

代码主要框架：

1. 使用 cv2 库中函数对原图片进行读取，同时进行灰度处理、找出表单的四角点以便透视变换操作；

```
image, gray, edged = Get_Outline(r'C:\Users\young\Desktop\1\4.jpg')
docCnt = Get_cnt(edged) # 四角点
```

2. 使用 imutils 中的工具进行透视变换；

```
result_img = four_point_transform(image, docCnt.reshape(4, 2))
```

3. 使用 cv2.adaptiveThreshold 进行二值化；

```
binary = cv2.adaptiveThreshold(~gray, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 35, -5)
```

4. 对二值化处理后的图片用 cv2.getStructuringElement 和 cv2.erode 进行横线和竖线提取；

```
#识别横线
kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (cols//scale, 1))
eroded = cv2.erode(binary, kernel, iterations = 1)
dilatedcol = cv2.dilate(eroded, kernel, iterations = 1)
```

```
kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (1, rows//scale))
eroded = cv2.erode(binary, kernel, iterations = 1)
dilatedrow = cv2.dilate(eroded, kernel, iterations = 1)
```

5. 对横线竖线进行与操作得到交点，由这些交点就可得出每个单元格的位置；

```
bitwiseAnd = cv2.bitwise_and(dilatedcol, dilatedrow)
```

由于横竖线有一定的宽度，所以可能造成获得多个交点的情况，为了解决这个问题，本人采用的是在一定范围内选取最靠前的点的坐标值：

```
#识别黑白图中的白色交叉点，将横纵坐标取出
ys_xs = np.where(bitwiseAnd>0)
mylisty=[] #纵坐标
mylistx=[] #横坐标
#通过排序，获取跳变的x和y的值，说明是交点，否则交点会有好多像素值相近，我只取相近值的最后一点
#这个10的跳变不是固定的，根据不同的图片会有微调，基本上为单元格表格的高度（y坐标跳变）和长度（x坐标跳变）
i = 0
myxs=np.sort(xs)
for i in range(len(myxs)-1):
    if(myxs[i+1]-myxs[i]>5):
        mylistx.append(myxs[i])
        i=i+1
mylistx.append(myxs[i]) #要将最后一个点加入

i = 0
myys=np.sort(ys)
#print(np.sort(ys))
for i in range(len(myys)-1):
    if(myys[i+1]-myys[i]>10):
        mylisty.append(myys[i])
        i=i+1
mylisty.append(myys[i]) #要将最后一个点加入
```

6. 根据交点将图片分割成为一个个的单元格；

```
ROI = result_img[mylisty[i]+3:mylisty[i+1]-3,mylistx[j]:mylistx[j+1]-3] #减去3的原因是由于我缩小ROI范围
```

7. 将分割后的图片通过 tessract-ocr 的接口传入进行文字识别；

```
pytesseract.pytesseract.tesseract_cmd = r'C:/Program Files/Tesseract-OCR/tesseract.exe'  
text1 = pytesseract.image_to_string(ROI,lang='chi_sim')
```

8. 最后将识别出的文字写入 excel 表存储起来

```
worksheet.write(i, j, label=text1)
```