

2023 年秋季学期 图像处理 编程作业 04

姓名：梁乐彬 学号：

对图像中的目标进行计数处理。

本次作业 100 分

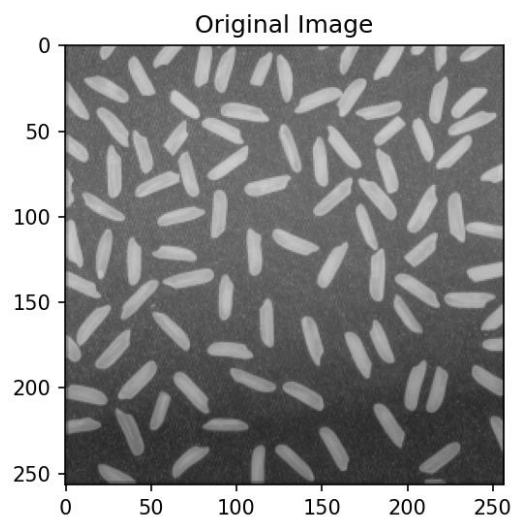
1. 请利用图像处理课程所学的相关技术，求图像 Fig01.tif 中米粒的个数及其各个米粒的大小。

【1】 分析米粒图像特征：

可以看到该图像的光照不均匀，上半部分和下半部分的图像曝光不同。

可以采用开运算得到背景图像，利用原图像减去背景图像可以得到除去背景光照的米粒图像，这样方便使用二值化提取米粒像素。

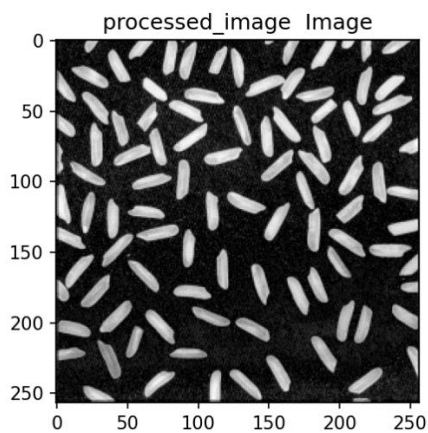
```
# 读取图像并转换为灰度图像
image = cv2.imread('Fig01.tif')
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```



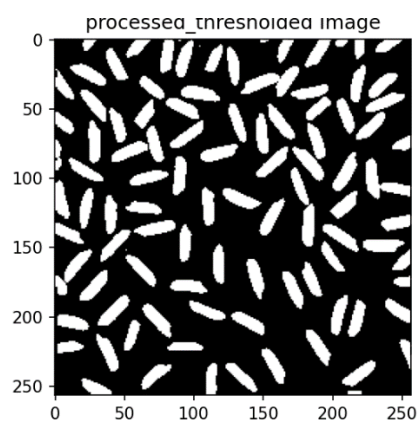
- ### 【2】 根据除去背景后的米粒图片，使用大津阈值法二值化分割后得到的二值图像。可以看到有部分米粒粘连在一起，这样会影响后续的连通域计算，所以要对图像进行下一步处理。

```
# 计算背景图像
background_image = cv2.dilate(opened_image, kernel_open, iterations=1)

# 原图减去背景图
processed_image = cv2.absdiff(gray_image, background_image)
```

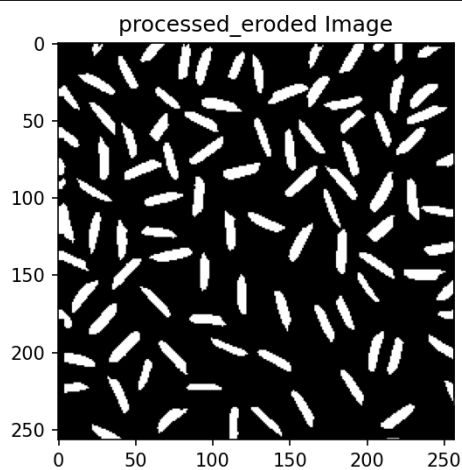


```
# 大津阈值法
_, processed_thresholded_image = cv2.threshold(processed_image, 0, 255, cv2.THRESH_OTSU)
```



【3】 对图像进行腐蚀操作，分离粘连在一起的米粒。

