

文件夹内含：

jpg 文件夹

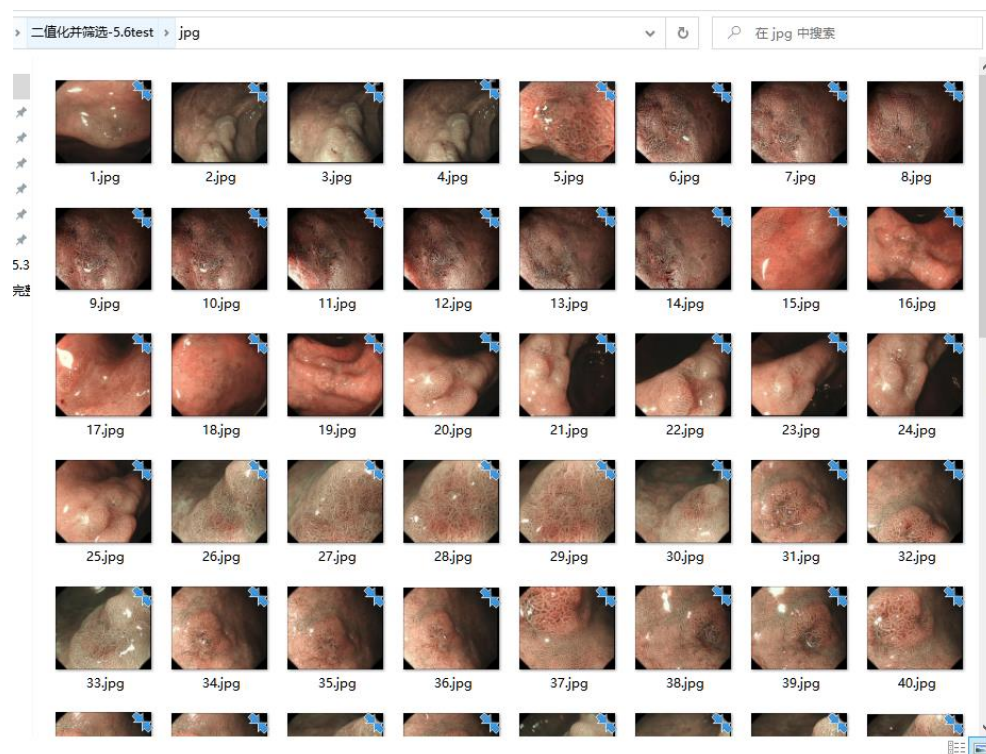
png 文件夹

1. 二值化.py
2. 统计占比.py
3. 筛选.py
4. 移除.py

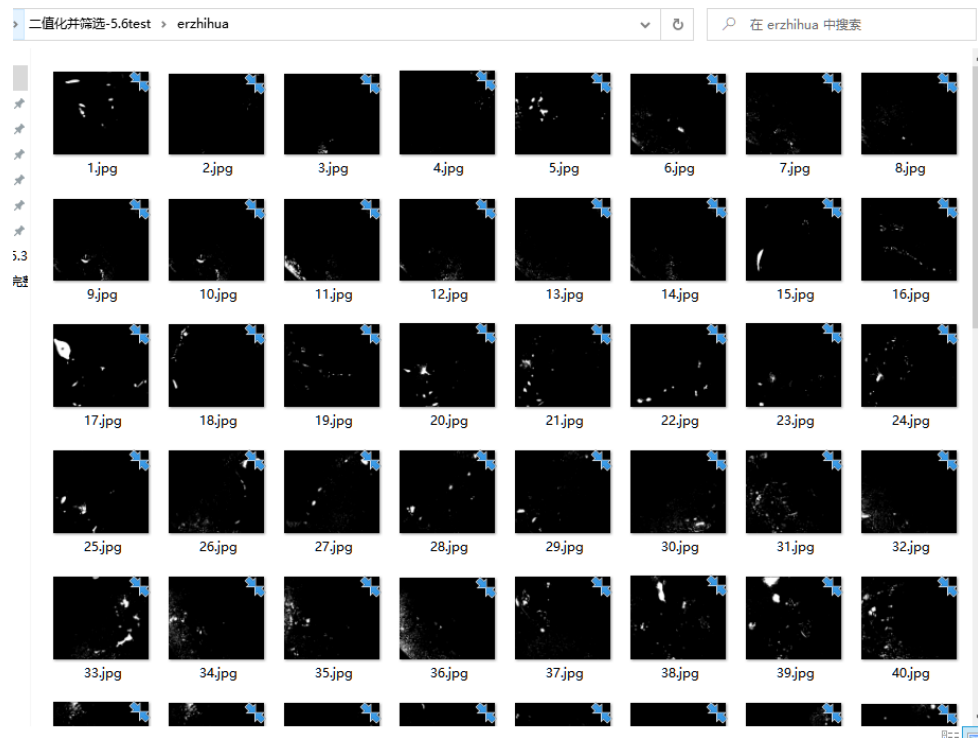
操作步骤：

### 1. 运行 1.二值化.py

实现批量二值化处理



二值化处理后的图片在文件夹 erzhihua (该文件夹自动生成无需创建)

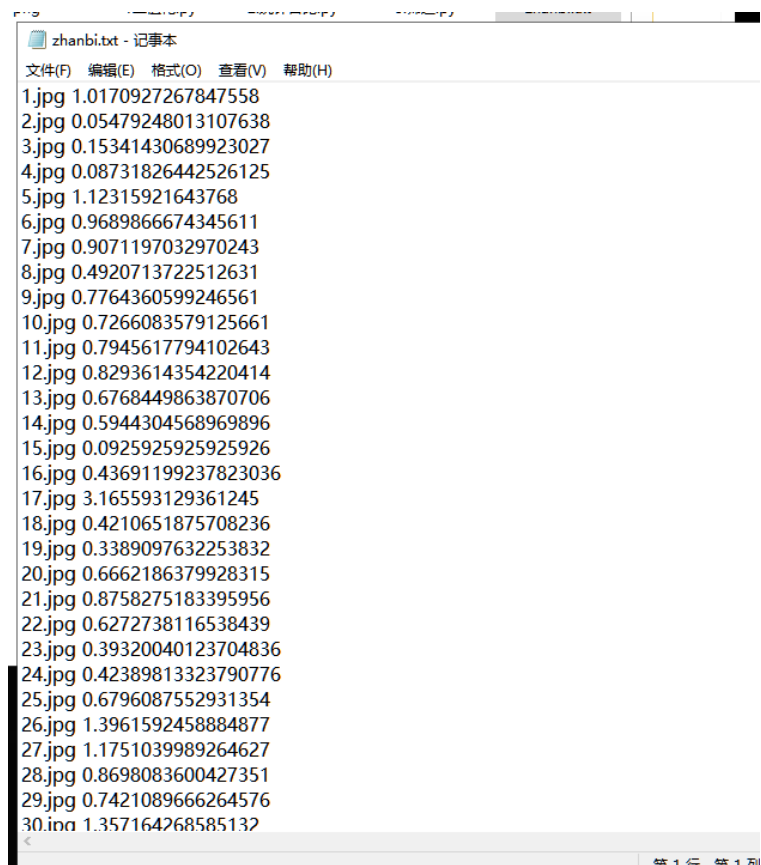


也可以自己修改代码中的阈值。越接近于 255 表示越白

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
ret, mask = cv2.threshold(gray, 230, 255, 0) #简单二值化
```

## 2. 运行 2.统计占比.py

生成了统计图片中白色所占比的 zhanbi.txt



```
zhanbi.txt - 记事本
文件(F)  编辑(E)  格式(O)  查看(V)  帮助(H)

