

(1) Assignment statement

Homework 8 (Due: 5/9)

Implement Otsu's thresholding method

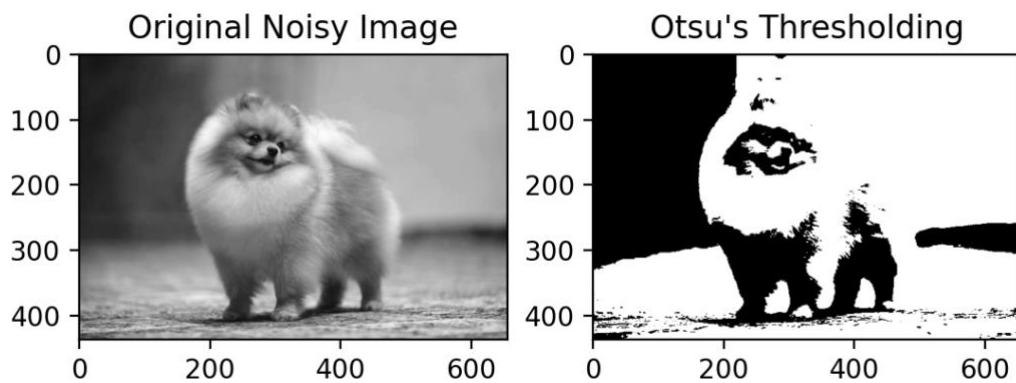
(2)

(a) Input/output images

Input



Output



(b) Source code

```
hw08.py > ...
1  import cv2 as cv
2  import numpy as np
3  from matplotlib import pyplot as plt
4
5  img = cv.imread('test.jpg', cv.IMREAD_GRAYSCALE)
6
7  # Otsu's thresholding
8  ret2,th2 = cv.threshold(img,0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)
9
10 plt.subplot(1,2,1),plt.imshow(img,'gray')
11 plt.title('Original Noisy Image')
12
13 plt.subplot(1,2,2),plt.imshow(th2,'gray')
14 plt.title('Otsu\'s Thresholding')
15
16 plt.show()
17
```

(c) Comments

這次的作業參考了網路上的資料。網路上有滿多人做過 Otsu's thresholding method 的，不過每個人的做法都不同，因此參考他人的寫法時，能夠學到不一樣的做法。大部分人都會使用現有函式庫，節省程式撰寫時間的同時，也能讓程式碼更精簡好讀。另外，這次我也看到有人要畫多張圖的時候，會將圖片先依序存在 python list 當中，要繪製圖片時再跑迴圈繪製，這種寫法的好處是若要跑大量圖片時，就不用每張圖都寫一個 plt.subplot、plt.title 等等函式。不過因為這次我要繪製的圖片只有兩張，數量沒有多到需要用這種寫法，因此我還是直接一張一張畫。