```
# 腐蚀操作
kernel_erode = cv2.getStructuringElement(cv2.MORPH_CROSS, (2, 2))
processed_eroded_image = cv2.erode(processed_thresholded_image, kernel_erode, iterations=2)
```



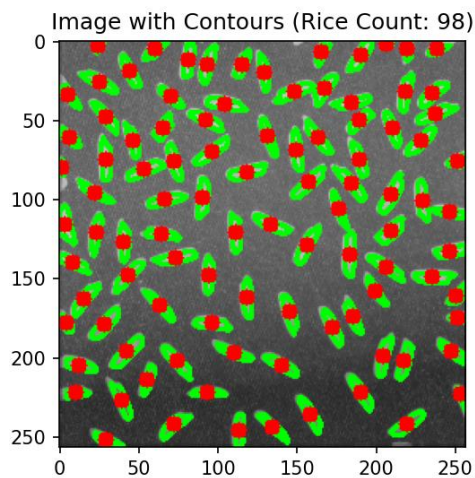
【4】 根据计数连通域，并根据米粒像素大小，计算米粒个数

```
# 轮廓查找
contours, _ = cv2.findContours(processed_eroded_image, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

# 统计米粒个数和计算平均大小
rice_count = 0
total_area = 0

for i, contour in enumerate(contours):
    area = cv2.contourArea(contour)
    if 20 < area < 5000:
        rice_count += 1
        total_area += area

print("米粒个数: ", rice_count)
```



在原图像中绘制轮廓和中心点

```
# 在图像上绘制轮廓和中心红点
for i, contour in enumerate(contours):
    area = cv2.contourArea(contour)
    if 20 < area < 5000:
        # 获取轮廓中心位置
        M = cv2.moments(contour)
        cX = int(M["m10"] / M["m00"])
        cY = int(M["m01"] / M["m00"])

        # 在原图上绘制轮廓
        cv2.drawContours(image, [contour], -1, (0, 255, 0), 2)

        # 在轮廓中心位置添加红点
        cv2.circle(image, (cX, cY), 5, (0, 0, 255), -1)
```

代码完整流程：

1.读取图像并转换为灰度图像：

通过 OpenCV 库读取名为 'Fig01.tif' 的图像文件。

将读取的彩色图像转换为灰度图像，以简化后续处理。

2.形态学操作 - 开运算：

使用椭圆形卷积核进行开运算 (morphological opening)。开运算有助于平滑图像、去除小的噪声和不规则结构。

3.计算背景图像：

通过膨胀操作，计算出平滑后的图像的背景图像。

4.原图减去背景图：

通过将原始灰度图像减去背景图像，突出图像中的前景目标。

5.大津阈值法二值化：

使用大津阈值法对差异图像进行二值化。大津阈值法自动计算最佳阈值，将图像转换为黑白二值图像。

6.腐蚀操作：

使用十字形卷积核进行腐蚀操作。腐蚀操作有助于减小目标区域的大小，进一步去除小的结构，分离连接在一起的米粒。

7.轮廓查找和统计：

使用 findContours 函数查找图像中的轮廓。通过循环遍历所有轮廓，统计符合面积条件的米粒个数和总面积。

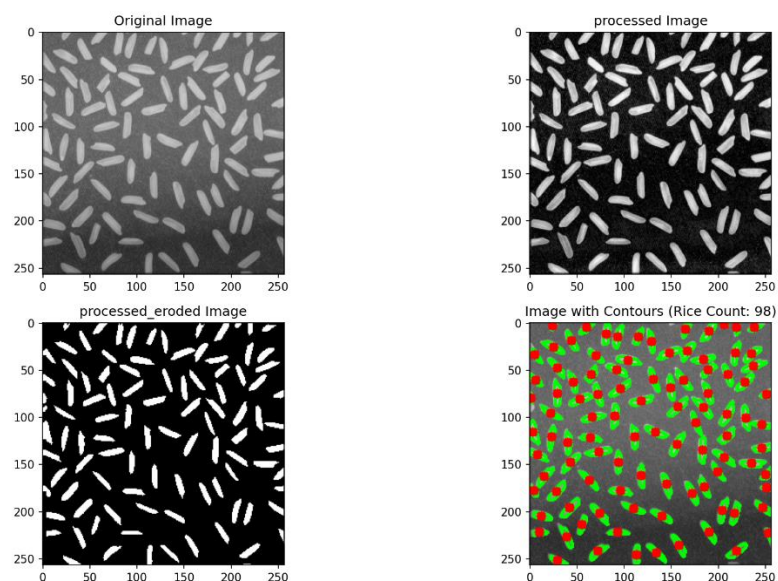
8.绘制轮廓和中心红点：

遍历符合条件的轮廓，获取轮廓中心位置。在原图上绘制符合条件的轮廓，并在米粒中心位置添加红点。

9.显示图像：

使用 Matplotlib 库显示包括原图、处理后的图像、腐蚀后的图像和带有轮廓的图像在内的多个子图。最终显示米粒个数。

最终效果图：（最终米粒计数为 98 颗，包含不完整米粒）



仍存在的问题：如最左边的米粒因为腐蚀操作被分割为两个，同时对于顶部较小的米粒被腐蚀去掉，无法计数。可能需要使用其他算法来进一步优化。

