

NTNU 影像處理 HW10

廖家緯

2020.5.20

- **Outline:**

Implement Otsu's thresholding method.

- **Code(Python):**

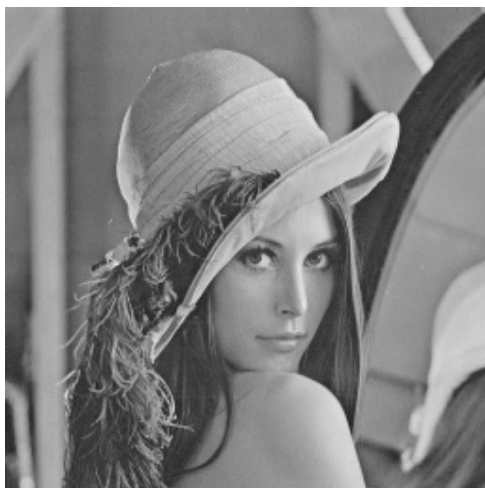
```
1  # coding: utf-8
2  import numpy as np
3  import matplotlib.pyplot as plt
4  import cv2
5
6  # function
7  def his(img, s):
8      [img_his, img_bin] = np.histogram(img.flatten(), range(257))
9      plt.bar(range(256), img_his, color = 'blue')
10     plt.savefig(s + '_his.png')
11     return img_his
12
13 def imgshow(img):
14     cv2.imshow('My Image', img)
15     cv2.waitKey(0)
16     cv2.destroyAllWindows()
17
18 def a(t, p):
19     return sum(p[:t+1])
20
21 def b(t, p):
22     return sum(p[t+1:256])
23
24 def m(t, p):
25     return sum(p[:256]*range(256))
26
27 def ma(t, p):
28     return sum(p[:t+1]*range(t+1))
29
30 def val(t, p):
31     return (ma(t, p)-m(t, p)*a(t, p))**2/(a(t, p)*b(t, p))
32
33
34
```

```

35 # main
36 I = cv2.imread('image.jpg', cv2.IMREAD_GRAYSCALE)
37 r, c = np.shape(I)
38 p = his(I, 'image')/(r*c)
39 maxval, t_index = max([(val(t, p), t) for t in range(50, 210)])
40
41 Out = np.zeros((r, c)).astype('uint8')
42
43 binary = lambda x:255 if x>t_index else 0
44
45 for i in range(r):
46     for j in range(c):
47         Out[i, j] = binary(I[i, j])
48
49 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）。