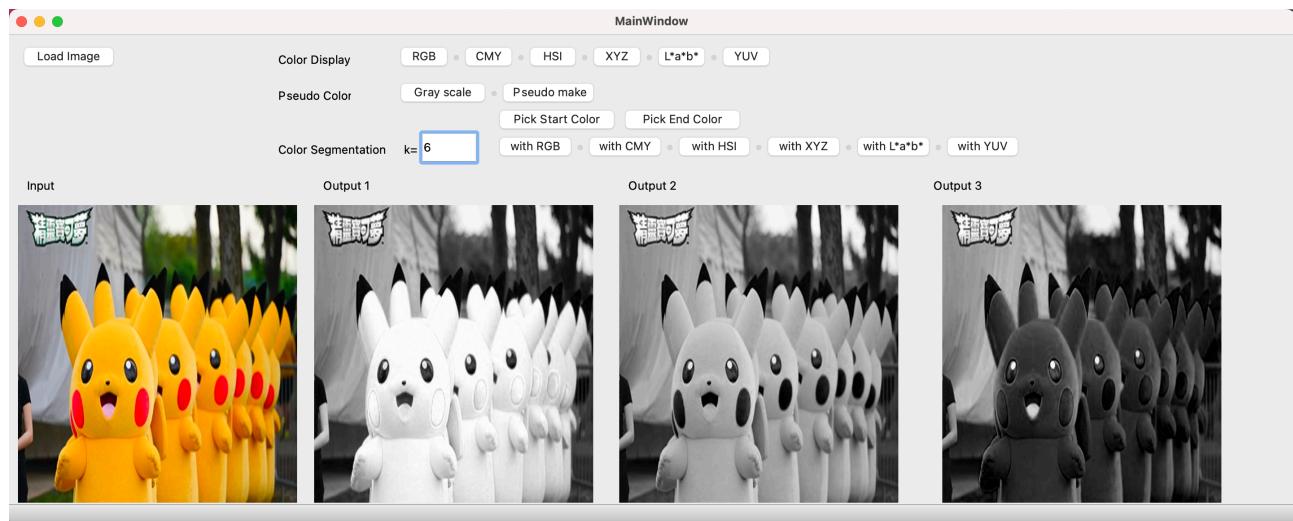


GUI介面圖：



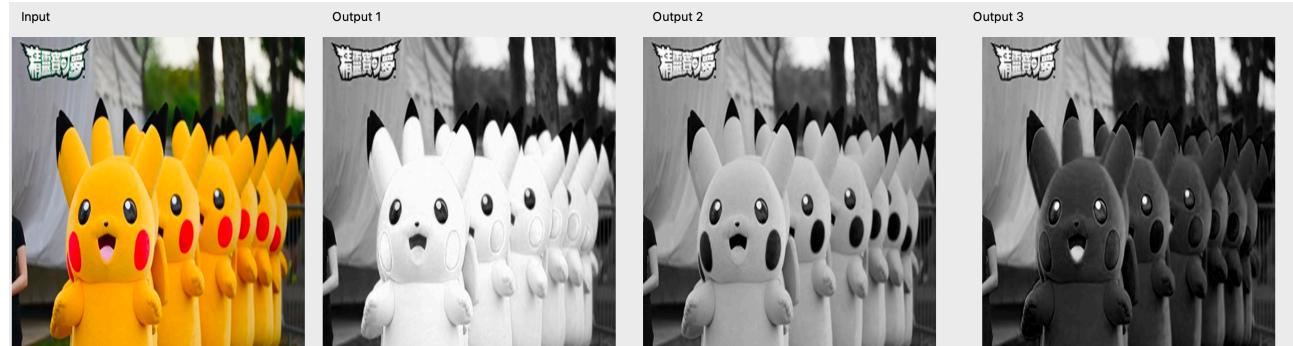
\* python hw5.py開啟主要程式

\*\* 使用時第一步必須先加載圖片

\*\*\* 欲使用Pseudo color須先按下 Start Color 與 End Color 選擇color bar的兩端顏色

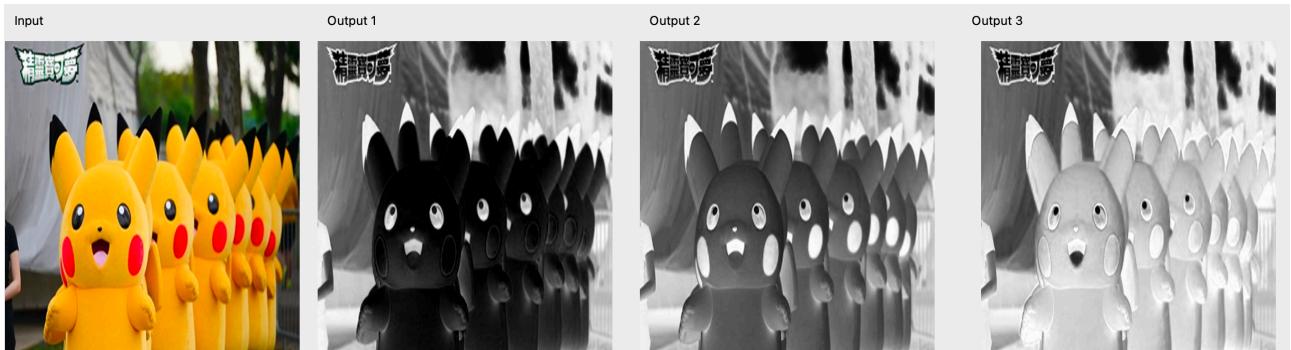
### Part 1:

#### RGB



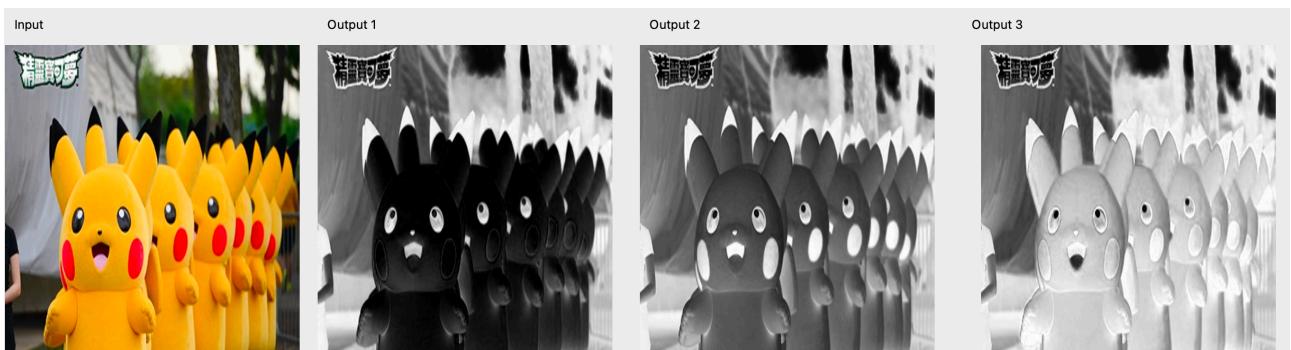
1. 轉換上使用 cv2.imread 讀檔後，將RGB通道分開，以獲得RGB成相圖
2. 由於圖片中紅色的元素最多，綠色次之，藍色最少，因此三張圖片看起來是R最亮，G次之，B最暗

## CMY



1. 轉換上使用 cv2.imread 讀檔後，將RGB通道分開，並取RGB的負片獲得
2. 由於CMY為RGB通道的負片，因此成相亮暗程度剛好跟RGB相反

## HSI



1. HSI三通道由由課本提供的公式獲得，詳細內容如下面方程式所示
2. 三個通道可視為色度,飽和度,明亮度的三個數值，在圖片中可以看到，接近紅色的臉頰部分，H角度極小，在第二張圖片中顏色接近黑色，又其飽和度較高，因此S值較大，在第三張圖片接近白色，最後其明亮度居中，因此在第三張圖片接近灰色

