NTNU 影像處理 HW4

廖家緯

2020.4.8

- **Outline**:

  1. Develop a histogram equalization (HE) program;
  2. Apply the HE to i) gray, ii) color images;
  3. For each input image, print out the input/output images and their histograms.
  4. Discuss your experiments.

  - Note that

    For a color image C,

    (i) Convert it into a gray image G;

    (ii) Apply HE to G to get G';

    (iii) For each pixel of C, modify its color (r, g, b) by $(r', g', b') = (r, g, b) \times G'/G$.

- **Code(Python):**

```python
# coding: utf-8

#套件
import numpy as np
import matplotlib.pyplot as plt
import cv2

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


# Histogram Equalization
def HE(img):
    [img_his, img_bin] = np.histogram(img.flatten(), range(257))
    img_cdf = np.cumsum(img_his)
```

```
    img_m_cdf = np.ma.masked_equal(img_cdf, 0) #除去直方圖中的0值
    img_cdf_tr = (img_m_cdf - img_m_cdf.min())*255 /
        (img_m_cdf.max()-img_m_cdf.min())
    img_rv = np.ma.filled(img_cdf_tr, 0).astype('uint8') #補回0
    img_HE = img_rv[img]
    [img_HE_his, img_HE_bin] = np.histogram(img_HE.flatten(), range(257))

    #長條圖
    plt.subplot(3, 1 ,1) #1列2行，編號1
    plt.bar(range(256), img_his, color = 'lightblue') #原始圖片長條圖
    plt.title('Original image')

    plt.subplot(3, 1 ,3) #1列2行，編號2
    plt.bar(range(256), img_HE_his, color = 'red') #圖片HE後長條圖
    plt.title("HE's image")

    return img_HE


# 讀取圖檔
img = cv2.imread('image.jpg')
n = np.shape(img)[0] #列
m = np.shape(img)[1] #行

# 轉為灰階
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# 灰階HE
gray_HE = HE(gray)
imgshow(gray_HE)

# 彩圖HE:分別對R,G,B做HE
[R, G, B] = cv2.split(img)
HE_R = HE(R)
HE_G = HE(G)
HE_B = HE(B)
img_HE = cv2.merge([HE_R, HE_G, HE_B])
imgshow(img_HE)

# 彩圖HE:老師給的公式
img = np.double(img)
[R, G, B] = cv2.split(img)
R_HE = R*gray_HE/gray
G_HE = G*gray_HE/gray
B_HE = B*gray_HE/gray
img_HE_tr = np.uint8(cv2.merge([R_HE, G_HE, B_HE]))
imgshow(img_HE_tr)
```

- **Result:**

彩圖

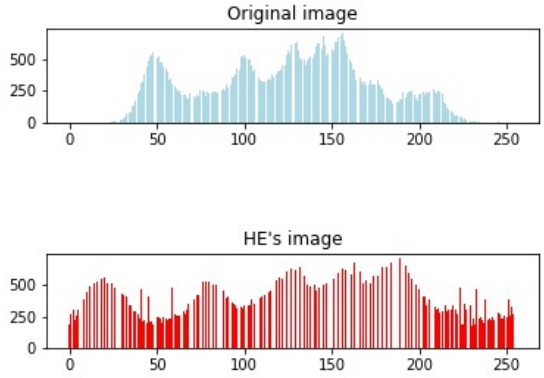
彩圖 Histogram Equalization

灰階

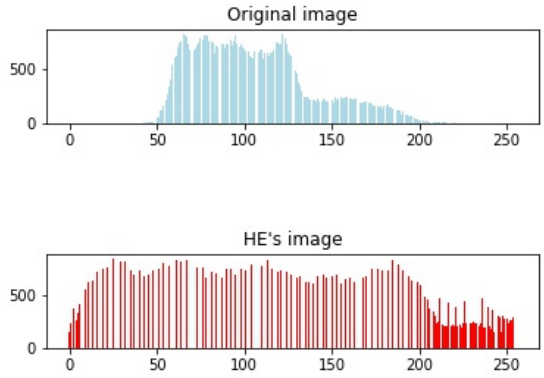
灰階 Histogram Equalization

- **Histogram:**

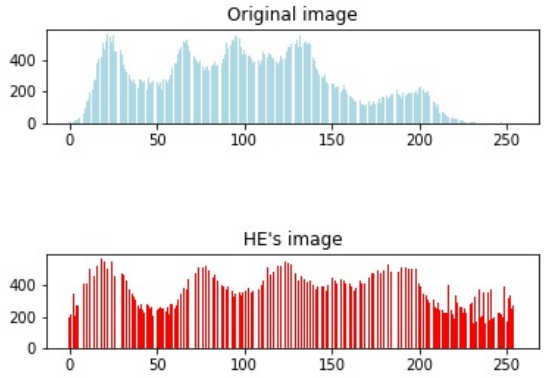
(a) 灰階



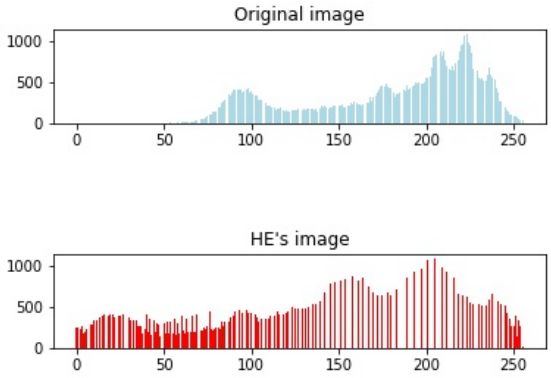
(b) 彩圖 R



(c) 彩圖 G



(d) 彩圖 B



- **Experience:**

  這次作業花了兩天的時間研究，尤其在弄懂 HE 的原理，過程中我不斷優化自己程式碼，也使用 PIL 內建函式 (ImageOps.equalize) 與自己的結果進行比較。另外我有用長條圖繪製每張圖片的分布，我們可以明顯發現經過 Histogram Equalization 後的分布更加均衡，範圍也變得比較廣，與老師 PPT 上的範例相同。最後感謝助教半夜的回信，讓我在崩潰的一整天後，幫助我了解 PPT 最後一段到底要我們做什麼。如果助教想看 jupyter notebook 版本可以到我的 GitHub (點此連結) 上觀看。