原 tesseract-ocr使用以及训练方法

2017年09月16日 15:50:11 zhou_zhu 阅读数 13406 更多

本人最近在做字符识别，所以自行在网上寻找方法，接触到tesseract，自己按照网上方法做的时候，也遇到一些问题，解决了一些。所以我决定写一个博客，一后查看，更新学习。二是方便和网友交流学习。

Tesseract介绍

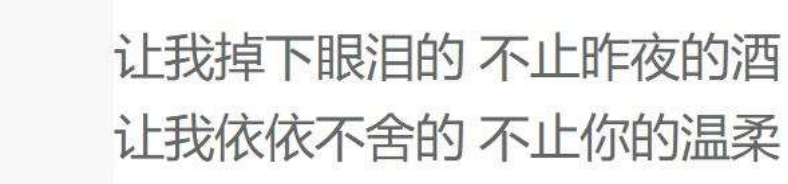
Tesseract是一个开源的OCR（Optical Character Recognition，光学字符识别）引擎，可以识别多种格式的图像文件并将其转换成文本，目前已支持60多种语言（包括文）。 Tesseract最初由HP公司开发，后来由Google维护，目前发布在Googel Project上。

安装Tesseract，从http://code.google.com/p/tesseract-ocr/downloads/list下载Tesseract，3.01上的版本支持中文。安装后在电脑上会有一个Tesseract-OCR目录，通过tesseract.exe程序就可以对图像的字符进行识别。考虑到万一有人上不了谷歌，这个Tesseract-OCR文件夹我也上传了，地址：点击打开链接。文件夹中除了Tesseract的相一个tesseract-vs2013-include-lib-dll文件，这个是VS2013用来调用API的配置文件，后面的博客会写到。打开如图所示。

doc	2015/9/25 14:38	文件夹	
include	2015/9/29 10:44	文件夹	
java	2015/9/25 14:38	文件夹	
lib	2015/9/29 16:06	文件夹	
tessdata	2016/8/26 10:47	文件夹	
tesseract-vs2013-include-lib-dll文件	2014/8/26 20:23	文件夹	
AAAAA.txt	2016/8/30 10:32	文本文档	2 KB
ambiguous_words.exe	2012/10/27 3:23	应用程序	1,066 KB
classifier_tester.exe	2012/10/27 3:23	应用程序	1,279 KB
cntraining.exe	2012/10/27 3:23	应用程序	602 KB
combine_tessdata.exe	2012/10/27 3:23	应用程序	567 KB
dawg2wordlist.exe	2012/10/27 3:23	应用程序	579 KB
font_properties	2016/8/26 10:57	文件	1 KB
gzip.exe	1997/12/23 16:14	应用程序	90 KB
liblept168.dll	2014/8/24 20:35	应用程序扩展	3,324 KB
liblept168d.dll	2015/9/28 11:19	应用程序扩展	3,249 KB
libtesseract302.dll	2014/8/26 19:41	应用程序扩展	4,262 KB
libtesseract302d.dll	2014/3/22 9:17	应用程序扩展	4,291 KB
mftraining.exe	2012/10/27 3:23	应用程序	930 KB
shapeclustering.exe	2012/10/27 3:23	应用程序	857 KB
tar.exe	2011/8/2 19:03	应用程序	344 KB
tesseract.exe	2012/10/27 3:23	应用程序	2,296 KB
tesseract-vs2013-include-lib-dll文件.r...	2015/9/28 11:14	WinRAR 压缩文件	2,625 KB
unicharset_extractor.exe	2012/10/27 3:23	应用程序	572 KB
Uninstall.exe	2015/9/28 11:16	应用程序	93 KB
wordlist2dawg.exe	2012/10/27 3:23	应用程序	661 KB

使用默认的语言库识别

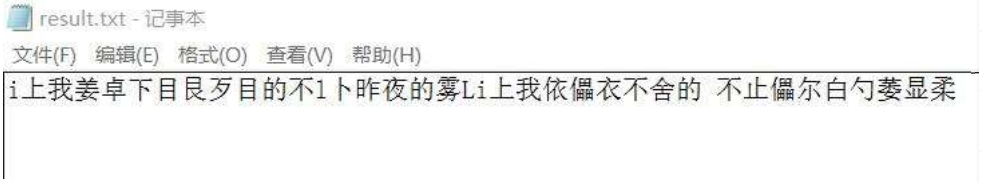
准备一张待识别的图片，我选取一段《成都》的歌词。



接着就可以打开命令行，进入Tesseract-OCR的目录，输入：

```
tesseract.exe gc.jpg result -l chi_sim
```

