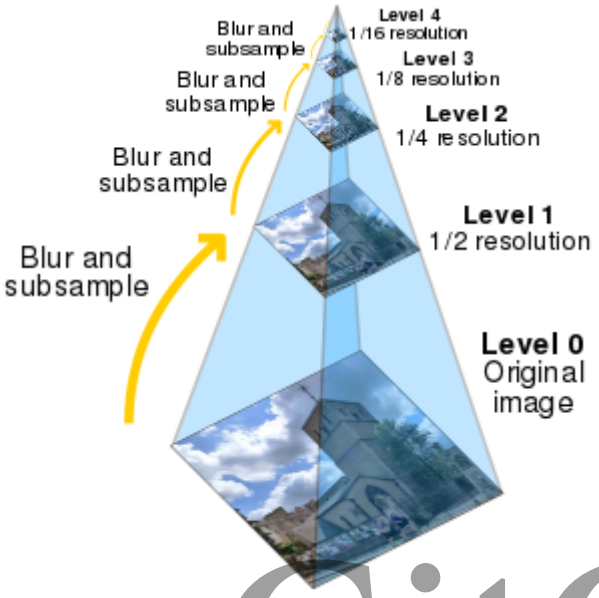


始出现了全链接网络，所以不能处理任意大小图片，需要利用PNet所截取的图片：

可以看到绝大部分膝盖的错误预测已经没有了。剩下的一个手那个地方的可信度也不高。再来看看比较像人脸的膝盖：

个最好的。具体的算法步骤是首先从堆叠的框中按照概率从大到小排序，从第一个开始，如果之后box的重叠面积与前一个超过一个设定值，那么就把其从box序列中删掉。之后再从box序列(删减后)中选择第二个重复上面的过程，通过这个过程来合并人脸识别的框(box)。

还有一个小的补充，在识别过程中由于人脸大小不一样，但是我们的MTCNN网络结构要求人脸大小不能有太大的变化，因此对于图片需要进行多个尺度的变换，这个图节选自维基百科体面的人都会给出引文地址。



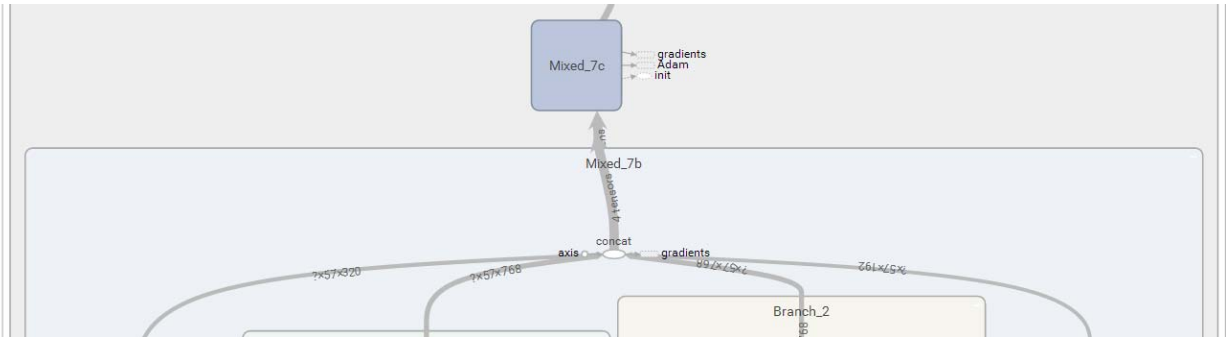
到此为止，内容已经介绍的差不多了，还差最后一步，直接输出最终结果，这个是通过所谓的ONet，其实它依然是一个比较浅的卷积神经网络。但是最后一层为一个全链接层。这个全链接层使得无法如第一个全卷积神经网络那样来处理任意大小的图片(用现有库的话肯定不成)，来看看最终的结果：

由于图片人脸清晰，使得识别效果很好，不重不漏，再来一张：

前面MTCNN的结构已经可以对拾取图片中的人脸了，那么接下来的工作就是区分每个脸是谁，这就是所说的recognise。当然我们需要用到MTCNN所截取的人脸。用于人脸ID这个网络就复杂了，因为网络层数很多，常用的一些结构包括Inception，ResNet都是一种很深的网络结构，而TensorFlow包含一个Inception网络实现：

```
Conv2d_1a_3x3
Conv2d_2a_3x3
Conv2d_2b_3x3
MaxPool_3a_3x3
Conv2d_3b_1x1
Conv2d_4a_3x3
MaxPool_5a_3x3
Mixed_5b
Mixed_5c
Mixed_5d
Mixed_6a
Mixed_6b
Mixed_6c
Mixed_6d
Mixed_6e
Mixed_7a
Mixed_7b
Mixed_7c
```


