

Cover Page

影像處理作業3 - Color image enhancement

409410019 資工四 王郁誠

HW due : 6/7 00:00

HW handed in : 6/6

程式執行流程：

執行 `python hw3.py` 後，會先看到aloe.jpg的執行結果，接著取消註解69~71行，並根據該行行末註解替換第5行的factors，再重新執行，即可以看到其他圖片的執行結果。

Technical description

實作 RGB to HSI

```
def RGB2HSI(img: np.ndarray):
    height, width, channel = img.shape
    hsi_img = np.zeros((height, width, 3), dtype=np.float64)
    for i in range(height):
        for j in range(width):
            r, g, b = img[i, j] / 255.0
            num = 0.5 * ((r - g) + (r - b))
            den = np.sqrt((r - g) ** 2 + (r - b) * (g - b))
            theta = np.where(den == 0, 0, np.arccos(num / den))
            H = np.where(b <= g, theta, 2*np.pi-theta)
            S = 1 - (3 / (r + g + b)) * min(r, g, b)
            I = (r + g + b) / 3.0
            hsi_img[i, j, 0] = H
            hsi_img[i, j, 1] = S
            hsi_img[i, j, 2] = I
    return hsi_img
```

遍歷img內所有pixel，先將他們標準化(值歸到[0,1])，接著套PPT公式(CH6 P.22)，算出各點之 θ (弧度)，再進而算出該pixel之 H, S, I ，並將它們儲存到新圖(hsi_img)裡面。

實作 HSI enhance

```
def HSI_enhance(hsi_img: np.ndarray):
    enhanced_hsi_img = hsi_img.copy()
    enhanced_hsi_img[:, :, 1] *= S_factor
    enhanced_hsi_img[:, :, 1] = np.clip(enhanced_hsi_img[:, :, 1], 0, 1)
    enhanced_hsi_img[:, :, 2] *= I_factor
    enhanced_hsi_img[:, :, 2] = np.clip(enhanced_hsi_img[:, :, 2], 0, 1)
    return enhanced_hsi_img
```

先將input image複製到enhanced_hsi_img，接著對S, I channel分別乘以各自的factor，將超出範圍([0,1])的裁切掉，再回傳。

實作 HSI to RGB

```
def HSI2RGB(hsi_img: np.ndarray):
    height, width, channel = hsi_img.shape
    rgb_img = np.zeros((height, width, 3), dtype=np.float64)
    for i in range(height):
        for j in range(width):
            H, S, I = hsi_img[i, j]
            if 0 <= H < (2/3)*np.pi:
                B = I * (1 - S)
                R = I * (1 + (S * np.cos(H) / np.cos((1/3)*np.pi - H)))
                G = 3 * I - (R + B)
            elif (2/3)*np.pi <= H < (4/3)*np.pi:
                H -= (2/3) * np.pi
                R = I * (1 - S)
                G = I * (1 + (S * np.cos(H) / np.cos((1/3)*np.pi - H)))
                B = 3 * I - (R + G)
            elif (4/3)*np.pi <= H < 2*np.pi:
                H -= (4/3)*np.pi
                G = I * (1 - S)
                B = I * (1 + (S * np.cos(H) / np.cos((1/3)*np.pi - H)))
                R = 3 * I - (G + B)
            rgb_img[i, j, 0] = R * 255
            rgb_img[i, j, 1] = G * 255
            rgb_img[i, j, 2] = B * 255
            rgb_img[i, j, 0] = np.clip(rgb_img[i, j, 0], 0, 255)
            rgb_img[i, j, 1] = np.clip(rgb_img[i, j, 1], 0, 255)
            rgb_img[i, j, 2] = np.clip(rgb_img[i, j, 2], 0, 255)

    return rgb_img.astype(np.uint8)
```

遍歷img內所有pixel，先將各個pixel之(H,S,I)取出，接著按照PPT公式(CH6 P.23,24)，計算出轉換完的(R,G,B)，將他們去標準化後(剛剛標準化除以255，現在乘回來)，存進rgb_img，並將超過範圍的值裁切掉。

實作 Lab enhance

```
def Lab_enhance(img: np.ndarray):  
    enhanced_img = img.copy()  
    enhanced_img[:, :, 0] *= L_factor  
    # enhanced_img[:, :, 0] = np.clip(enhanced_img[:, :, 0], 0, 100)  
    return enhanced_img
```

