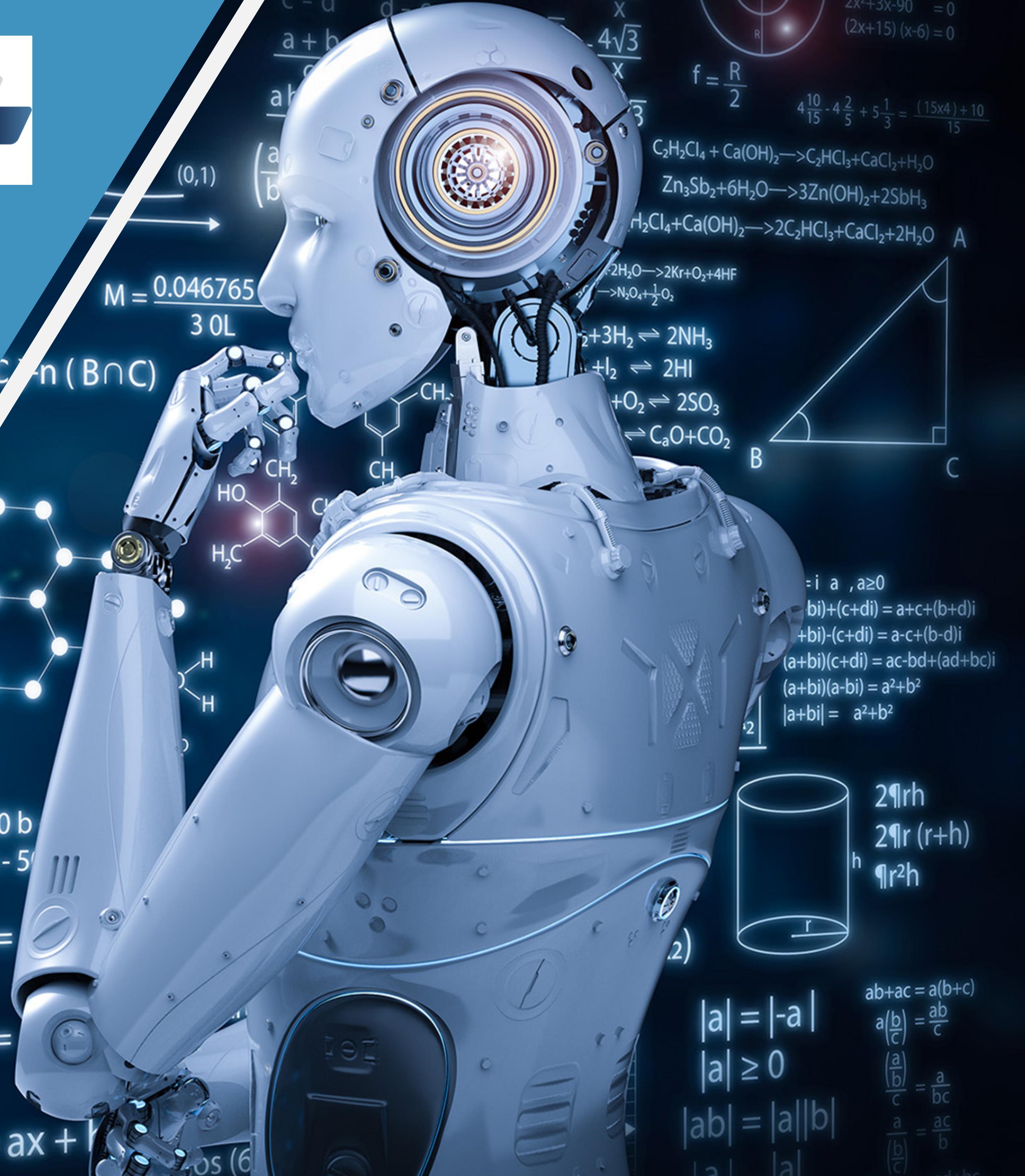




Day 03

深度學習與電腦視覺 學習馬拉松

Cupay 陪跑專家：楊鎮銘





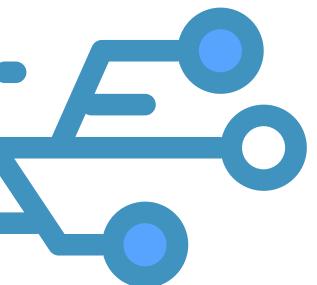
OpenCV 影像處理

顏色相關的預處理



重要知識點

- 了解 color space 的轉換與改圖的相關操作
- 了解 histogram equalization 的概念



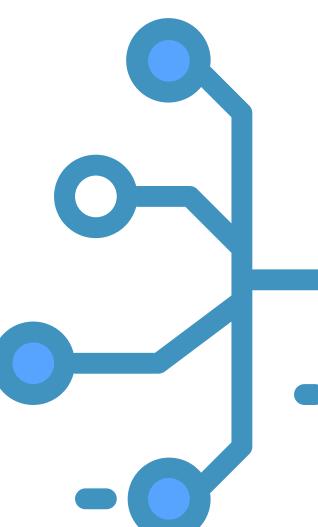
OpenCV 實作 - 透過 color space 轉換改圖

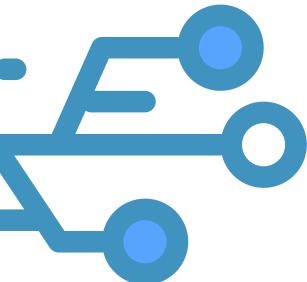


前面介紹了多種顏色的表示方式，以及如何透過 OpenCV 去做不同 color space 之間的轉換

有時候為了做 Data Augmentation 也會透過 color space 的轉換去更改我們想要改變的顏色屬性 (e.g. 飽和度, 明亮度)