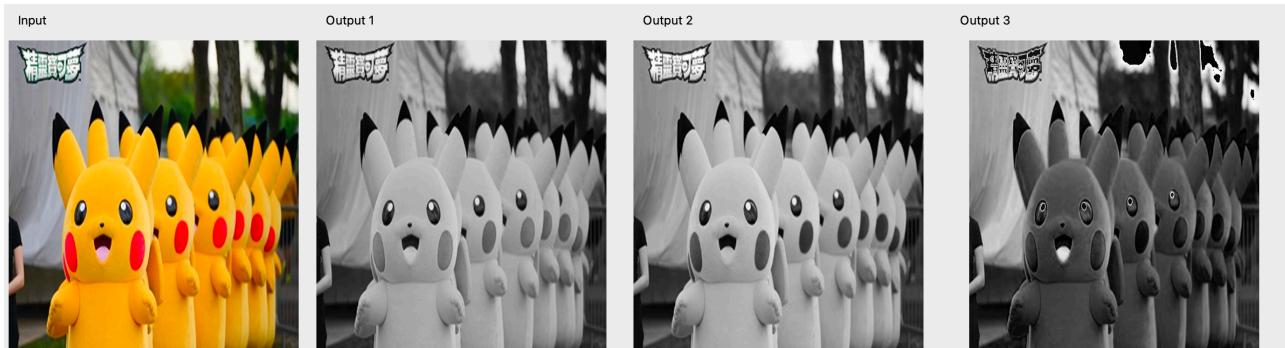
```

im_rgb = cv2.cvtColor(self.img, cv2.COLOR_BGR2RGB)

i = np.mean(im_rgb, axis=2) # 平均數
s = 1 - np.min(im_rgb, axis=2) / (i + 1e-6)
s[s < 0] = 0
R, G, B = im_rgb[:, :, 0]/255, im_rgb[:, :, 1]/255, im_rgb[:, :, 2]/255
th = np.arccos((2 * R - G - B) / 2 / (
    np.sqrt((R - G) ** 2 + (R - B) * (G - B)) + 1e-6))
h = th
h[B > G] = 2 * np.pi - h[B > G]
h = cv2.convertScaleAbs(h, alpha=255/h.max())
s = cv2.convertScaleAbs(s, alpha=255/s.max())
i = cv2.convertScaleAbs([i, alpha=255/i.max()])

