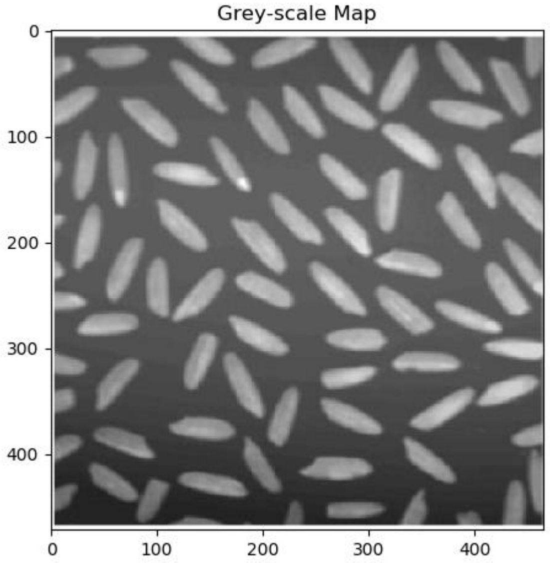
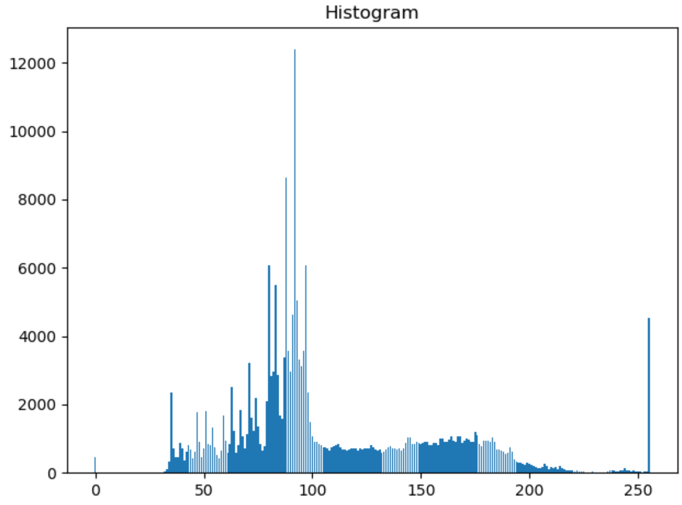
Homework 10