舉例來說：針對圖片明亮度做調整，會考慮把圖片轉到 HSV (或是其他有明亮度的 color space) 再去調整明亮度的值

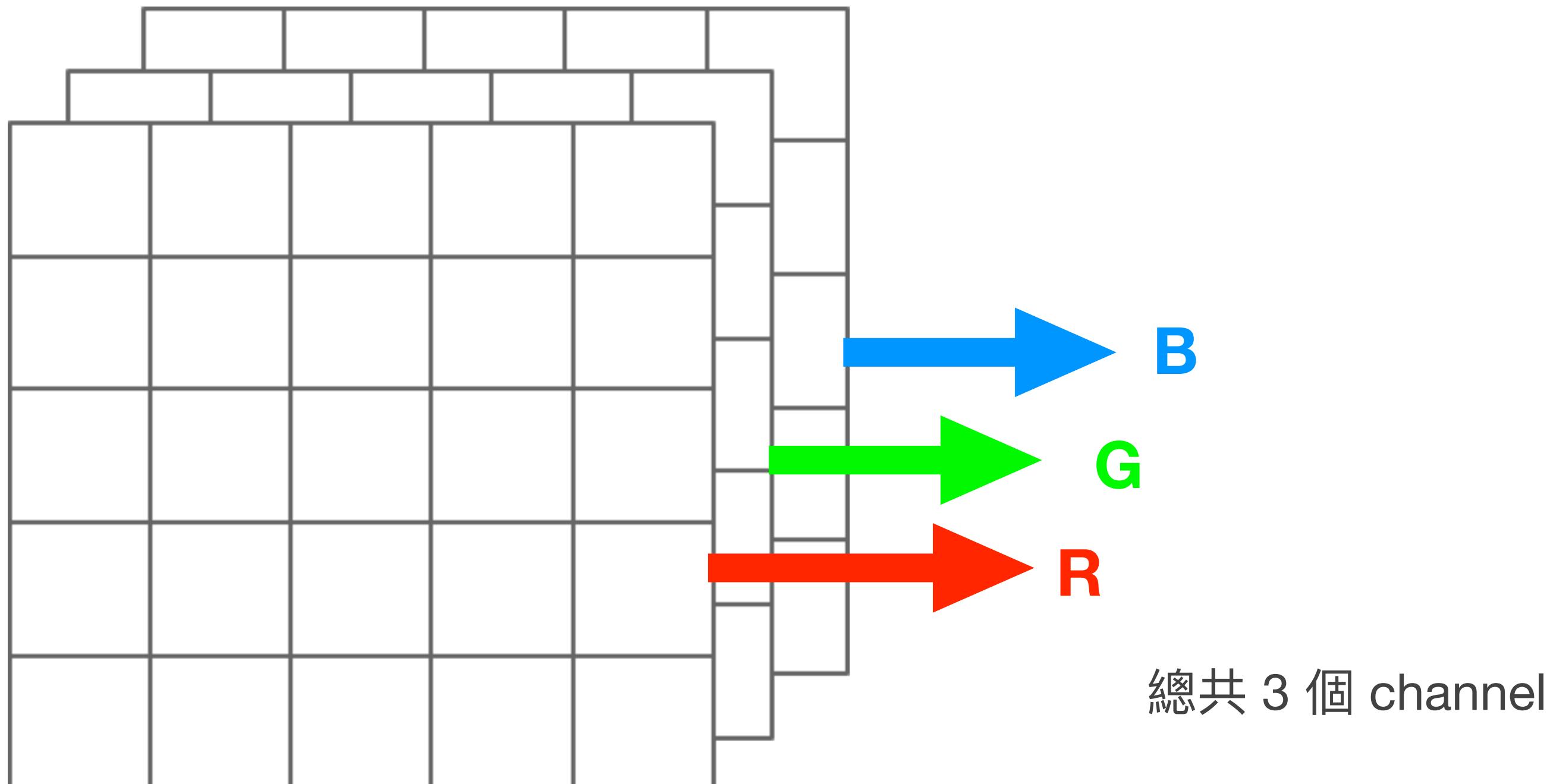




OpenCV 實作 - 指定要操作的 channel



以 5*5 的 RGB 圖片舉例，我們通常稱呼 RGB 各為一個 channel



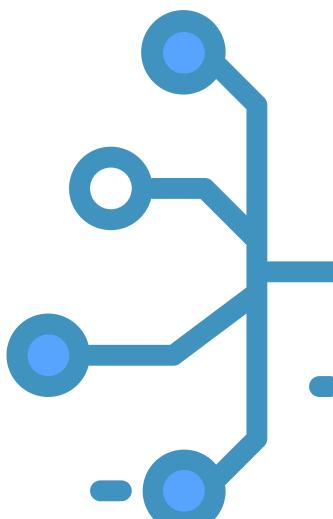
假設我們要指定 G channel

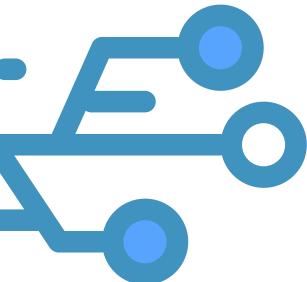
因為 Python 是從零開始數
所以 G channel = 第 1 個 channel

寫法會是 `img[:, :, 1]`
表示取得第 1 個 channel 的
