OCR文本识别

基于传统OCR与神经网络结合

复旦大学计算机科学与技术
16307130335
方煊杰

1

想法来源

- Cs 扫描仪　　上传文件（作业）很方便
- Pdf 论文提取　　PDF 转化为word是个十分常见的功能，有时候做自然语言处理并不想总是去用adobe reader 做预处理，而是希望能够在代码中一起实现

2

步骤

- 拍摄图片预处理
  - 边缘提取，透射变换，锐化增强
- 字符定位与切割
  - 投影算法
- 生成训练数据集
- 卷积神经网络

3

拍摄图片预处理

边缘提取

原图

强度变化

R = G =B

灰度

高斯降噪

核：5 X 5

模糊

中间点”取”周围点”的平均值，就会变成1。在数值上，这是一种”平滑化

适当大小的核

$$Gray(i,j) = 0.299 * R(i,j) + 0.578 * G(i,j) + 0.114 * B(i,j)$$

4

拍摄图片预处理

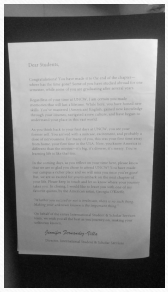
边缘提取

膨胀

轮廓

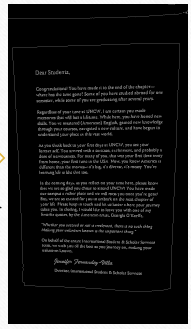
最大矩形

扩大



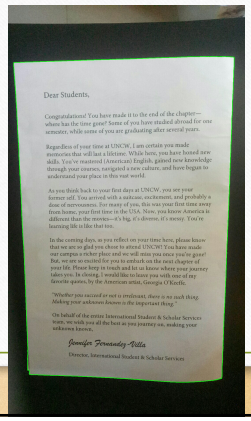
边缘提取

Canny算子



矩形自定义核

找外轮廓



5

拍摄图片预处理

透射算法

- 现在拥有了外轮廓的点集，接下来要做的是通过这个点集，找到该四边形的四个顶点
- 最小外接矩形拟合
- 凸包算法

由于拍照可能存在各种角度,以及背景各式各样，最小外接矩形拟合无法准确的拿到四个顶点进行后续变换

目的为了识别文本，所以要尽可能在预处理时去掉对比强烈的区域

6

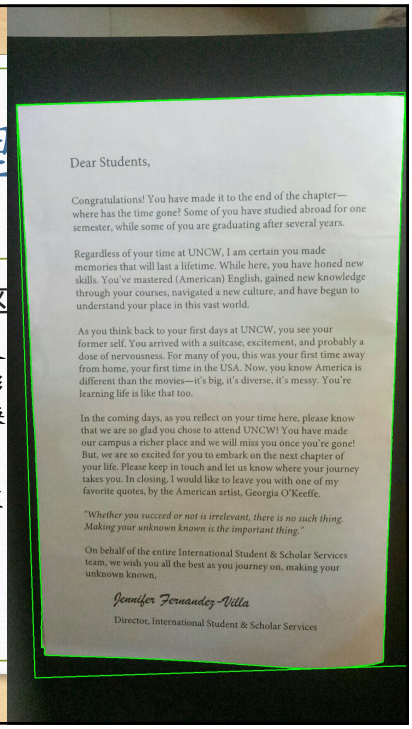
拍摄图片预处理

透射算法

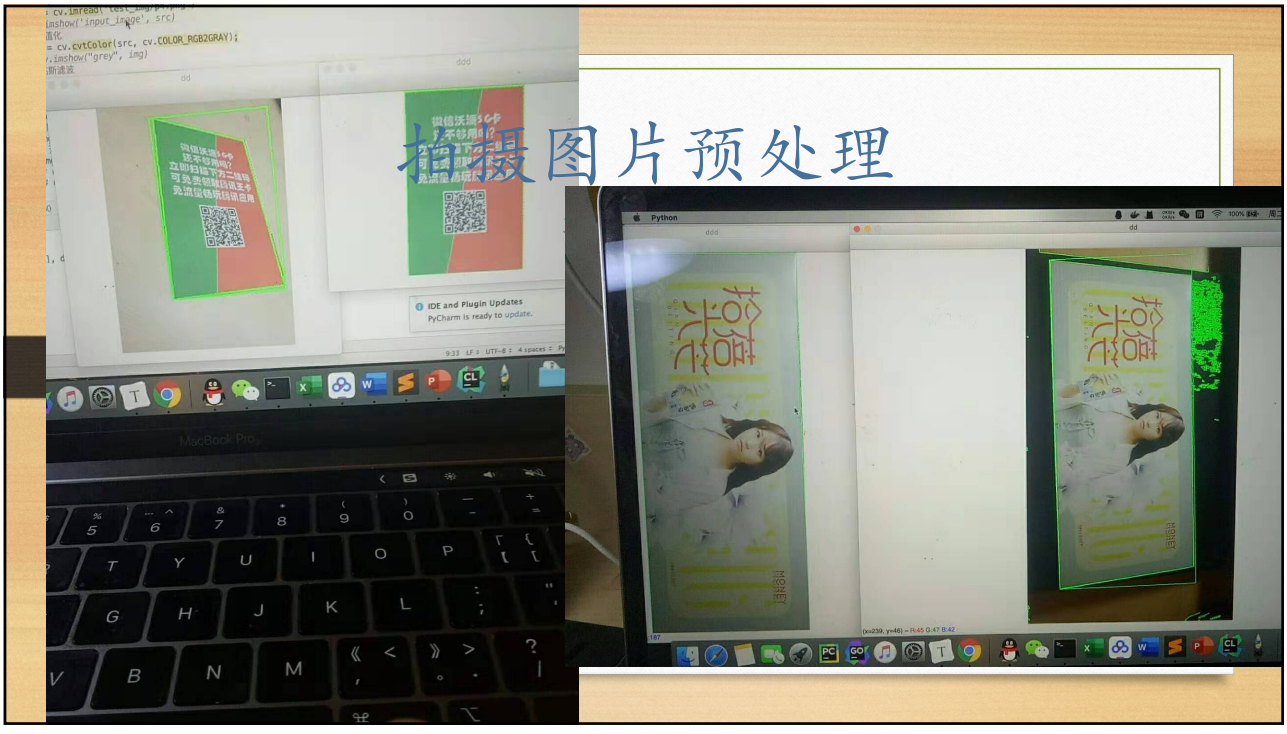
- 现在拥有了外轮廓的点集，接下来要做的是通过这
四边形的四个顶点
- 最小外接矩形拟合
- 凸包算法

由于拍照可能存在各
各样，最小外接矩形
个顶点进行后续变换

目的为了识别文本，
时去掉对比强烈的区



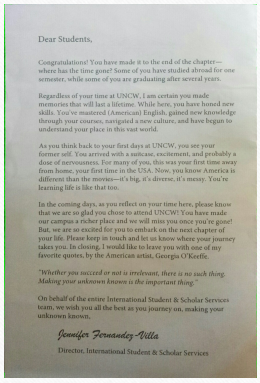
7



8

拍摄图片预处理

透射算法



Dear Students,

Congratulations! You have made it to the end of the chapter—where has the time gone? Some of you have studied abroad for one semester, while some of you are graduating after several years.

Regardless of your time at UNCW, I am certain you made memories that will last a lifetime. While here, you have honed new skills. You've mastered (American) English, gained new knowledge through your courses, navigated a new culture, and have begun to understand your place in this vast world.

