

笔记本: OCR

创建时间: 2019-05-09 22:05

URL: https://blog.csdn.net/and_bjdbbc/article/details/86234679

jTessBoxEditor训练字库以及字库的合并

2019年01月10日 16:14:31 [and_bjdbbc](#) 阅读数: 440

个人学习使用jTessBoxEditor资料整理, 文章原地址如下:

- 1.<https://blog.csdn.net/zzb4702/article/details/51722942>
- 2.<https://blog.csdn.net/zhanghaiming012/article/details/80522992>
- 3.<https://blog.csdn.net/a745233700/article/details/80175883>
- 4.<https://blog.csdn.net/xiaojun111111/article/details/54377154>

先总结下个人遇到的问题:

1.训练的时候, 工具识别圈中的范围有问题, 可能与识别命令中的 -psm 7 有关, 关于这个参数可以参考上面的第四篇博客, 看一下命令代表的含义, 然后对应调整一下 -psm 后面的参数

1、需要的工具

(1) 安装Tesseract-OCR

可以网上自己查找资源文件, 我装的是tesseract-ocr-setup-3.02.02.exe

配置环境变量Path中加入OCR的根目录路径, 新建变量
TESSDATA_PREFIX, 填入OCR的根目录路径

安装完成后, 打开命令窗, 在命令窗输入tesseract如果出现下面结果就说明安装正确:

```
C:\windows\system32\cmd.exe
Microsoft Windows [版本 10.0.10586]
(c) 2015 Microsoft Corporation。保留所有权利。

C:\Users\ZZB>tesseract
Usage:tesseract imagename outputbase [-l lang] [-psm pagesegmode] [configfile...]

pagesegmode values are:
0 = Orientation and script detection (OSD) only.
1 = Automatic page segmentation with OSD.
2 = Automatic page segmentation, but no OSD, or OCR
3 = Fully automatic page segmentation, but no OSD. (Default)
4 = Assume a single column of text of variable sizes.
5 = Assume a single uniform block of vertically aligned text.
6 = Assume a single uniform block of text.
7 = Treat the image as a single text line.
8 = Treat the image as a single word.
9 = Treat the image as a single word in a circle.
10 = Treat the image as a single character.
-l lang and/or -psm pagesegmode must occur before anyconfigfile.

Single options:
-v --version: version info
--list-langs: list available languages for tesseract engine

C:\Users\ZZB>
```

(2) jTessBoxEditor工具下载

下载地

址: <http://sourceforge.net/projects/vietocr/files/jTessBoxEditor/>

安装包解压后双击里边的“jTessBoxEditor.jar”，或者双击该目录下的“train.bat”脚本文件，就可以打开该工具了。

2、样本图片准备：（进行训练的样本图片数量越多越好）

可以手动刷新某网站验证码⁹⁵⁴⁰，手动或者写程序，

3、使用jTessBoxEditor生成训练样本的的合并tif图片：

(1) 打开jTessBoxEditor，选择Tools->Merge TIFF，进入训练样本所在文件夹，选中要参与训练的样本图片。

注：好多文章写道在这步之前需要将样本图片转为tif格式，使用jTessBoxEditor2.2.0尝试了下提示 couldn't seek 。所以直接用PNG格式执行这步操作。

(2) 点击“打开”后弹出保存对话框，选择保存在当前路径下，文件命名为“zwp.test.exp0.tif”，格式只有一种“TIFF”可选。

tif文面命名格式[lang].[fontname].exp[num].tif

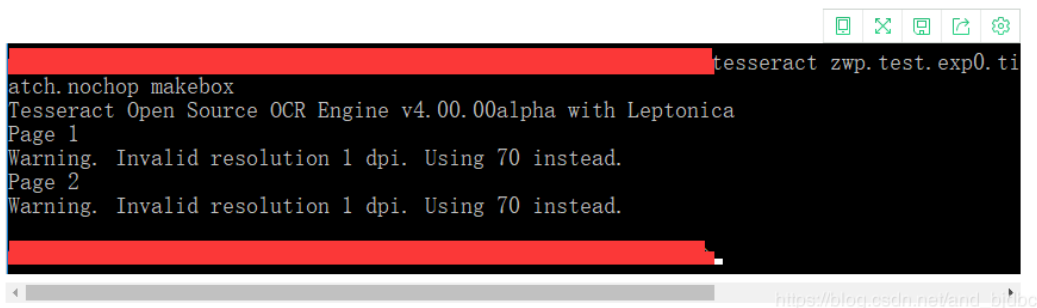
lang是语言，fontname是字体，num为自定义数字。

比如我们要训练自定义字库 zwp，字体名test，那么我们把图片文件命名为 zwp.test.exp0.tif

4、使用tesseract生成.box文件：

在上一步骤生成的“zwp.test.exp0.tif”文件所在目录下打开命令程序（即在cmd中切换盘符到文件目录下），执行下面命令,执行完之后会生成zwp.test.exp0.box文件。