这个mix代表一系列网络结构的综合，之于**文章在**这里，有兴趣的可以翻阅。可以用tensorboard点开一个节点看看：



输出是一个几百个长度的向量谷歌的facenet的向量长度是128，整个过程类似于将一个图片压缩成一个定长的向量，这是一个信息压缩的过程，如果两张人脸的向量二范数很小的话那么就说明很大可能是同一个人脸。具体例子就不演示了，训练起来实在是挺麻烦。当然实现并不麻烦TensorFlow已经实现了，麻烦是数据寻找和处理，网上开源数据库一大堆，数据量不大几百万个图片吧。

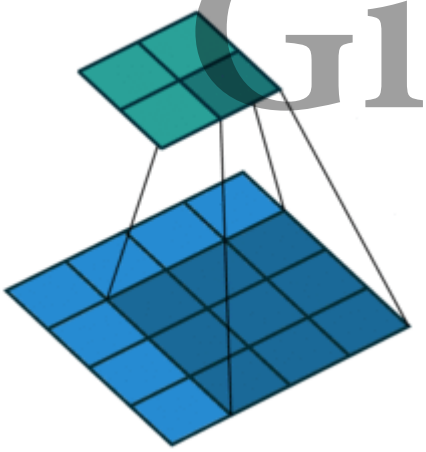
GAN

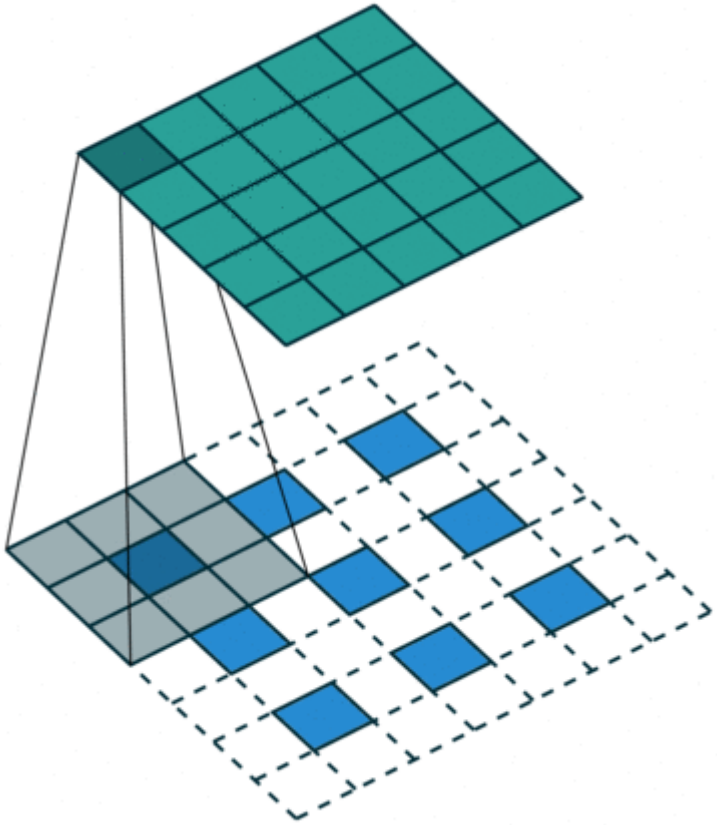
这是附加内容

最后来聊聊老夫感觉挺神奇的一个应用就是对抗生成网络，**有些英语不好的人喜欢叫GAN网络**，这个网络结构并不复杂，但是可以完全用神经网络来生成图片。也就是给一些随机数列，自动生成需要的图片，当然可以用来开车，这完全取决于你的猥琐程度。这个网络中很重要的一个函数就是：

```
tf.nn.conv2d_transpose
```