As you think back to your first days at UNCW, you see your former self. You arrived with a suitcase, excitement, and probably a dose of nervousness. For many of you, this was your first time away from home, your first time in the USA. Now, you know America is different than the movies—it's big, it's diverse, it's messy. You're learning life is like that too.

In the coming days, as you reflect on your time here, please know that we are so glad you chose to attend UNCW! You have made our campus a richer place and we will miss you once you're gone! But, we are so excited for you to embark on the next chapter of your life. Please keep in touch and let us know where your journey takes you. In closing, I would like to leave you with one of my favorite quotes, by the American artist, Georgia O'Keeffe.

"Whether you succeed or not is irrelevant, there is no such thing. Making your unknown known is the important thing."

On behalf of the entire International Student & Scholar Services team, we wish you all the best as you journey on, making your unknown known.

Jennifer Fernandez-Villa

Director, International Student & Scholar Services

仿射变换是透视变换的特例

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \begin{cases} X' = \dfrac{X}{Z} \\ Y' = \dfrac{Y}{Z} \\ Z' = \dfrac{Z}{Z} \end{cases} \quad \begin{cases} X' = \dfrac{a_{11}x + a_{12}y + a_{13}}{a_{31}x + a_{32}y + a_{33}} \\ Y' = \dfrac{a_{21}x + a_{22}y + a_{23}}{a_{31}x + a_{32}y + a_{33}} \\ Z' = 1 \end{cases}$$