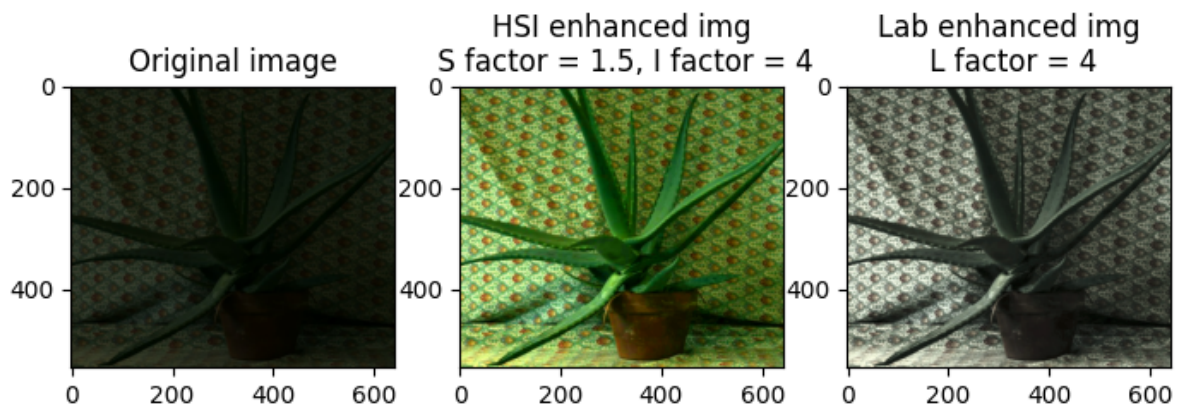
將input image複製到enhanced_img，接著把L channel的值乘上factor，再回傳。

Experimental results

因為L*a*b*色彩模型的a*,b*代表色彩，調了就會破壞原本圖片色彩比例，所以我在Lab強化中只調整了L*的部分。

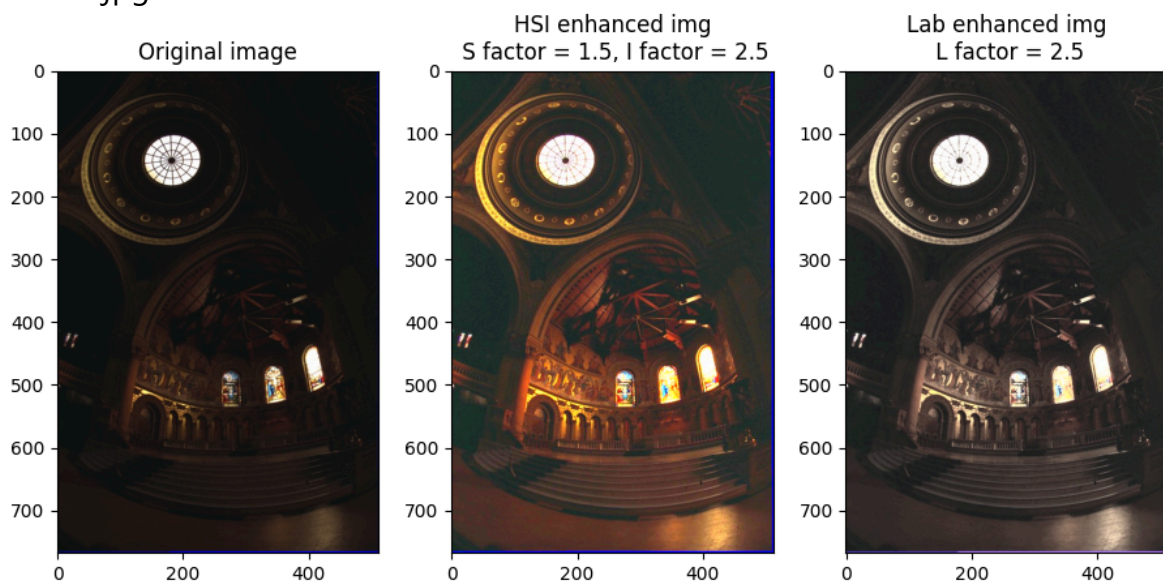
執行程式後，會看到一個視窗"result"，裡面由左至右為<原圖, HSI強化圖, Lab強化圖>