```

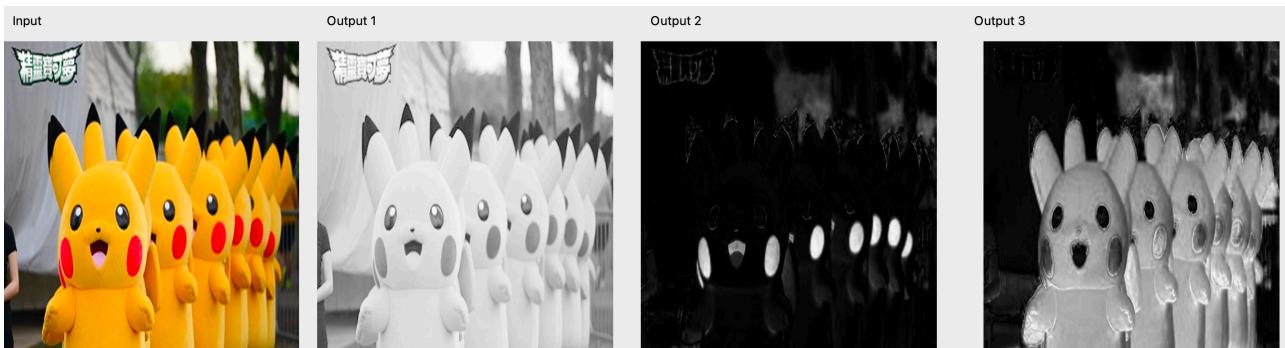
## XYZ



1. XYZ三通道由RGB通道透過轉移矩陣獲得，轉移矩陣由下圖所示
2. 由第四張圖可知。由於整張圖片缺乏藍色元素，所以整體圖片較暗

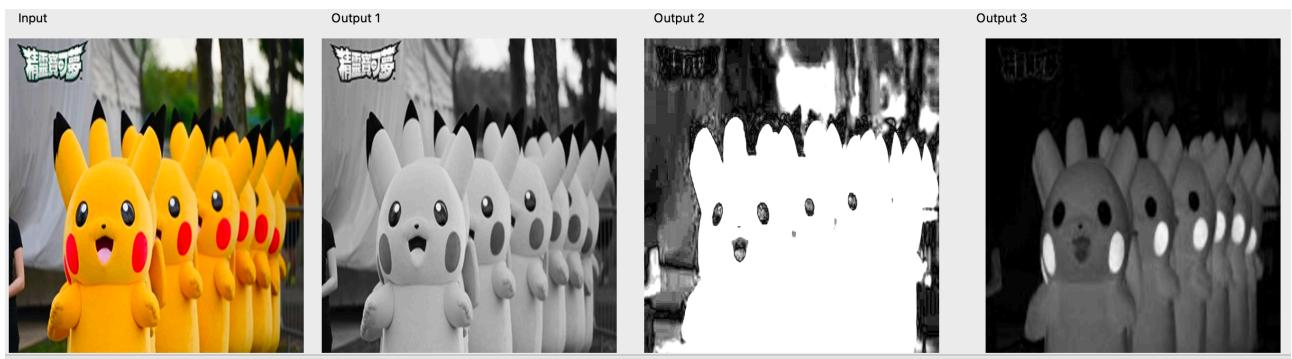
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

## La\*b\*



1. Lab三通道算法跟課本裡一樣，就不多寫了
2. 在L通道中，越明亮代表色彩越接近白色，越黑代表色彩越接近黑色，所以可以看到耳朵的地方是黑的
3. 在a\*通道中，越明亮越接近紅色，越暗代表越接近綠色，所以可以看到嘴巴跟臉頰的部分特別亮
4. 在b\*通道中，越亮越接近黃色，越暗越接近藍色，因此可以看到身體的部分特別明亮