原图像里面的直线，经透视变换后仍为直线

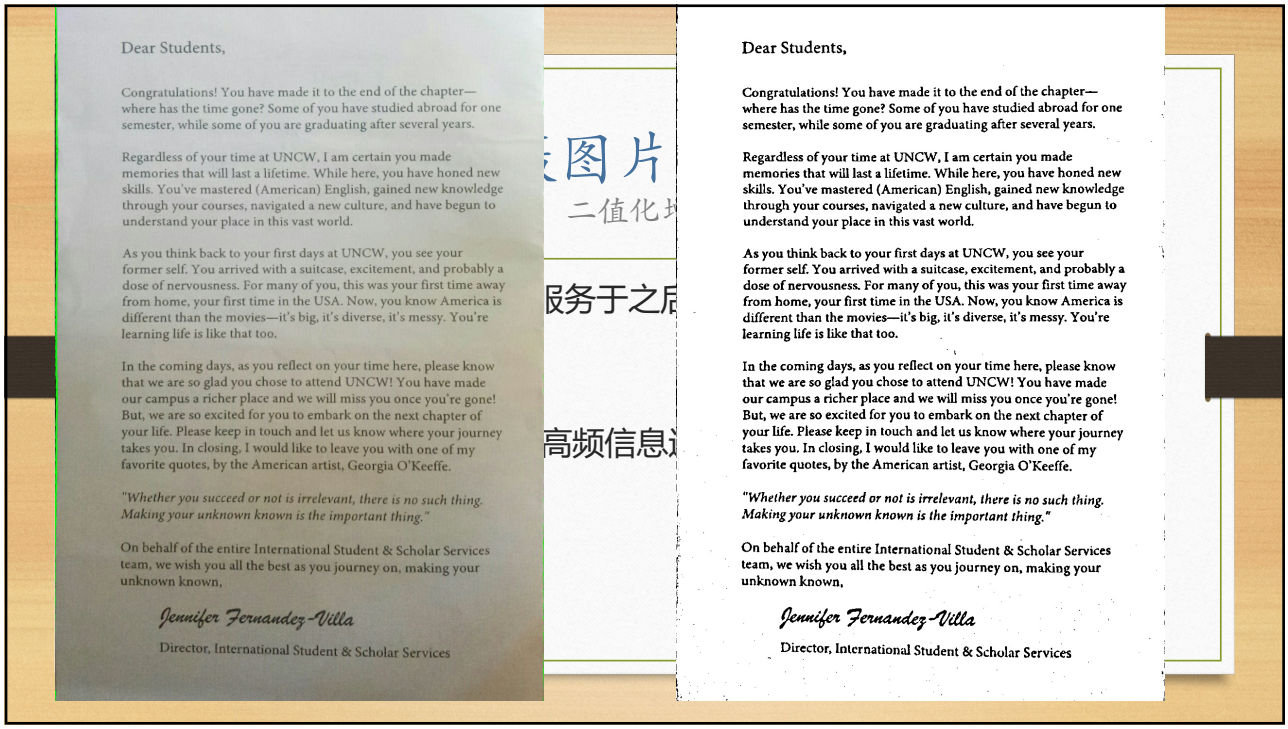
9

拍摄图片预处理

二值化增强

- 为了实现扫描效果，更好服务于之后的文本识别，还需要对图片进行二值化增强处理
- 图像锐化： 高通滤波（让高频信息通过，过滤低频信息）边缘得到增益

10



11

字符定位与切割

投影算法

- 水平投影：统计图像每一行有多少像素点是黑色的
- 通过定位波的区间来确定存在字符的位置

12

字符定位与切

投影算法

- 水平投影：统计图像每一行有多少像素点是黑

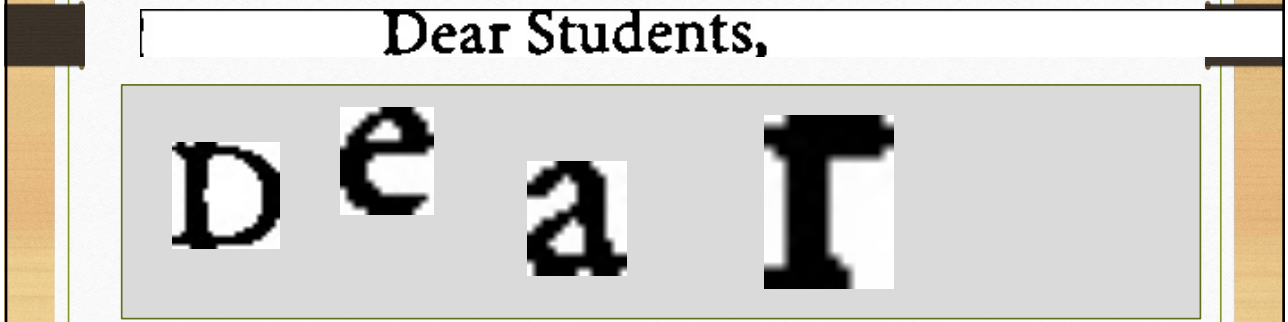
- 通过定位波的区间来确定存在字符的位置

13

字符定位与切割

投影算法ç

- 将每一行再进行垂直投影，获得单个字符：（还需再次水平投影）

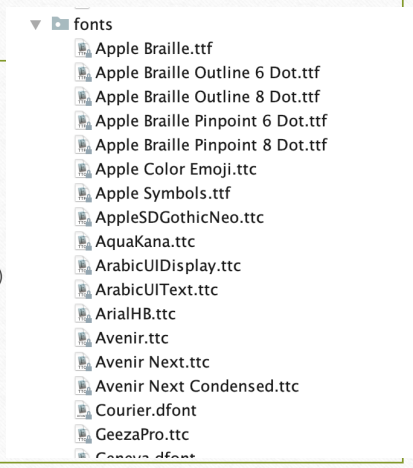


14

生成训练字符

plt

- 使用plt 生成大量字符训练集
  - 网上并无大量字符印刷数据集
  - 从自身电脑中导出字体：/Libarary/fonts
  - *# 创建Font对象:*
    font = ImageFont.truetype('font_test/Geneva.dfont', 60)
    # 创建Draw对象:
    draw = ImageDraw.Draw(image)
    # 输出文字:
    draw.text((0, 0), string[k], font=font, fill=(0, 0, 0))

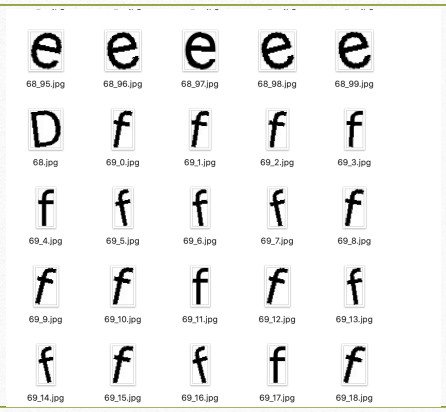


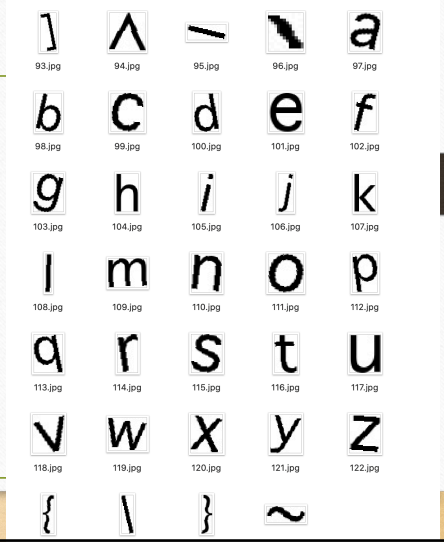
15

生成训练字符

plt

- 字体
- 角度
- 颜色





16

卷积神经网络

cnn

- 网络定义

Conv(kernel size 8*8, stride [1, 1, 1, 1])*16 个 feature map→Relu→
Max_pooling(kernel size 2*2, stride [1, 2, 2, 1])→
Conv(kernel size 5*5, stride [1, 1, 1, 1])*32 个 feature
