

Intro. to Image Processing HW1 Report

110550088 李杰穎

March 14, 2023

1 Method

在本次作業中，我使用了 `cv2` 和 `numpy` 來對圖片進行操作。

1.1 Exchange Position

使用雙層迴圈和 `slice` 的方法去將兩個二維陣列的部份內容互相交換。其中複製使用 `shallow copy (.copy())` 而不是預設的 `deep copy`，使數值不會因為被修改而一同影響其他變數。

1.2 Gray Scale

灰階即是將 R, G, B 的數值取平均，要注意的是，因為 `cv2` 讀取圖片後的儲存資料型態是 `np.uint8`，所以如果直接對陣列中的內容相加會導致 `overflow`。必須先轉成 Python 中的 `int` 型態再進行運算才不會出現 `overflow` 的問題。

1.3 Intensity Resolution

我們先將圖片轉為灰階，這個步驟和前面相同。之後在透過取底的運算，先將灰階的數值除以 64 再取底，這樣算出來的數值一定是 0, 1, 2, 3 其中一個。之後再將這個數字乘以 64 就可以產生 Intensity Resolution 為 4 的灰階圖片。具體計算公式可以參考下面式子。

$$\text{value} = \left\lfloor \frac{R+G+B}{3} \right\rfloor \times 64 \quad (1)$$

1.4 Red Color Filter

一樣使用雙層迴圈去判斷該格的顏色是否符合投影片上的條件，如果沒有的話就轉成灰階。要注意的是透過 `cv2` 讀取的圖片，其 index 0, 1, 2 分別為 B, G, R。並非正常的 R, G, B，撰寫程式碼的時候需要特別小心。也要記得先將 `np.uint8` 轉換成內建的 `int` 格式。

1.5 Yellow Color Filter

與紅色濾鏡類似，不再贅述。

1.6 Channel Operation

綠色是對應到 index 1，故將每一像素的 index 1 乘以 2。但因為這樣的作法可能會造成 `overflow` 的問題，故在補上 `min(255, val*2)` 將數值限制在 255 內。

1.7 Bilinear Interpolation

雙線性和投影片和上課講授的相同，即是將最接近的四個點，先對 x 軸進行線性插值，分別插出兩個值。在用這兩個插值出的值，對 y 軸進行最後一步的線性插值，得到最後的結果。¹

1.8 Bicubic Interpolation

雙三次插值則是透過最近的 16 個點，透過近似的方式求導，並解出多條三次函數，再透過此三次函數插值出

¹此處提到的 x 軸為向下，y 軸則為向右。

中間的值。具體方式為先對 x 軸進行線性插值，產生四個三次函數，並產生四個相對應的數值。再利用這四個相對應的數值解出一條三次函數，最後再對 y 軸進行插值，得到最後的數值。要注意的是，不同於線性插值，三次插值的方式可能會使插值出來的數值不借於 0 255 間，導致 overflow。故會需要使用 `np.clip()` 函數，將陣列中的數值限制在 0 到 255 間。

2 Result

Figure 1 即為最終的結果。



Figure 1: 透過 Python 程式碼所產生的結果圖

3 Feedback

在本次作業中，我了解如何透過 `cv2` 和 `numpy` 操作圖片，也學到了 `bicubic` 這個常見的插值方式具體是怎麼運作的，也對 Python 更加熟悉。