数字媒体技术作业2

`罗剑杰 15331229 数字媒体`

作业内容：使用你第一次作业的结果,将 A4 纸上的线条单独识别出来。

实验报告

1. 实验环境

- Ubuntu 16.04 LTS
- python 3.5.2
- opencv 3.1.0, 通过这篇文章来进行配置
- numpy 1.13.3
- 如果还是不能成功运行，请在提交目录的根目录下使用`requirements.txt`来补全可能缺失的依赖库

2. 实验步骤

1. 通过阀值进行前后景分割(将线条和A4纸分开)

```
img = cv2.imread('./input.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# 通过阀值进行前后景的分割
# 这里的170是试出来的，为的是让所有的线段都可以被分离出来
ret, thresh = cv2.threshold(gray, 170, 255, 0)
```

需要注意的问题是，阀值分割的对象需要是灰度图。

`cv2.threshold`第一参数是原始图片（为灰度模式图片）；第二个参数是阀值，用来分类像素；第三个参事指定大于（有时小于）阀值时，重新给它分配的值；第四个参数指定阀值模式，有五种模式：

- cv2.THRESH_BINARY
- cv2.THRESH_BINARY_INV
- cv2.THRESH_TRUNC

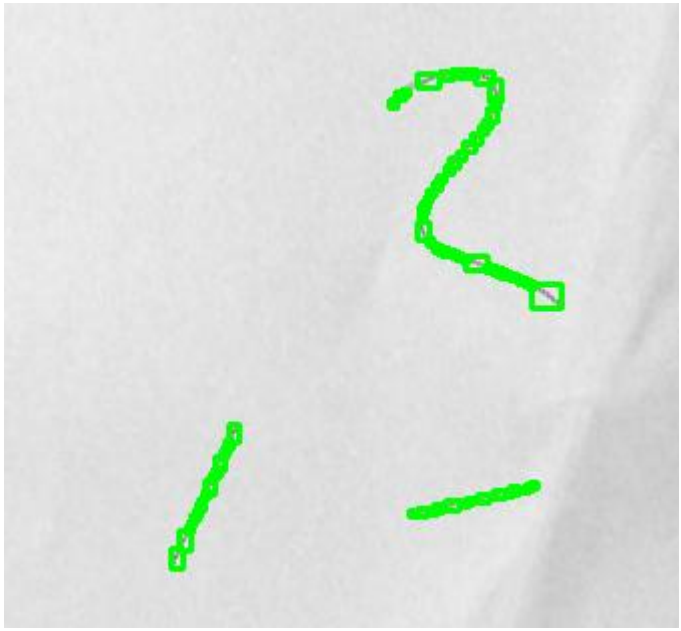
- cv2.THRESH_TOZERO
- cv2.THRESH_TOZERO_INV

因此这里需要重点设置的就是调第二个参数，如果使用网上的经验值127的话对于本张图片来说会有一条边不会被分离出来，经过尝试，第二个参数设置为170的时候所有的线段都可以被分离出来。

2/1/21

2. 形态学处理，腐蚀图像

直接使用`thresh.jpg`来进行轮廓检测的话会出现如下结果，为了解决这个问题我们需要在轮廓检测前先对图像进行腐蚀处理。



```
# 形态学图像处理(膨胀腐蚀)
# 这里迭代的次数2是试出来的，kernel的设置是参考了网上的教程的
kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (5, 5))
erosion = cv2.erode(thresh, kernel, iterations=2)
```