其中result表示输出结果文件txt名称，chi_sim表示用以识别的语言文件为英文。执行后文件夹中会多一个result.txt。



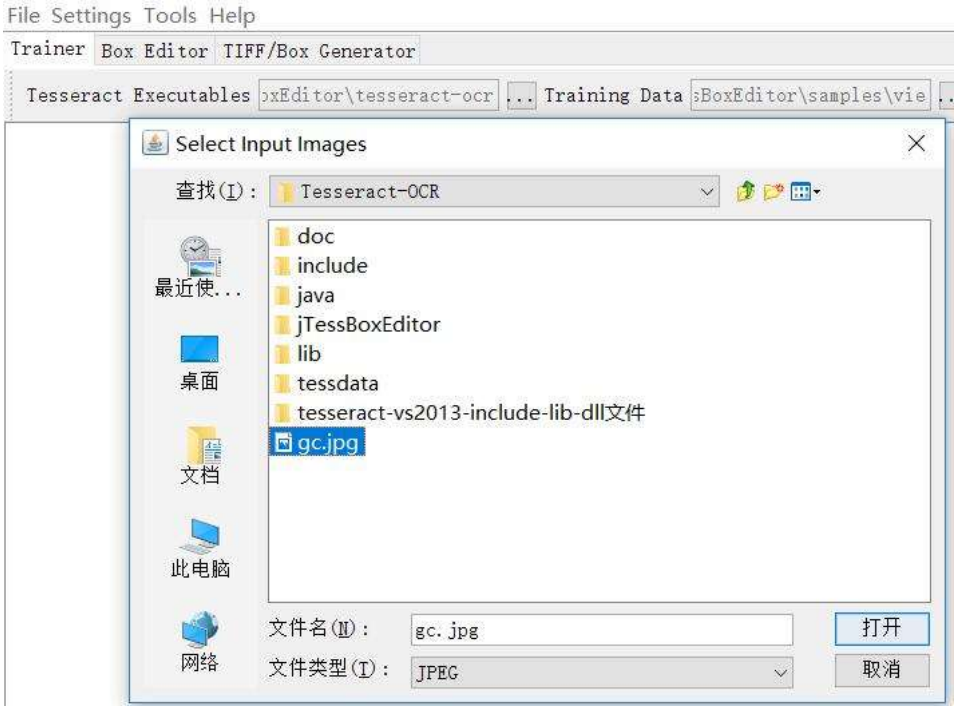
效果非常不好，因为很多汉字是左右结构，比如：眼泪。所以我要自己训练自己的中文库。

训练样本

训练样本需要一个工具，jTessBoxEditor，下载地址：点击打开链接。这个工具是用java开发的，需要jre7以上的版本支持。

1、获取训练的图片，为了方便我使用了原来的图片一张，样本当然是越多越好。

2、合并样本文件，打开jTessBoxEditor，点开train.bat。在菜单栏中Tools->Merge TIFF。在弹出的窗口中可以选择多张样本图片（网上之前有说要.tif格式的图片，测行），我这边就用了一张样本图片。