- aloe.jpg



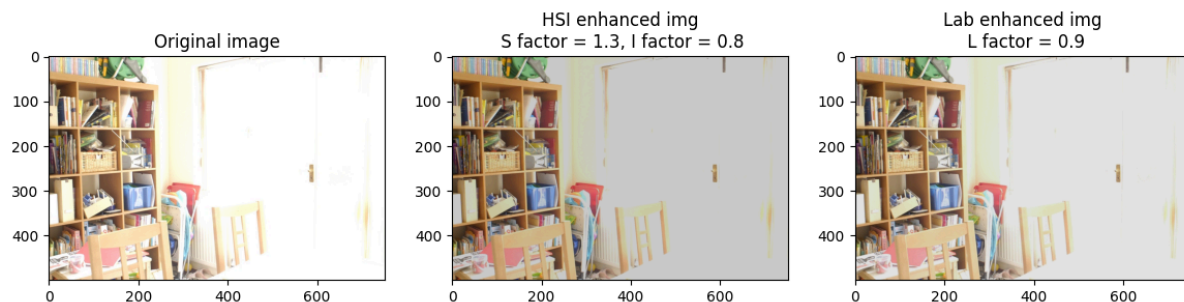
此case中，HSI在亮度&飽和度的強化效果明顯，Lab則在亮度表現挺好。而Lab在亮度部分的強化也不賴，是我這四張測試圖強化下來最滿意的一張。

- church.jpg



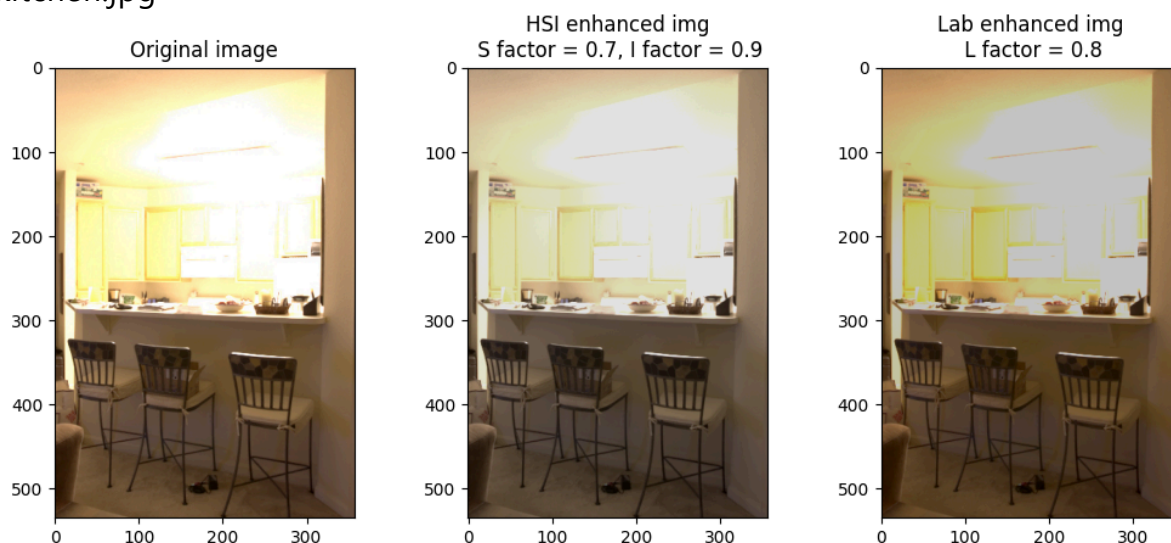
此case中，HSI在亮度&飽和度的強化效果也蠻好，而Lab在亮度部分的強化也還行，許多細節都更清楚了。

- house.jpg



此case中，HSI飽和度強化使圖片色彩更飽滿了些，而調低亮度對於中間超級過曝部分的調整比較還好，因為整體圖片都條暗，導致中間部分還是相對的過曝。Lab強化中亮度factor只用0.9，感覺有比HSI強化的效果好一點點。