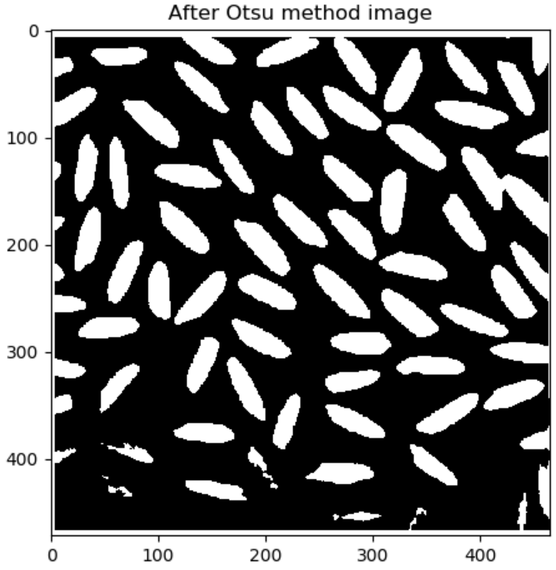
(1) Problem statement

Implement Otsu's thresholding method

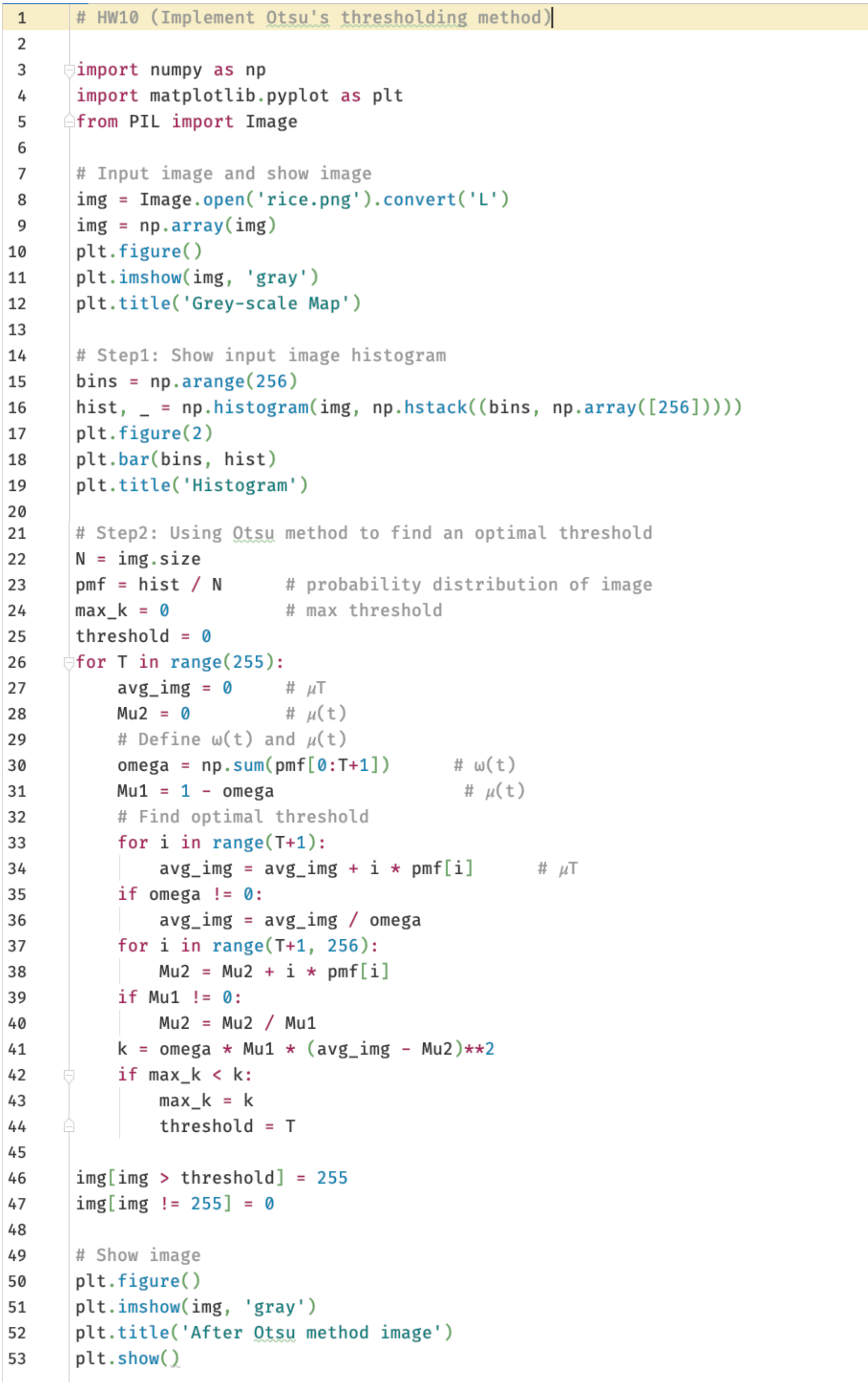
(2.1) Experimental results







(2.2) Source code



```python
# HW10 (Implement Otsu's thresholding method)

import numpy as np
import matplotlib.pyplot as plt
from PIL import Image

# Input image and show image
img = Image.open('rice.png').convert('L')
img = np.array(img)
plt.figure()
plt.imshow(img, 'gray')
plt.title('Grey-scale Map')

# Step1: Show input image histogram
bins = np.arange(256)
hist, _ = np.histogram(img, np.hstack((bins, np.array([256]))))
plt.figure(2)
plt.bar(bins, hist)
plt.title('Histogram')

# Step2: Using Otsu method to find an optimal threshold
N = img.size
pmf = hist / N      # probability distribution of image
max_k = 0           # max threshold
threshold = 0
for T in range(255):
    avg_img = 0     # μT
    Mu2 = 0         # μ(t)
    # Define ω(t) and μ(t)
    omega = np.sum(pmf[0:T+1])      # ω(t)
    Mu1 = 1 - omega                  # μ(t)
    # Find optimal threshold
    for i in range(T+1):
        avg_img = avg_img + i * pmf[i]      # μT
    if omega != 0:
        avg_img = avg_img / omega
    for i in range(T+1, 256):
        Mu2 = Mu2 + i * pmf[i]
    if Mu1 != 0:
        Mu2 = Mu2 / Mu1
    k = omega * Mu1 * (avg_img - Mu2)**2
    if max_k < k:
        max_k = k
        threshold = T

img[img > threshold] = 255
img[img != 255] = 0

# Show image
plt.figure()
plt.imshow(img, 'gray')
plt.title('After Otsu method image')
plt.show()
```