一张或者多张图片可以合成一张tif文件。

3、生成box文件， 打开命令行，输入：

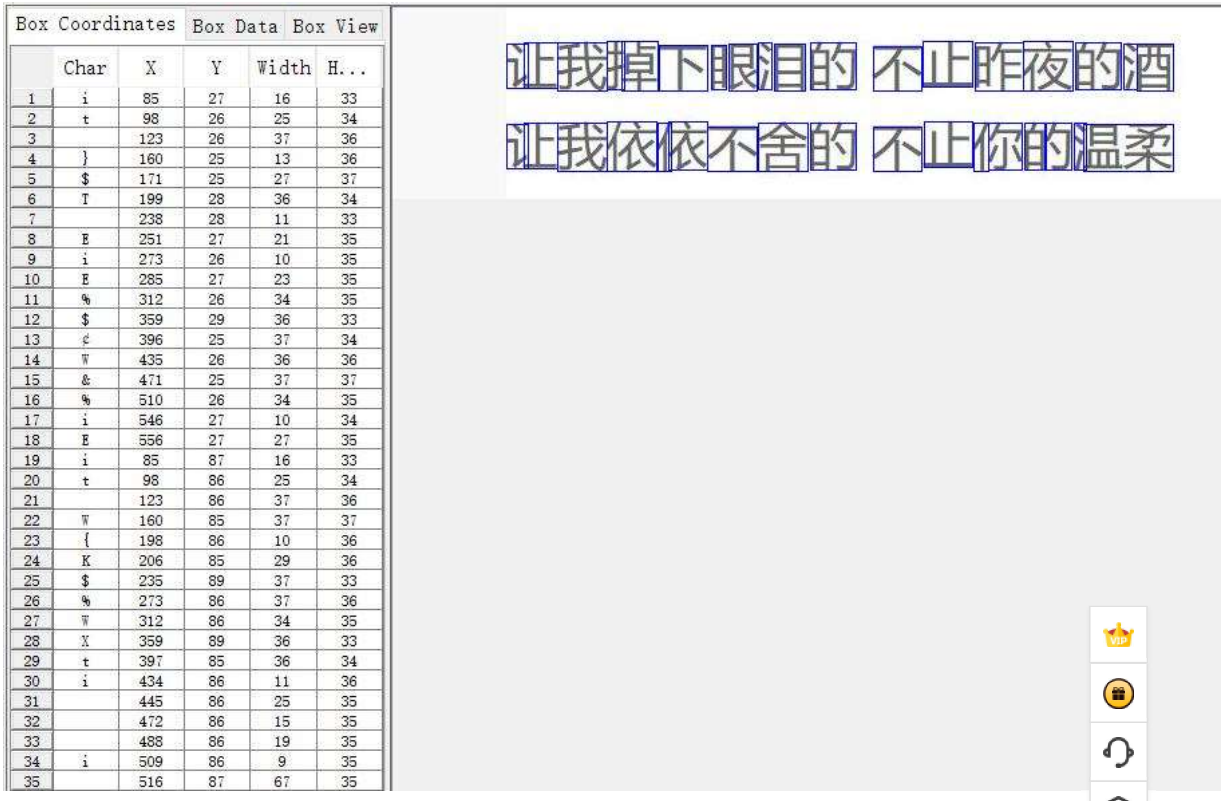
```
tesseract.exe gc.font.exp1.tif gc.font.exp1 batch.nochop makebox
```

生成的BOX文件为gc.font.exp1.box，BOX文件为Tessercat识别出的文字和其坐标。Make BOX的命名的个数为：

```
tesseract [lang].[fontname].exp[num].tif [lang].[fontname].exp[num] batch.nochop makebox
```

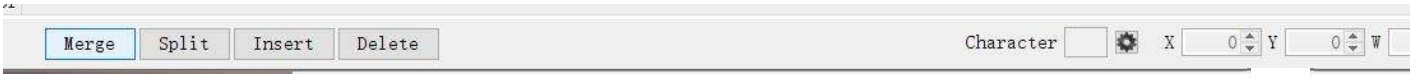
其中lang为语言名称，fontname为字体名称，num为序号，可以随便定义。有些博客说对于这个命名无所谓，但是我尝试到后免出错了，是tr文件名的问题，在下面我图。读者也可以试试，不知是不是我之前步骤哪里做错了。

4、文字矫正，打开jTessBoxEditor工具，打开gc.font.exp1.tif文件（必须将上一步生成的.box和.tif样本文件放在同一目录），如下图所示。可以看出有些字符分割和识别以通过该工具手动对每张图片中识别错误的字符进行校正。校正完成后保存即可。（注：发现中文打不上去，在菜单Setting->Font中可以修改，改为宋体即可）