row*column 的值

Optional: 這邊有另外一個等價的寫法
`img[..., 1]`

同理可證：假如是要做飽和度的操作，轉成 HSV 之後也是要取第 1 個 channel 當作目標



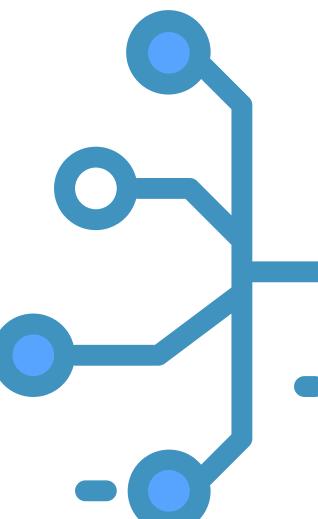


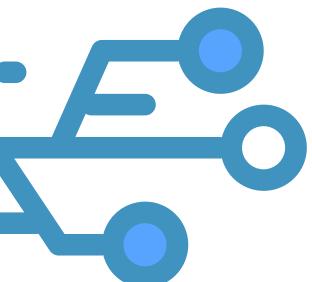
OpenCV 實作 - color space 範圍



在我們選定要操作的範圍後有些事情需要特別注意

- OpenCV 會以 0~255 的值表示
 - 有些值是以百分比或是包含負值的方式表示
- OpenCV 顯示圖片會以 uint8 (可儲存範圍 0~255) 方式表示
 - 如果計算過程包含小數或是負值要自己轉換





