NTNU 影像處理 HW10

廖家緯

2020.5.20

- **Outline**:

  Implement Otsu's thresholding method.

- **Code(Python):**

```python
# coding: utf-8
import numpy as np
import matplotlib.pyplot as plt
import cv2

# function
def his(img, s):
    [img_his, img_bin] = np.histogram(img.flatten(), range(257))
    plt.bar(range(256), img_his, color = 'blue')
    plt.savefig(s + '_his.png')
    return img_his

def imgshow(img):
    cv2.imshow('My Image', img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

def a(t, p):
    return sum(p[:t+1])

def b(t, p):
    return sum(p[t+1:256])

def m(t, p):
    return sum(p[:256]*range(256))

def ma(t, p):
    return sum(p[:t+1]*range(t+1))

def val(t, p):
    return (ma(t, p)-m(t, p)*a(t, p))**2/(a(t, p)*b(t, p))



```

```
# main
I = cv2.imread('image.jpg', cv2.IMREAD_GRAYSCALE)
r, c = np.shape(I)
p = his(I, 'image')/(r*c)
maxval, t_index = max([(val(t, p), t) for t in range(50, 210)])

Out = np.zeros((r, c)).astype('uint8')

binary = lambda x:255 if x>t_index else 0

for i in range(r):
    for j in range(c):
        Out[i, j] = binary(I[i, j])

imgshow(Out)
```

- **Result:**

input image

Otsu’ s thresholding method

- **Experience:**

  Otsu’s thresholding method 精神在於群內最小變異、群間最大變異。而再找尋 t 的過程，需求取最大值。我們目前都採取暴力法計算，而我想到一個天馬行空的想法，可以先找 Histogram 中具有代表性的點進行多項式的插值，然後再用積分取代 $\sum$ 來計算，這樣就可以將問題轉成求分式函數的最大值，最後用 Gradient decent 來求取 t 的近似值 (以上純屬我的想法，沒有實作過 XD)。