**tesseract zwp.test.exp0.tif zwp.test.exp0 -l chi_sim -psm 7
batch.nochop makebox**



```
tesseract zwp.test.exp0.tif zwp.test.exp0 -l chi_sim -psm 7 batch.nochop makebox
Tesseract Open Source OCR Engine v4.00.00alpha with Leptonica
Page 1
Warning. Invalid resolution 1 dpi. Using 70 instead.
Page 2
Warning. Invalid resolution 1 dpi. Using 70 instead.
```

5、使用jTessBoxEditor矫正.box文件的错误：

.box文件记录了每个字符在图片上的位置和识别出的内容，训练前需要使用jTessBoxEditor调整字符的位置和内容。

打开jTessBoxEditor点击Box Editor ->Open，打开步骤2中生成的“zwp.test.exp0.tif”，会自动关联到“zwp.test.exp0.box”文件，这两文件要求在同一目录下。调整完点击“save”保存修改。

6、生成font_properties文件：（该文件没有后缀名）

（1）执行命令，执行完之后，会在当前目录生成font_properties文件

echo test 0 0 0 0 0 >font_properties

（2）也可以手工新建一个名为font_properties的文本文件，输入内容“test 0 0 0 0”表示字体test的粗体、倾斜等共计5个属性。这里的“test”必须与“zwp.test.exp0.box”中的“test”名称一致。

7、使用tesseract生成.tr训练文件：

执行下面命令，执行完之后，会在当前目录生成zwp.test.exp0.tr文件。

tesseract zwp.test.exp0.tif zwp.test.exp0 nobatch box.train

```
Tesseract Open Source OCR Engine v4.00.00alpha with Leptonica
Page 1
Warning. Invalid resolution 1 dpi. Using 70 instead.
APPLY_BOXES:
  Boxes read from boxfile:      8
  Found 8 good blobs.
Generated training data for 8 words
Page 2
Warning. Invalid resolution 1 dpi. Using 70 instead.
FAIL!
APPLY_BOXES: boxfile line 4/闊?((570,0),(570,0)): FAILURE! Couldn't find a matching b
APPLY_BOXES:
  Boxes read from boxfile:      8
  Boxes failed resegmentation:  1
  Found 7 good blobs.
Generated training data for 2 words
https://blog.csdn.net/and_bjdb
```

8、生成字符集文件：

执行下面命令：执行完之后会在当前目录生成一个名为“unicharset”的文件。

unicharset_extractor zwp.test.exp0.box

```
>unicharset_extractor zwp.t
Extracting unicharset from zwp.test.exp0.box
Wrote unicharset file ./unicharset.
```

9、生成shape文件：

执行下面命令，执行完之后，会生成 shapetable 和 zwp.unicharset 两个文件。

**shapeclustering -F font_properties -U unicharset -O zwp.unicharset
zwp.test.exp0.tr**

10、生成聚字符特征文件：

执行下面命令，会生成 inttemp、pffmtable、shapetable和zwp.unicharset四个文件。

**mftraining -F font_properties -U unicharset -O zwp.unicharset
zwp.test.exp0.tr**

```

C:\Users\ Administrator>mftraining -F font_propert
Read shape table shapetable of 9 shapes
Reading zwj.test.exp0.tr ...
Bad properties for index 3, char 涓? 0,255 0,255 0,0 0,0 0,0
Bad properties for index 4, char 缙? 0,255 0,255 0,0 0,0 0,0
Bad properties for index 5, char 棕? 0,255 0,255 0,0 0,0 0,0
Bad properties for index 6, char 闾? 0,255 0,255 0,0 0,0 0,0
Bad properties for index 7, char 鎬? 0,255 0,255 0,0 0,0 0,0
Bad properties for index 8, char 璫? 0,255 0,255 0,0 0,0 0,0
Bad properties for index 9, char 褰? 0,255 0,255 0,0 0,0 0,0
Bad properties for index 10, char 锄? 0,255 0,255 0,0 0,0 0,0
Bad properties for index 11, char 鍏? 0,255 0,255 0,0 0,0 0,0
Warning: no protos/configs for Joined in CreateIntTemplates()
Warning: no protos/configs for |Broken|0|1 in CreateIntTemplates()
Done!
https://blog.csdn.net/and_bjdb