OpenCV 實作 - color space 範圍 (範例: 飽和度)

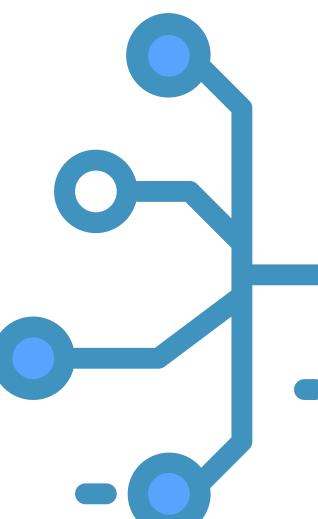


```
# 轉換 color space (BGR -> HSV)
img_hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

# 操作飽和度時候需要轉成百分比 (小數點)
img_hsv = img_hsv.astype('float32')

# 轉成小數點並調降兩成飽和度
# 除以 255 之後範圍就變成
img_hsv[..., 1] = img_hsv[..., 1]/255 - 0.2

# 後面在做一些邊緣條件判斷，確保所有的值都在合理範圍 (0~1)
# 在把圖片依序轉回 0~255 區間 > uint8 格式 > BGR 格式 就可以顯示圖片了
...
```





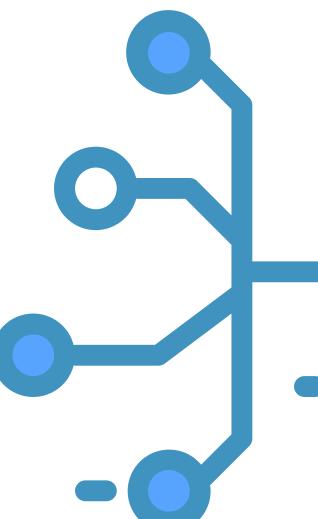
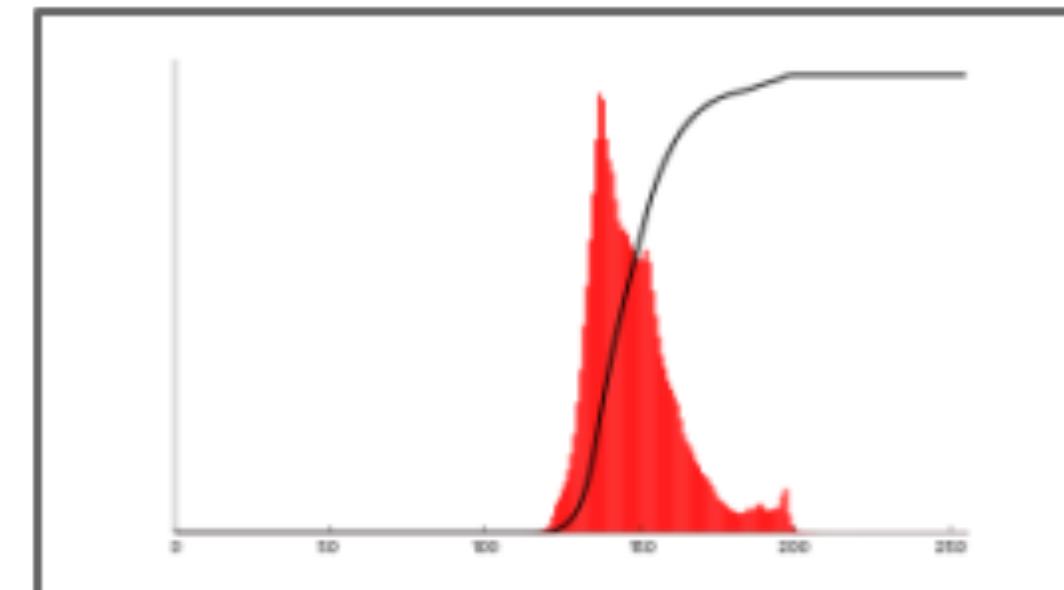
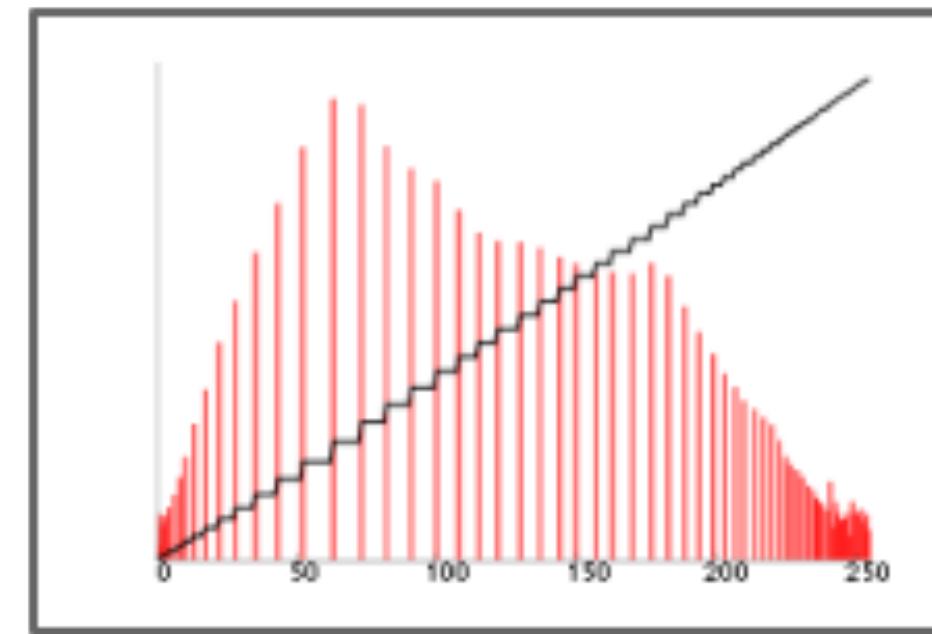
直方圖均衡 (Histogram Equalization)