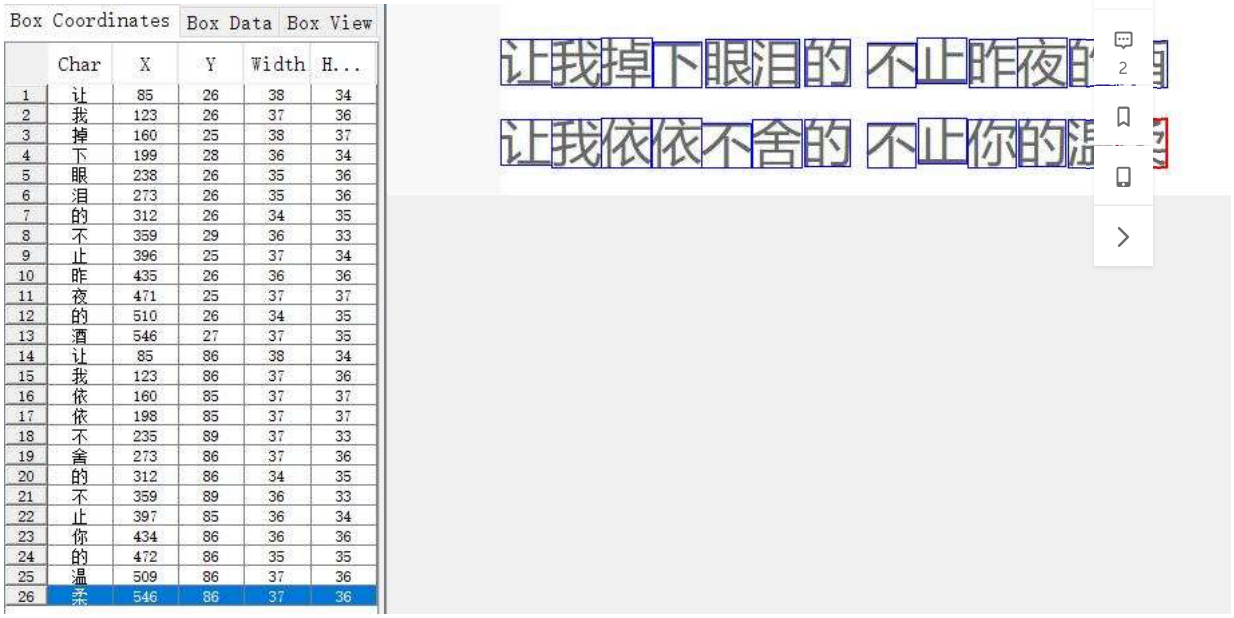


	Char	X	Y	Width	H...
1	i	85	27	16	33
2	t	98	26	25	34
3		123	26	37	36
4	}	160	25	13	36
5	$	171	25	27	37
6	T	199	28	36	34
7		238	28	11	33
8	E	251	27	21	35
9	i	273	26	10	35
10	E	285	27	23	35
11	%	312	26	34	35
12	$	359	29	36	33
13	¢	396	25	37	34
14	W	435	26	36	36
15	&	471	25	37	37
16	%	510	26	34	35
17	i	546	27	10	34
18	E	556	27	27	35
19	i	85	87	16	33
20	t	98	86	25	34
21		123	86	37	36
22	W	160	85	37	37
23	{	198	86	10	36
24	K	206	85	29	36
25	$	235	89	37	33
26	%	273	86	37	36
27	W	312	86	34	35
28	X	359	89	36	33
29	t	397	85	36	34
30	i	434	86	11	36
31		445	86	25	35
32		472	86	15	35
33		488	86	19	35
34	i	509	86	9	35
35		516	87	67	35

对于标定的方框以及识别的字符进行修改。



选择两个或两个以上的框，Merge可进行合并；Split将框进行拆分；Insert插入框，如果图片上一个框也没有，那无法进行插入；Delete删除框。选择修改的字符框中输入想要修改的字，再点击齿轮，即可修改。修改后，如下图所示：



5、生成.tr文件，在命令行中输入：

```
tesseract gc.font.exp1.tif  gc.font.exp1  nobatch box.train
```

6、计算字符集，从生成的box文件中提取，继续输入：

```
unicharset_extractor gc.font.exp1.box
```

7、生成字体特征文件，在当前文件夹中新建任意名称的文件，里面格式为：

```
<fontname> <italic> <bold> <fixed> <serif> <fraktur>
```

例如：我建了一个名为font的文件，里面内容为：font 0 0 0 0 0

这个文件可以是手动生成的txt文件，也可以在在命令行中输入：

```
echo font 0 0 0 0 0 >font
```

即可。

8、特征训练，继续在命令行输入：

```
mftraining -F font -U unicharset gc.font.exp1.tr
```

在这一步我出现了好几个错误，如下图

（1）Failed to load unicharset from file uncharset，这是因为刚刚的font的文件，如果是在txt中写的，一定要写成font.txt，加上后缀。

```
D:\F\Tesseract-OCR\Tesseract-OCR>mftraining -F font -U uncharset mytest.tr
Warning: No shape table file present: shapetable
Failed to load unicharset from file uncharset
Building unicharset for training from scratch...
Failed to load unicharset from file uncharset
Building unicharset for boosting from scratch...
Failed to load unicharset from file uncharset
Building unicharset for boosting from scratch...
Failed to load unicharset from file uncharset
Building unicharset for boosting from scratch...
Failed to load font_properties from font
```

（2）feature training for Tesseract已停止工作。命令行显现：

Reading num.tr ...