执行的作用与卷积相反，传统卷积是这样的：





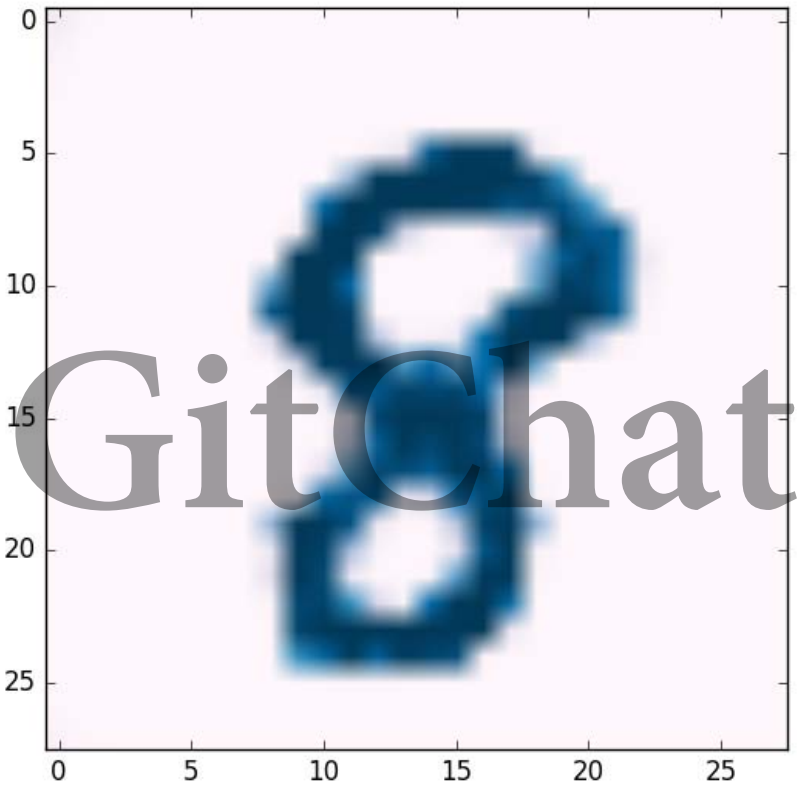
这两张图片都节选自开源项目体面人抄袭都会给链接。简单点说就是把一张图片像素数提高了。

有些宅用这些函数可以画漫画：

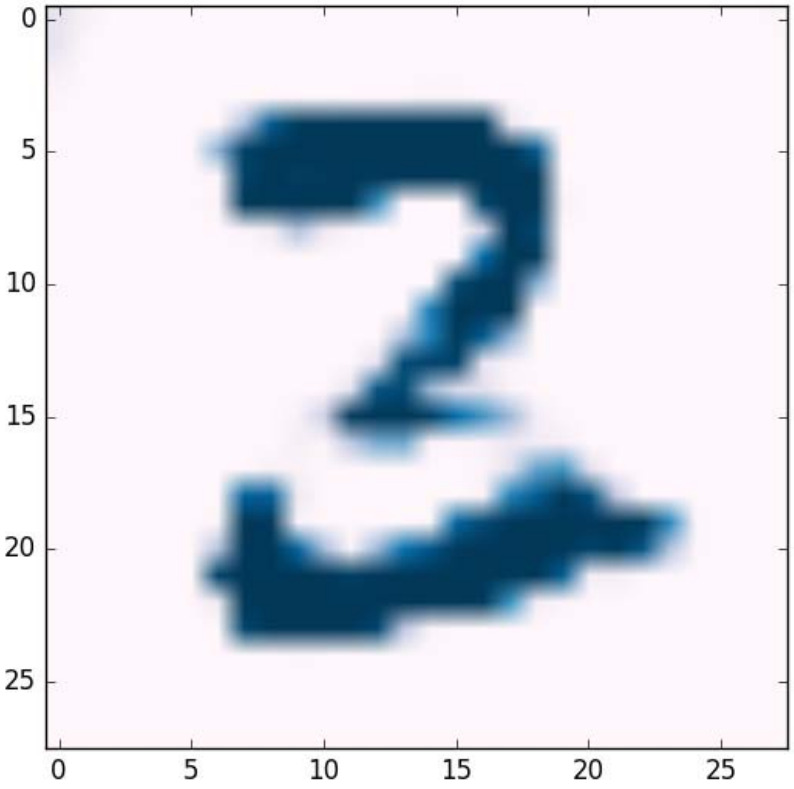
还发了文章，文章名字叫：Learning to Simplify: Fully Convolutional Networks for Rough Sketch Cleanup。图片就是节选自这个文章。

接着说我们的GAN网络，其分为两个部分，第一部分为生成器，第二部分为判别器：生成器用于生成图片以骗过判别器，而判别器用于判别图片是否为生成器所生成，于是二人产生了一种博弈。
生成器将随机噪声生成图片，用到的就是conv2d_transpose函数。其实可以看成是判别器网络倒过来，而判别器是几层卷积网络，最后有一个输出，当图片是真实图片输出1，当为生成器生成的图片输出为0。

最后来看一看结果：



效果还可以，但是给多个标签，神经网络就会懵逼。



比如同时给了两个标签2，3就会这样。其实可以看成2、3两个数字的合体。