COMP9517 report1,
z5216214 Haojin Guo

1. Task1,

First, read a grayscale image (image = cv2. imread("rice.png",0)). Applying the method iso-data intensity thresholding can get the corresponding binary image. Using the following expression, foreground = (image < t), background = (image > t), can get Boolean Matrix.
Meanwhile, μ0 = np.mean(image[rice_kernels]) and μ1 = np.mean(image[background]) can get the mean of intensity values respectively.
Here, epsilon = 0,005
In addition, the line graph corresponding to different T values is realized through the following code:

```
x = [i for i in range(len(t_list))]
y = [j for j in t_list]plt.plot(x, y, 's-', color='r', label="t_Value")
plt.xlabel("Iteration times of t value")
plt.ylabel("The value of t")
plt.show()
```

Since the value of t adopts the method of random value, the output result of t-plot may not be the same each time it runs.

- Task1.1 for "rice_img1.png", the output images as follows:

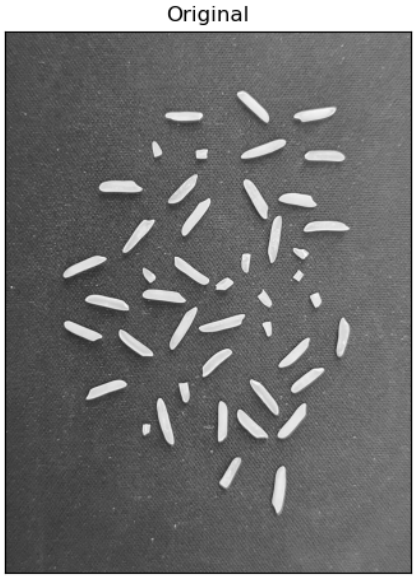

Figure1.1: rice1_Task1.png

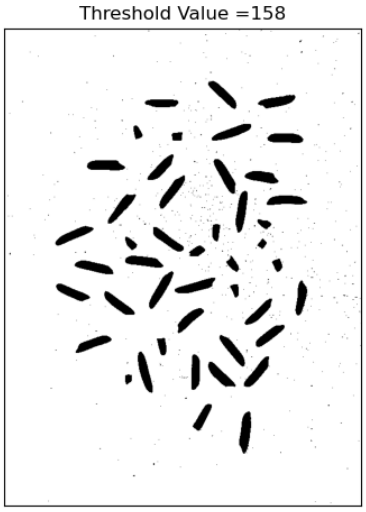

Figure1.1: rice1_Task1.png

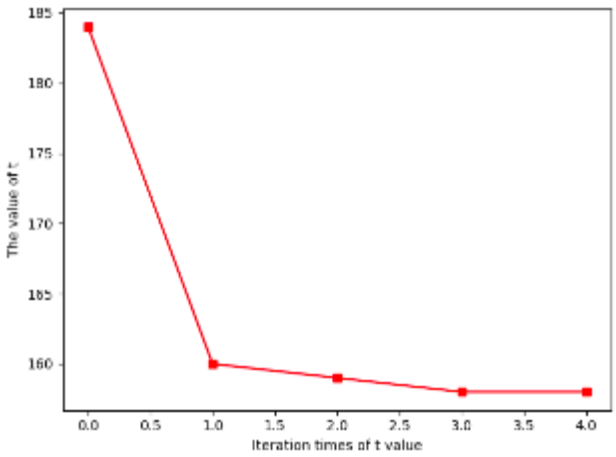

Figure1.1: rice1_Task1.png

- Task1.2, the outputs of rice_img2.png as follows:

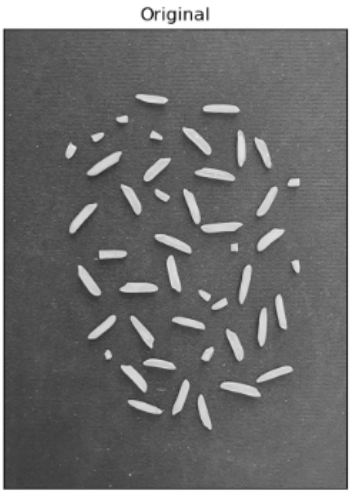

Figure1.2: rice2_Task1.png

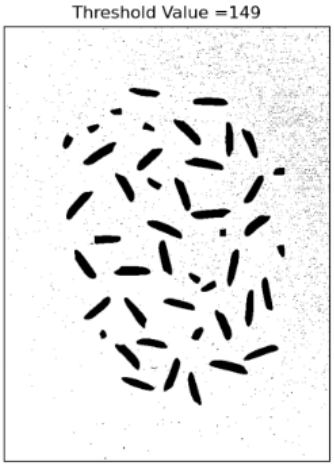

Figure1.2: rice2_Task1.png

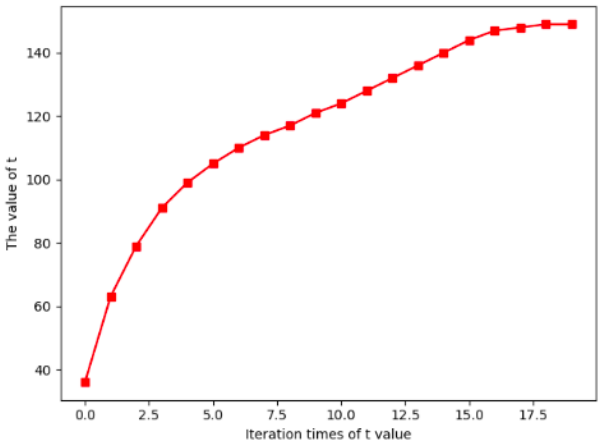

Figure1.2: rice2_Task1.png

- Task1.3, the outputs of “rice_img6.png" as follows:

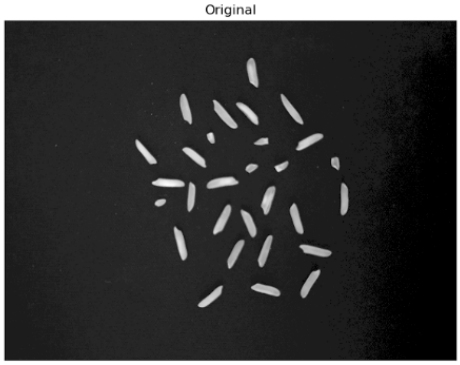

Figure1.3: rice6_Task1.png

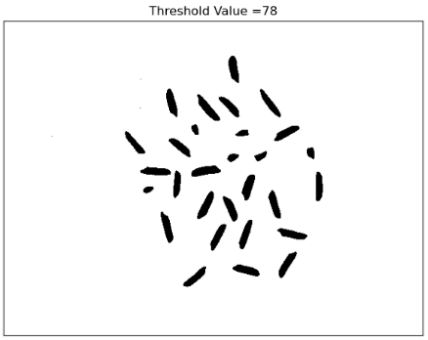

Figure1.3: rice6_Task1.png

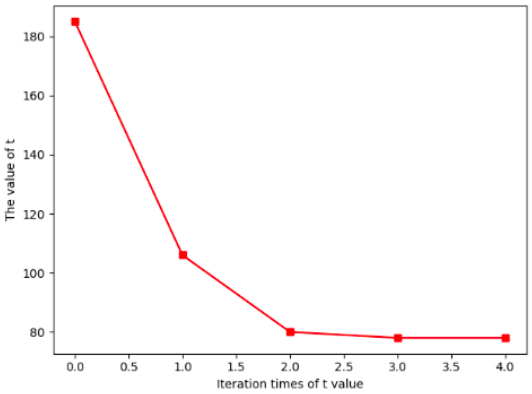

Figure1.3: rice6_Task1.png

- Task1.4, the outputs of “rice_img7.png” as follows:

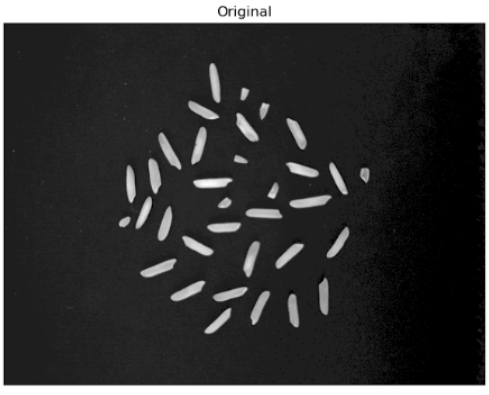

Figure1.4: rice7_Task1.png

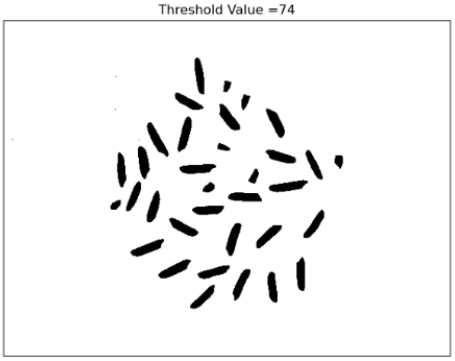

Figure1.4: rice7_Task1.png

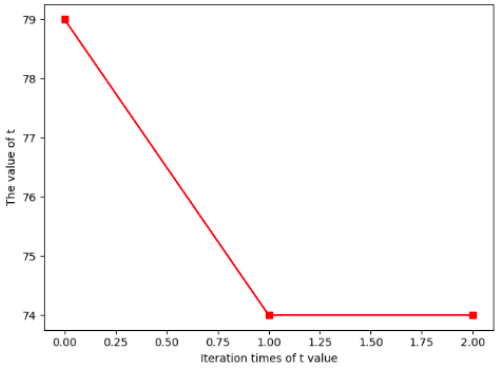

Figure1.4: rice7_Task1.png

Task2,

First, the image needs to be median filtered. Here I use the function of ndimage of ‘scipy’ library. The part of codes as below,

```
selem = ndi.generate_binary_structure(img.ndim, img.ndim)

median_blur_img = ndi.median_filter(img, … )
```

The results of median filtering is as follows:

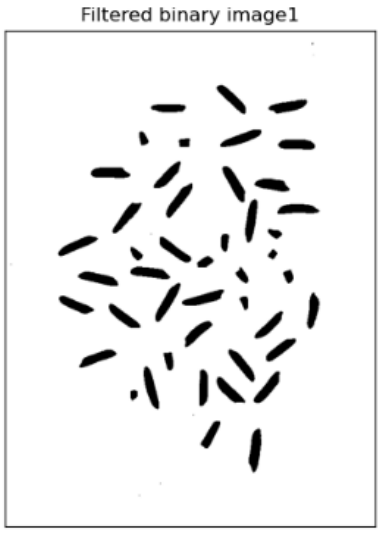

Figure2.1: rice1_Task2.png

Figure2.1: rice2_Task2.png

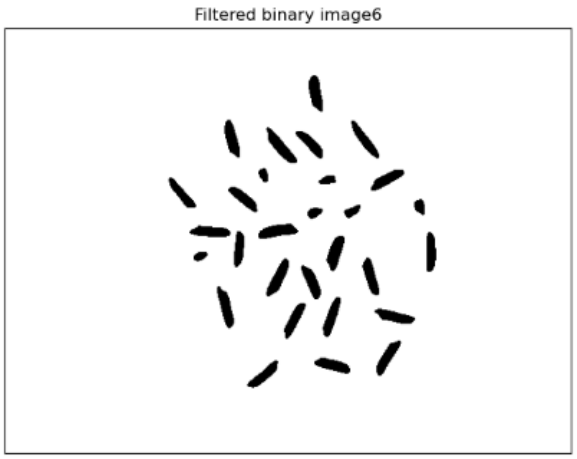

Figure2.1: rice6_Task2.png

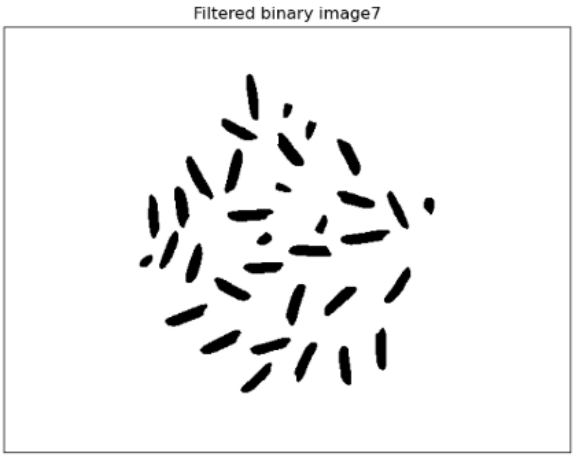

Figure2.1: rice7_Task2.png

Secondly, using two-pass algorithm calculates the number of rice kernels.

Task3.

The result of removing the bad seeds is not ideal. That's because second pass of Task2 has too many problems, this procedure is somewhat difficult for me.

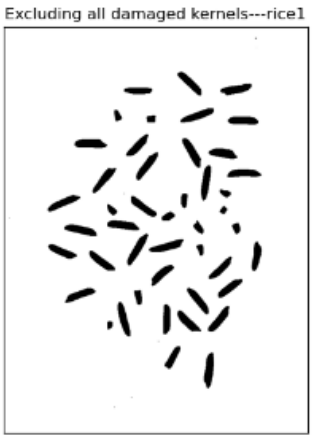

Figure3.1: rice1_Task3.png

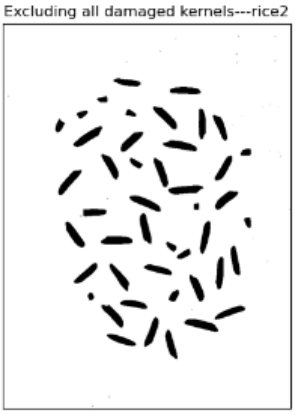

Figure3.2: rice2_Task3.png

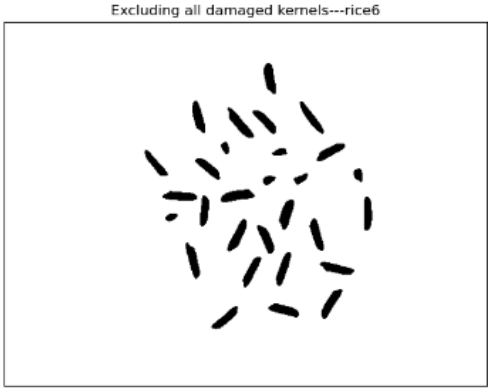

Figure3.3: rice6_Task3.png

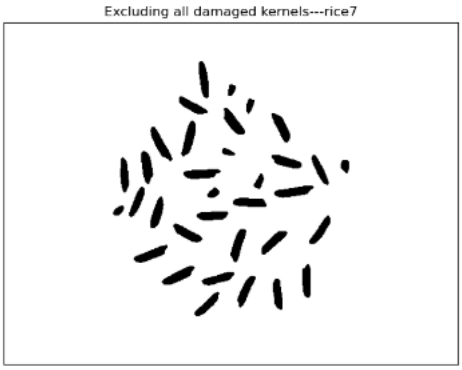

Figure3.4: rice7_Task3.png