腐蚀后图片中的线就会变得更粗，方便轮廓检测

2

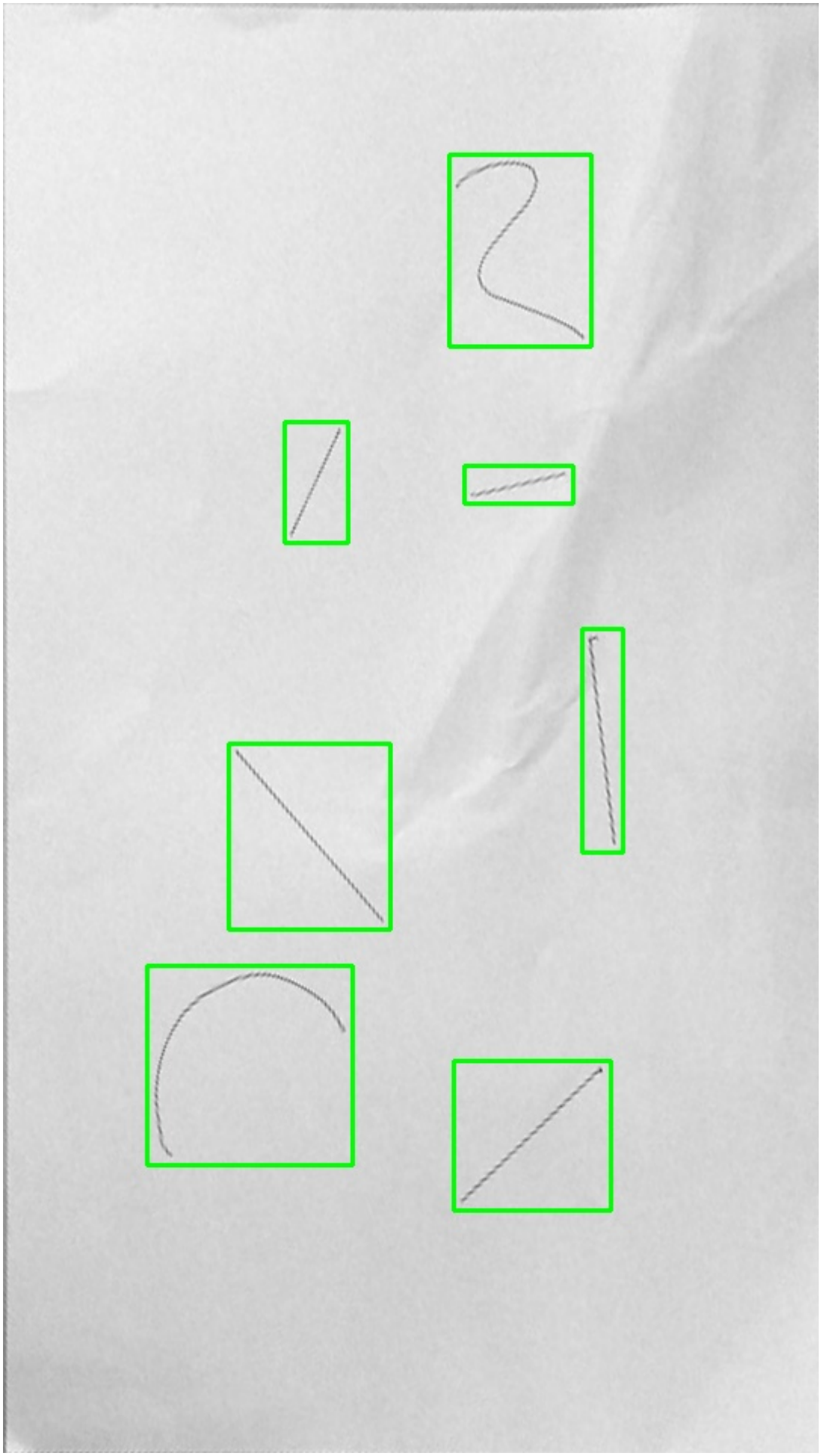
3. 轮廓检测

```
image, contours, hierarchy = cv2.findContours(thresh,                cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE )
```

使用`cv2.CHAIN_APPROX_SIMPLE`参数可以使得返回的轮廓`contours`只包含了外包矩阵的4个点。然后遍历`contours`这个列表里面的所有轮廓，通过`cv2.boundingRect`来定位矩阵，通过`cv2.rectangle`来画出轮廓矩阵，就可以把所有的线段检测给画出来。

```
n = len(contours)
outputs = []
# conturs的第一个元素是框出整个图片的大矩阵，故不选
for i in range(1, n):
    cnt = contours[i]
    x, y, w, h = cv2.boundingRect(cnt)
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
```

画出的结果如下：



4. 通过**ROI**将每张图片输出

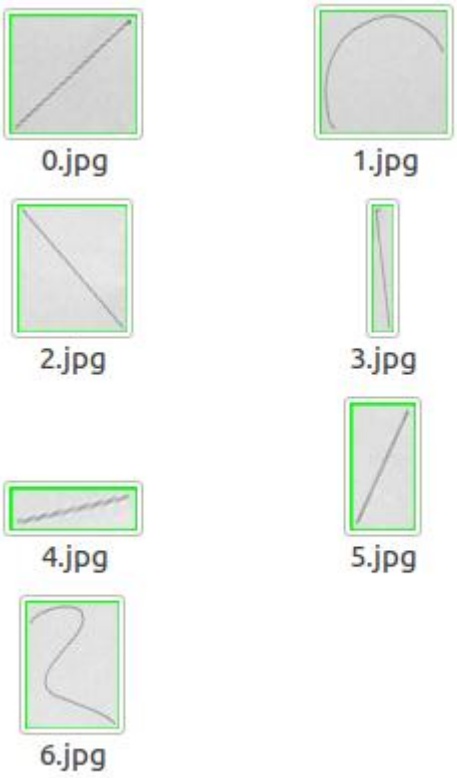
```
n = len(contours)
outputs = []
# conturs的第一个元素是框出整个图片的大矩阵，故不选
for i in range(1, n):
    cnt = contours[i]
    x, y, w, h = cv2.boundingRect(cnt)
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)

    # 使用ROI来进行分割
    # 这里两边的坐标体系有点迷
    part_img = img[y:y + h, x:x + w]
    outputs.append(part_img)
```

这里需要注意的问题是**opencv**里面的坐标系统和**numpy**里面的坐标系统有一点小小的出入，所以在取图片中的一段的时候**x**和**y**的位置需要调换一下。最后把所有的分割的图片输出到`./ouput`的文件夹中。

3. 实验结果

input

output



参考资料

1. opencv-python形态学处理
2. opencv-python阀值分割1
3. opencv-python阀值分割2
4. opencv-python轮廓检测