- kitchen.jpg



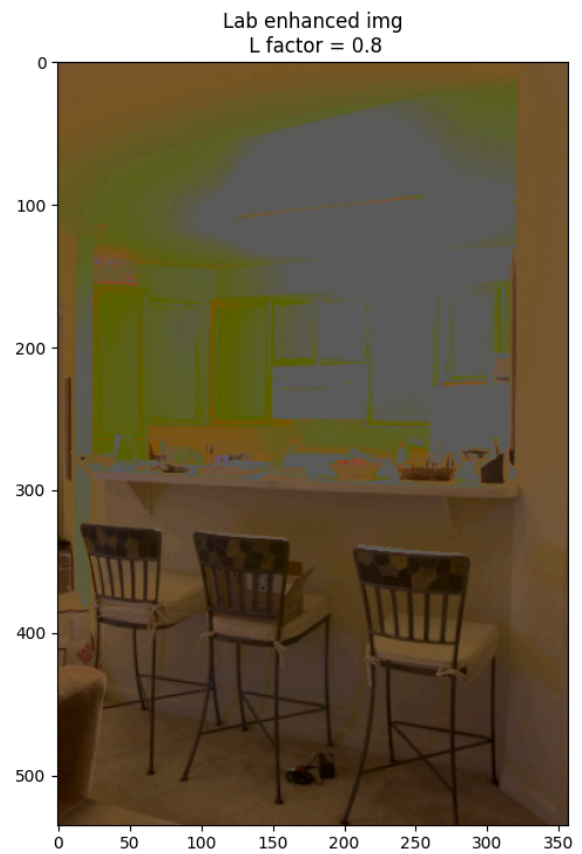
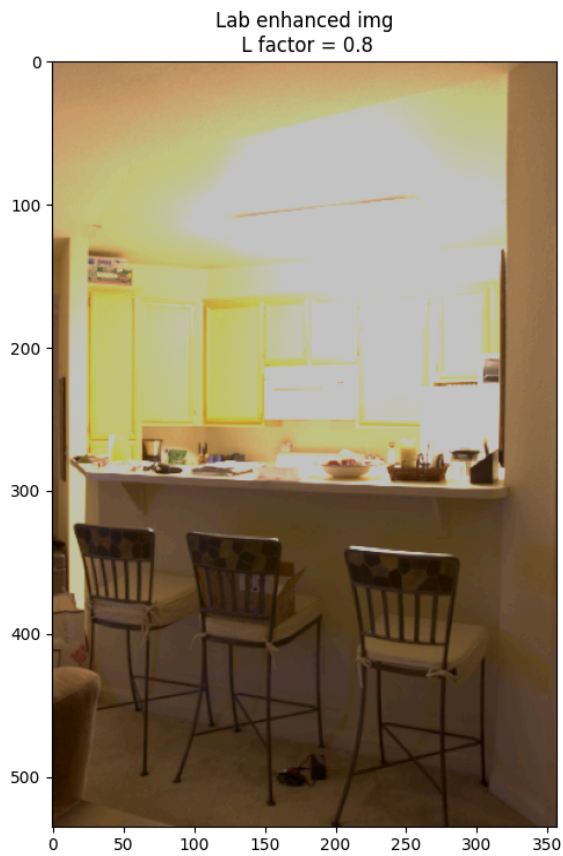
此case中，HSI強化中將飽和度及彩度個別調低一些，變成有點灰濛濛的版本，而Lab強化中將亮度*0.8倍，感覺有很稍微的減少一些原圖過曝的部分。

Discussions & Appendix

這次作業我寫下來，我覺得HSI整體比較好進行調整，因為 $L^*a^*b^*$ 若要調整飽和度，就要調 a^*b^* 的比例，但我每次調出來都很詭異，所以我放棄了，還是HSI調得舒服，aloe.jpg跟church.jpg用HSI調的真的很漂亮，讓我心情很好。

至於這次作業有個地方我覺得很怪，就是Lab enhance()的地方，照理說應該要像HSI enhance()那樣，在乘上factor後，需要將值給clip到合理的範圍，但若我這麼做，Lab enhance()出來的圖反而會很詭異。

以kithen.jpg為例：



左、右圖分別為沒有、有使用`clip()`將亮度裁切到 $[0,100]$ 的結果，很明顯左圖就比較好看，右圖這亮度感覺就不只是乘了0.8倍，感覺至少被砍了一半，整個表現就很奇怪，所以我在實驗中使用的`Lab enhance()`都是沒有`clip`的，但在`HSI enhance`中使用`clip`就不會有這個問題，這是這次作業我最疑問的一個點。