

(1) Assignment statement

Homework 12 (Due: 5.30)

Perform contrast enhancements for a color image.

- (1) Input a *RGB* color image;
- (2) Scale its *RGB* pixel values to *rgb* values,
where $0 \leq r, g, b \leq 1$;
- (3) Transfer *rgb* pixel values into *hsi* values;
- (4) Apply the histogram equalization algorithm to the *i* component to obtain the *i'* component;
- (5) Transfer the *hsi'* values into *r'g'b'* values;
- (6) Scale *r'g'b'* values to *R'G'B'* values,
where $0 \leq R', G', B' \leq 255$;
- (7) Output the *RGB* and *R'G'B'* color images.

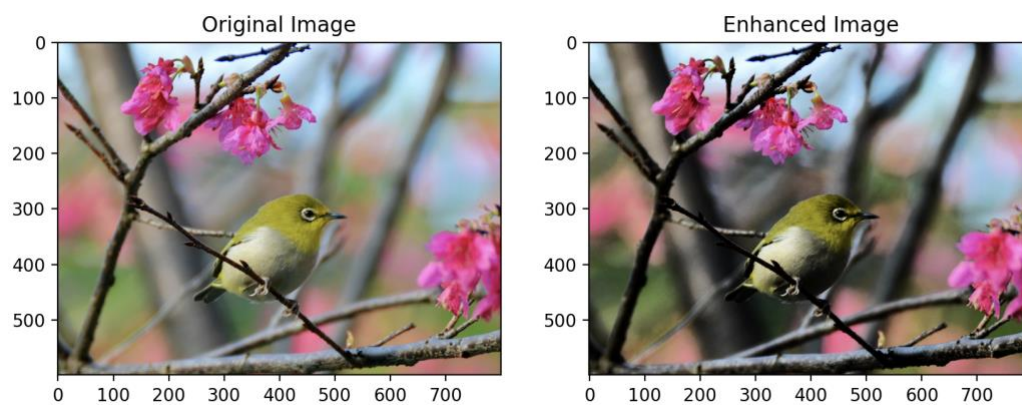
(2)

(a) Input/output images

Input



Output



(b) Source code

```
hw10.py > ...
1  import cv2
2  import numpy as np
3  from skimage import exposure, color
4  import matplotlib.pyplot as plt
5
6
7  image = cv2.imread('image.jpg')
8  image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
9
10 rgb = image / 255.0
11
12 hsi = color.rgb2hsv(rgb)
13
14 i = hsi[:, :, 2]
15 i_eq = exposure.equalize_hist(i)
16
17 hsi_eq = np.copy(hsi)
18 hsi_eq[:, :, 2] = i_eq
19 rgb_eq = color.hsv2rgb(hsi_eq)
20
21 rgb_eq = (rgb_eq * 255).astype(np.uint8)
22
23 cv2.imwrite('output_image.jpg', cv2.cvtColor(rgb_eq, cv2.COLOR_RGB2BGR))
24 plt.figure(figsize=(10, 5))
25
26 plt.subplot(1, 2, 1)
27 plt.title('Original Image')
28 plt.imshow(image)
29
30 plt.subplot(1, 2, 2)
31 plt.title('Enhanced Image')
32 plt.imshow(rgb_eq)
33
34 plt.show()
```

(c) Comments

這次作業我繞了彎路。我原本以為要自己寫函式轉換 RGB 跟 HSI，就花了一些時間手刻函式，做完後才發現其實可以用 `skimage` 套件的 `color.rgb2hsv()`、`color.hsv2rgb()` 函式，由於使用現成函式能讓程式碼更簡潔，所以我最後改用套件的函式。另外，我在寫程式時，也真心感覺到 `numpy` 套件的方便。無論是其中的函式或是 `numpy array` 的處理方式，都能夠讓程式碼更精簡好讀。而且因為有些地方可以用 `np array` 的 `index` 處理 `array` 中的值，而不是使用迴圈處理，所以程式跑起來的速度也能夠變得較快。