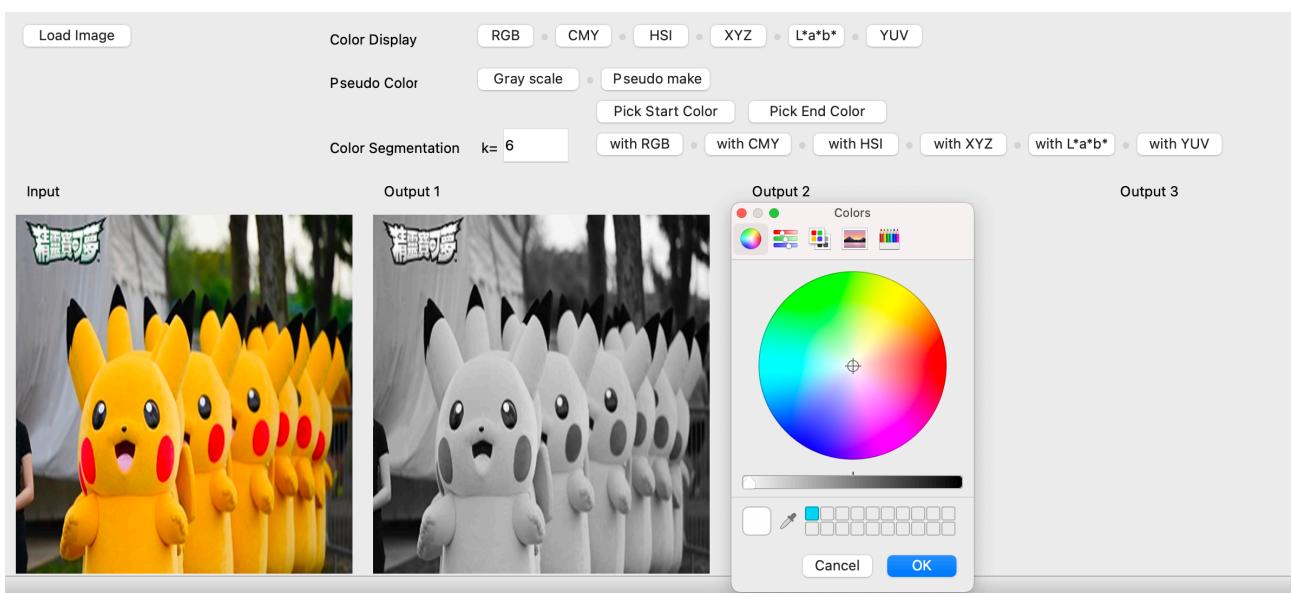
## YUV

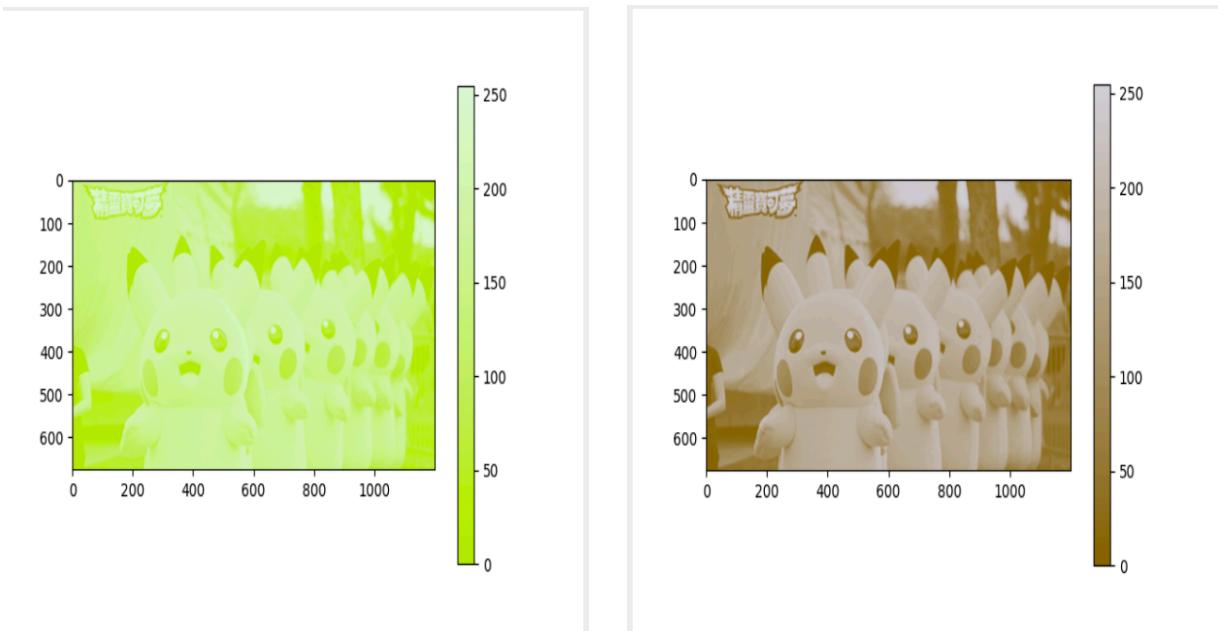


1. YUV三通道由RGB通道透過轉移矩陣獲得，轉移矩陣由下圖所示
2. 在Y通道中，Y表示明度，前面有很多張圖都有使用了，不多再敘述
3. 在U通道中，越明亮越接近黃色，越暗代表越接近藍紫色，所以可以看到身體的部分特別亮
4. 在V通道中，越亮越接近紅色，越暗越接近藍綠色，因此可以看到臉頰的部分特別明亮