Font id = -1/0, class id = 1/13 on sample 0

font_id >= 0 && font_id < font_id_map_.SparseSiz..\..\classify\trainingsampleset.cpp, line 622

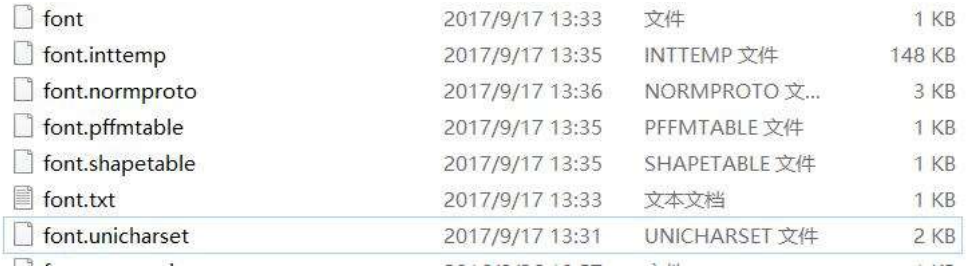
这个问题就是上面命名所导致的，所以还是规范命名。

9、聚集tesseract识别的训练文件，命令行输入：

```
cntraining gc.font.exp1.tr
```

有人会说其他还有一条shapeclustering语句，说下这个步骤可有可无，这个是在3.02中新加的，主要针对印度语，所以我们在做的时候会有一个警告warning No sh present。

这时候文件夹中会多了四个文件，在unicharset，inttemp，normproto，pfftable文件名前面加上font.。如下图所示：



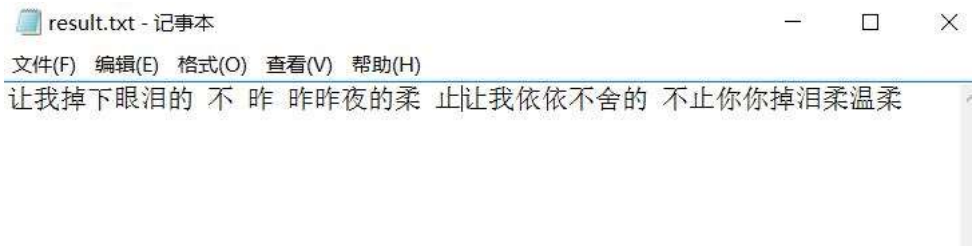
10、最后，合并相关文件，生成字典文件，输入：

```
combine_tessdata font.
```

所有输入命令如下图所示

```
D:\F\Tesseract-OCR\Tesseract-OCR>tesseract gc.font.exp1.tif  gc.font.exp1 batch.nochop makebox
Tesseract Open Source OCR Engine v3.02 with Leptonica

D:\F\Tesseract-OCR\Tesseract-OCR>tesseract gc.font.exp1.tif gc.font.exp1 nobatch box.tr
Tesseract Open Source OCR Engine v3.02 with Leptonica
APPLY_BOXES:
   Boxes read from boxfile:      26
   Found 26 good blobs.
TRAINING ... Font name = font
Generated training data for 4 words

D:\F\Tesseract-OCR\Tesseract-OCR>unicharset_extractor gc.font.exp1.box
Extracting unicharset from gc.font.exp1.box
Wrote unicharset file ./unicharset.

D:\F\Tesseract-OCR\Tesseract-OCR>echo font 0 0 0 0 0 >font

D:\F\Tesseract-OCR\Tesseract-OCR>mftraining -F font -U unicharset gc.font.exp1.tr
Warning: No shape table file present: shapetable
Reading gc.font.exp1.tr ...
Flat shape table summary: Number of shapes = 17 max unichars = 1 number with multiple unichars = 0
Done!

D:\F\Tesseract-OCR\Tesseract-OCR>cntraining gc.font.exp1.tr
Reading gc.font.exp1.tr ...
Clustering ...

Writing normproto ...

D:\F\Tesseract-OCR\Tesseract-OCR>combine_tessdata font.
Combining tessdata files
TessdataManager combined tesseract data files.
Offset for type 0 is -1
Offset for type 1 is 140
Offset for type 2 is -1
Offset for type 3 is 1338
Offset for type 4 is 152560
Offset for type 5 is 152739
Offset for type 6 is -1
Offset for type 7 is -1
Offset for type 8 is -1
Offset for type 9 is -1
Offset for type 10 is -1
Offset for type 11 is -1
Offset for type 12 is -1
Offset for type 13 is 155110
Offset for type 14 is -1
```

最终，在当前目录中会产生一个为font.traineddata文件，将其拷到tessdata文件夹中，再测试一下。



虽然不是全部识别出来，但是较之前的识别率提高了很多，这个和样本数量也是有关系的，而且这句话中左右结构的字特别多，原图26个字，却识别出31个字出来了，还没想到什么方法，单个字训练？。我也试了其他字符训练，效果还可以

这是我第一次写博客，想到哪里写到哪里，如果哪些地方写的不恰当的，还请大神指出来，谢谢。