```

```

C:\Users\ Administrator>mftraining -F font_propert
Read shape table shapetable of 9 shapes
Reading zwj.test.exp0.tr ...
Bad properties for index 3, char 涓? 0,255 0,255 0,0 0,0 0,0
Bad properties for index 4, char 缙? 0,255 0,255 0,0 0,0 0,0
Bad properties for index 5, char 棕? 0,255 0,255 0,0 0,0 0,0
Bad properties for index 6, char 闾? 0,255 0,255 0,0 0,0 0,0
Bad properties for index 7, char 鎬? 0,255 0,255 0,0 0,0 0,0
Bad properties for index 8, char 璫? 0,255 0,255 0,0 0,0 0,0
Bad properties for index 9, char 褰? 0,255 0,255 0,0 0,0 0,0
Bad properties for index 10, char 锄? 0,255 0,255 0,0 0,0 0,0
Bad properties for index 11, char 鍏? 0,255 0,255 0,0 0,0 0,0
Warning: no protos/configs for Joined in CreateIntTemplates()
Warning: no protos/configs for |Broken|0|1 in CreateIntTemplates()
Done!
https://blog.csdn.net/and_bjdb

```

11、生成字符正常化特征文件：

执行下面命令，会生成 normproto 文件。

cntraining zwj.test.exp0.tr

```

C:\Users\ Administrator>cntraining zwj.test.exp0.tr
Reading zwj.test.exp0.tr ...
Clustering ...
Writing normproto ...

```

12、文件重命名：

重新命名inttemp、pffmtable、shapetable和normproto这四个文件的名字为 [lang].xxx。

这里修改为zwj.inttemp、zwj.pffmtable、zwj.shapetable和zwj.normproto

执行下面命令：

```

rename normproto zwj.normproto
rename inttemp zwj.inttemp
rename pffmtable zwj.pffmtable
rename shapetable zwj.shapetable

```

13、合并训练文件:

执行下面命令, 会生成zwp.traineddata文件。

combine_tessdata zwp.

```
>combine_tessdata zwp.  
Combining tessdata files  
Output zwp.traineddata created successfully.  
1:unicharset:size=677, offset=168  
3:inttemp:size=136886, offset=845  
4:pfmtable:size=130, offset=137731  
5:normproto:size=1280, offset=137861  
13:shapetable:size=166, offset=139141
```

Log输出中的Offset 1、3、4、5、13这些项不是-1, 表示新的语言包生成成功。

将生成的“zwp.traineddata”语言包文件复制到Tesseract-OCR 安装目录下的tessdata文件夹中, 就可以使用训练生成的语言包进行

图像文字识别了。

14、测试:

输入下面命令, -l后面为训练生成的语言包。

tesseract test.PNG test -l zwp

使用新训练的语言包进行文字识别后, 会发现之前识别不出来的文字也可以识别出来了。

15、多字库合并

原文地址: <https://blog.csdn.net/zhanghaiming012/article/details/80522992>

首先, 需要 生成的字符集.tif文件, 位置文件 .box ,只要有这两个文件在, 就可以合并字典

好了, 我现在有三个 需要合并的字典 why3 why4 why5, 他他们的名字修改为 name.num 的形式, 分别改为 why.3 why.4 why.5

1、先生成相对应的 .tr 文件

```
tesseract why.3.tif why.3 nobatch box.train  
tesseract why.4.tif why.4 nobatch box.train  
tesseract why.5.tif why.5 nobatch box.train
```

2、从所有文件中提取字符

```
unicharset_extractor why.3.box why.4.box why.5.box
```

3、生成字体特征文件

新建的font文件中 把所有box文件对应的 字体特征都加进去

```
why.4 0 0 0 0 0
```

```
why.3 0 0 0 0 0
```

```
why.5 0 0 0 0 0
```

```
mftraining -F font -U unicharset why.3.tr why.4.tr why.5.tr
```

4、聚集所有.tr 文件

```
cntraining why.3.tr why.4.tr why.5.tr
```

6、重命名文件，我把unicharset, inttemp, normproto, pfftable 这几个文件加了前缀why.

7、合并所有文件 生成一个大的字库文件

```
combine_tessdata why.
```