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

## Part 2:

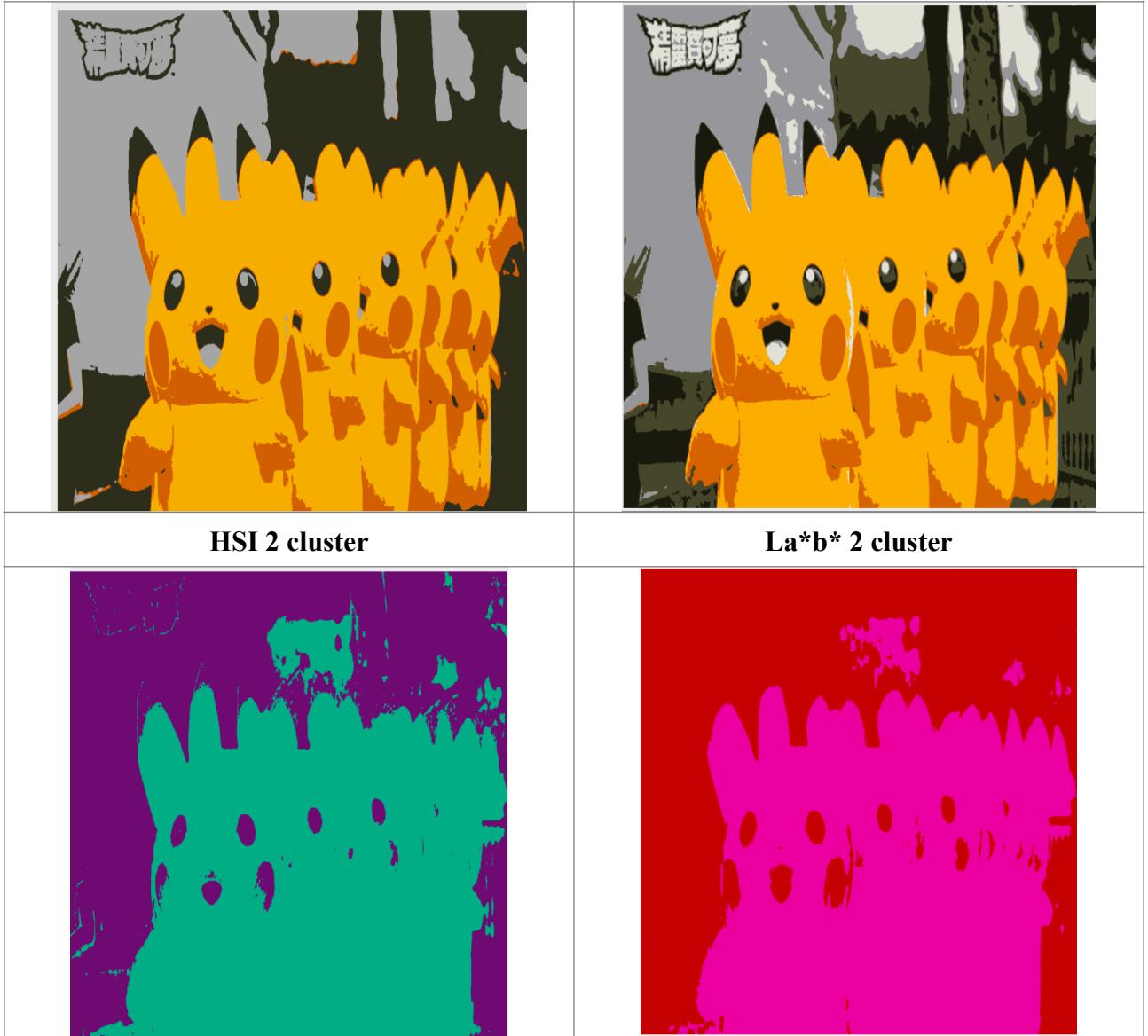




1. 透過選擇兩個顏色作為灰階最亮與最暗的顏色，然後轉到XYZ色彩空間進行內插，最後再將內插完的顏色map回灰階，就出現上方的圖片
2. 其實我不太懂這題到底想問什麼

### Part 3:

原圖	RGB 2 cluster
RGB 4 cluster	RGB 6 cluster



1. Kmeans 使用sklearn 套件完成，先將顏色轉到不同色域後，再透過Kmeans分類
2. 由RGB中的不同cluster數作為比較，可以看到cluster越多，圖片越接近原圖，並且在cluster為6時就已經可以將大部分的輪廓線切割出來了
3. 在cluster為2時，如果主要需區分的內容為五官，則可以看到La\*b\* 的效果最好，因為五官中眼睛為黑色，而臉頰與嘴巴接近紅色，這兩個顏色在a\*通道中都有很明顯的區分效果，因此作為五官分類的效果最佳。