當我們想增強對比的時候，可以考慮使用直方圖均衡的方式

- 對比不強：顏色的值集中在某個區域
- 增強對比：透過數學公式將值分散到 0~255 區間

Note: OpenCV 實作的 function `cv2.equalizeHist(img)`



直方圖均衡 (Histogram Equalization)



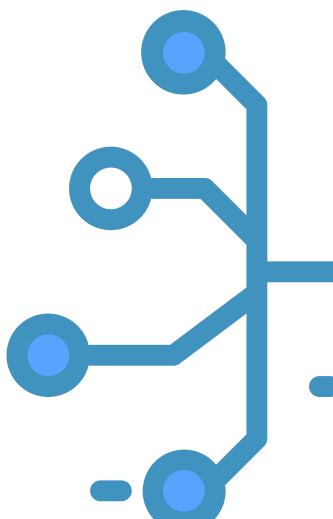
雖然 [Wiki](#) 中有提到直方圖均衡主要是處理灰圖

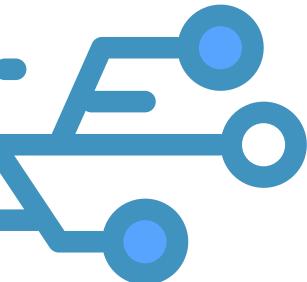
如果處理 RGB 圖通常是會轉到 HSV 再對明亮度做直方圖均衡

不過我們也可以個別對 RGB 的 3 個 channel 做直方圖均衡



看的出來哪張是「對明亮度處理」哪張是「RGB 直接處理」嗎？





調整對比度 / 明亮度

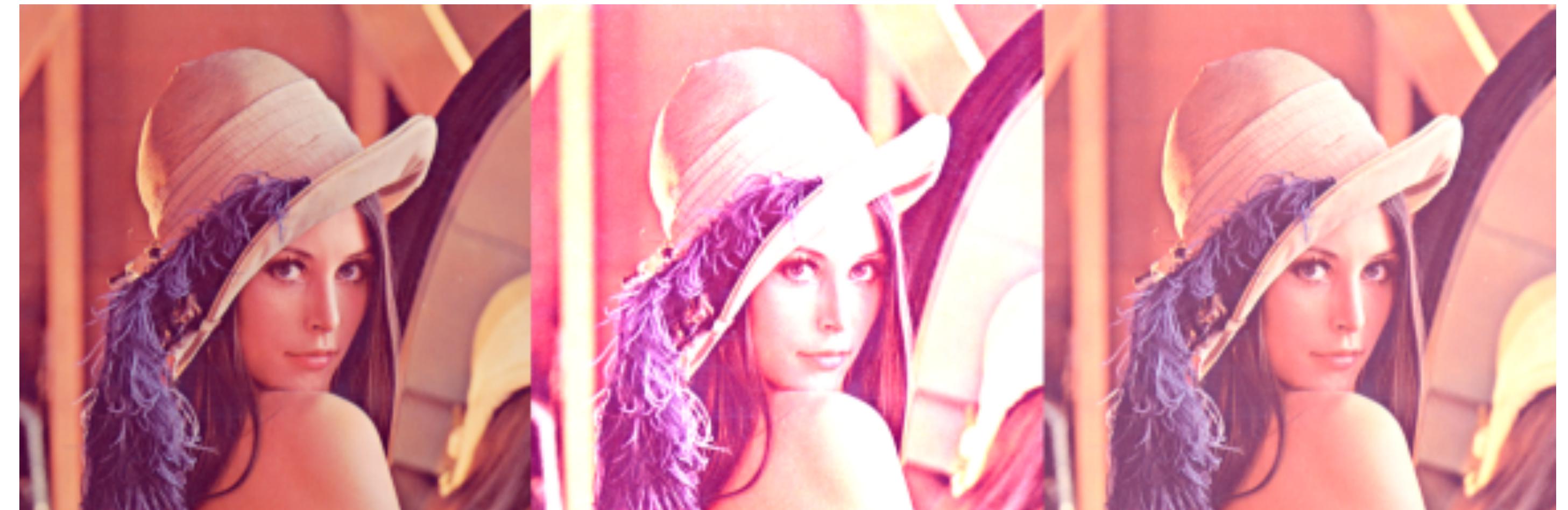


- 直方圖均衡是參考全圖做對比度跟明亮度的調整
- 我們如果要手動調整可以簡單透過乘法跟加法來改變

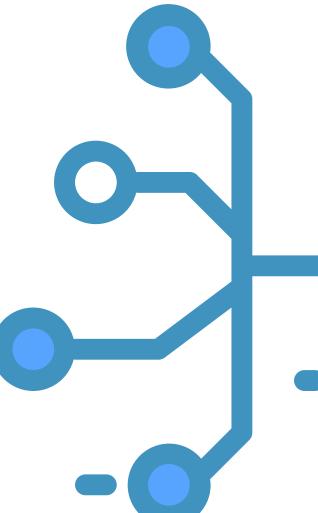
$$g(x) = \alpha * f(x) + \beta$$

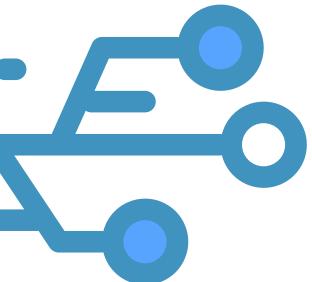
- $f(x)$ 代表圖片原本的 pixel 值
- $g(x)$ 代表調整過後的 pixel 值
- alpha 調整對比度 ($\alpha > 0$)
- beta 調整明亮度

因為圖片的值限定在 0~255 之間
所以參數值不能太大，參考官方範例
 α (1.0~3.0), β (0~100)