Max_pooling(kernel size 1*1, stride [1, 1, 1, 1])→
FullConnectedLayer→Dropout(keep_prob 0.5)→
Adam Optimizer, softmax_cross_entropy_with_logits

```
Test Loss: 0.000395, Acc: 0.994681
Test Loss: 0.000293, Acc: 0.996277
Test Loss: 0.000304, Acc: 0.995213
Test Loss: 0.000297, Acc: 0.995745
Test Loss: 0.000439, Acc: 0.992021
Test Loss: 0.000293, Acc: 0.995213
Test Loss: 0.000323, Acc: 0.995213
Test Loss: 0.000124, Acc: 0.998936
Test Loss: 0.000110, Acc: 0.998936
Test Loss: 0.000101, Acc: 0.999468
```

17

识别

- 演示

	数字&字母
Train set	12192
Validation set	371
Accuracy(validation)	98.6%
Test set	612
Accuracy(test)	97.1%
Time cost	840.2s

18

总结算法动机

- 选择透射而非仿射： 自然场景文字框可能存在由于拍摄角度不同而导致的变形，有可能出现梯形
- 选择凸包拟合而非最小外接矩形： 对于文本识别，需要尽可能的去除背景的干扰，保留“文本框”，以及由于角度变形可能导致变换后文字变形过大
- 需要三次投影：第一次水平，得到一行文字；第二次在一行文字图像基础上垂直，得到单个字符，但此时单个字符竖直方向有空白，所以需要再次水平投影去除上下空白

19

问题及改进

- 针对场景较特殊，需要背景与所需文本框色差较大，否则矩形轮廓无法找到（需要针对不同图片调整opencv里的参数）
- 单独纯文字无法准确提取，这个需要使用（直线矫正）来进行预处理
- 识别率有待提高，可能还需要增加字体训练
- 当前并不支持手写体（外网新闻报道外国人写字五花八门特别是斜体粘连分割有很大麻烦）

20