1.jpg 1.0170927267847558
2.jpg 0.05479248013107638
3.jpg 0.15341430689923027
4.jpg 0.08731826442526125
5.jpg 1.12315921643768
6.jpg 0.9689866674345611
7.jpg 0.9071197032970243
8.jpg 0.4920713722512631
9.jpg 0.7764360599246561
10.jpg 0.7266083579125661
11.jpg 0.7945617794102643
12.jpg 0.8293614354220414
13.jpg 0.6768449863870706
14.jpg 0.5944304568969896
15.jpg 0.0925925925925926
16.jpg 0.43691199237823036
17.jpg 3.165593129361245
18.jpg 0.4210651875708236
19.jpg 0.3389097632253832
20.jpg 0.6662186379928315
21.jpg 0.8758275183395956
22.jpg 0.6272738116538439
23.jpg 0.39320040123704836
24.jpg 0.42389813323790776
25.jpg 0.6796087552931354
26.jpg 1.3961592458884877
27.jpg 1.1751039989264627
28.jpg 0.8698083600427351
29.jpg 0.7421089666264576
30.inn 1.357164268585132
第 1 行 第 1 列
```

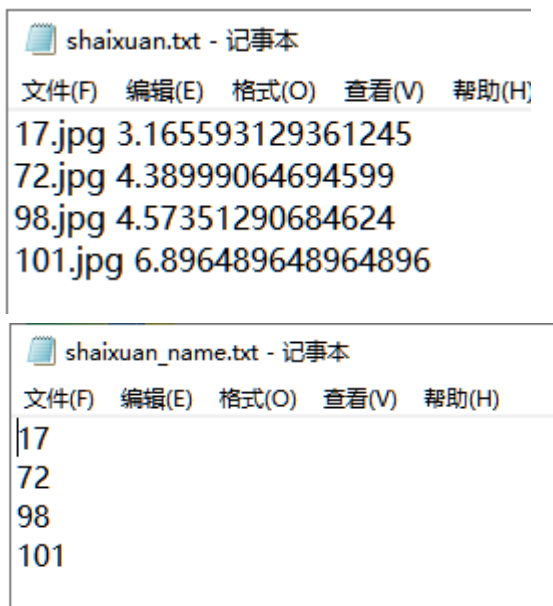
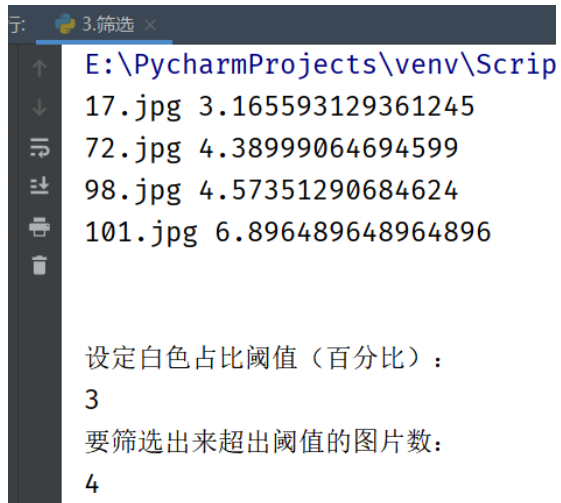
### 3. 运行 3.筛选.py

可以根据需要自己设定阈值，这里设定了阈值为 3（即白色占 3%）



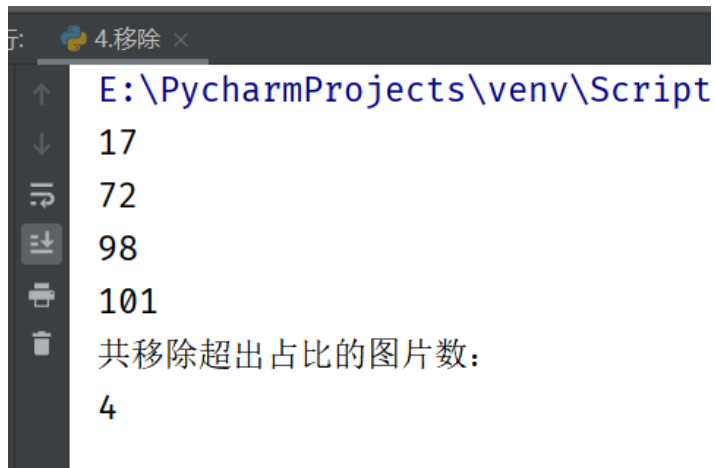
```
1 # coding:utf-8
2 import ...
5
6 output_dir = './erzhihua/' # 二值化处理后存放的目录
7 txt_dir = 'zhanbi.txt' #txt文件夹目录
8 thresh = 3 # 自定义设置白色占比的阈值
```

生成统计超出占比阈值的 shaixuan.txt 和图片名单 shaixuan\_name.txt



#### 4. 运行 4.移除.py

自动筛选超出占比的 jpg 和 png



于是超出占比的图片 jpg 和 png 被移动到了新的文件夹 fail 中(该文件夹自动生成无需创建)

而此时 jpg 和 png 文件夹中已没有了超出占比的图片，接下来即常规流程，如重命名等。

二值化筛选完后的完整文件夹应有如下内容：