分別只調整明亮度與對比度的結果





實作調整對比度 / 明亮度

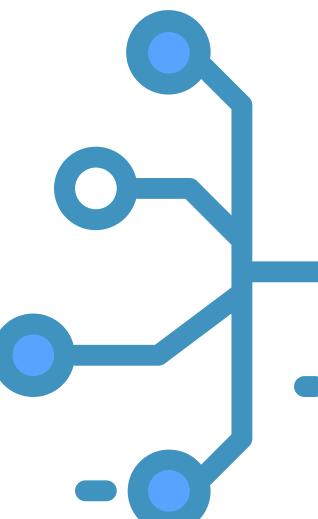


我們可以透過 for loop 逐一對每個 pixel 調整

```
for b in range(img.shape[0]):  
    for g in range(img.shape[1]):  
        for r in range(img.shape[2]):  
            # 透過 np.clip 限制範圍在 0~255  
            img[b,g,r] = np.clip(alpha * img[b,g,r] +  
beta, 0, 255)
```

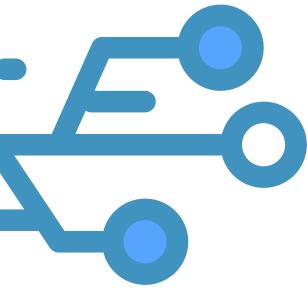
或是透過 OpenCV 內建的 function 來做

```
img = cv2.convertScaleAbs(img, alpha=alpha, beta=beta)
```



知識點回顧

- 了解實作上轉換 color space 來調整圖片屬性的相關細節
- 了解調整對比 / 明亮有多種方式可以實作
 - * 直方圖均衡
 - * alpha / beta correlation



推薦延伸閱讀



Histogram Equalization [連結](#)

該 blog 詳細講解 Histogram Equalization 的數學式，並且有圖幫助了解

Jason Chen's Blog

HOME ABOUT PRESS CONTACT

【影像處理】灰階直方圖均化 Histogram Equalization

1/10/2019 0 評論

Jason Chen

如果說懷才就像懷孕，時間久了才能讓人看出來。那麼還有另一個相似的點就是..在開花結果之前，都要先給人幹到不要不要的Q.Q

文章分類

- 全部
- 介紹
- 分享
- 加密貨幣 Cryptocurrency
- 影像處理 DIP
- 教學
- 消息
- 演算法 Algorithm
- 物聯網 IoT
- 資料科學 Data Science
- 問題

一、灰階直方圖 Histogram

既然要做Histogram Equalization 那麼最基本的，首先要了解什麼叫作 Histogram 吧！

OpenCV 官方文件 [連結](#)

裡面有額外提到一種調整亮度的方式叫做 gamma correlation，相較於 alpha/beta 的調整，他對飽和度的影響會比較低

OpenCV 3.4.8-dev

Main Page Related Pages Modules Namespaces Classes Files Examples Java documentation

OpenCV Tutorials / The Core Functionality (core module)

Changing the contrast and brightness of an image!

Prev Tutorial: Adding (blending) two images using OpenCV
Next Tutorial: Discrete Fourier Transform

Goal

In this tutorial you will learn how to:

- Access pixel values
- Initialize a matrix with zeros
- Learn what cv::saturate_cast does and why it is useful
- Get some cool info about pixel transformation
- Improve the brightness of an image on a practical example

Theory

Note: The explanation below belongs to the book Computer Vision: Algorithms and Applications by Richard Szeliski.

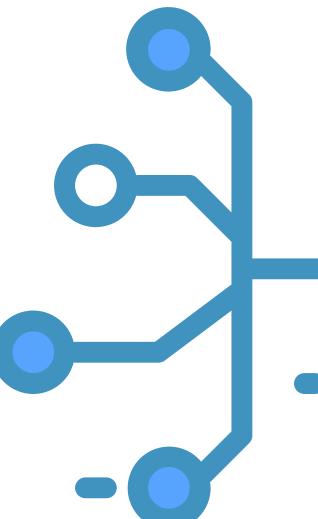
Image Processing

- A general image processing operator is a function that takes one or more input images and produces an output image.
- Image transforms can be seen as:
 - Point operators (pixel transforms)
 - Neighborhood (area-based) operators

Pixel Transforms

- In this kind of image processing transform, each output pixel's value depends on only the corresponding input pixel value (plus, potentially, some globally collected information or parameters).
- Examples of such operators include brightness and contrast adjustments as well as color correction and transformations.

Brightness and contrast adjustments



解題時間 Let's Crack It



請跳出 PDF 至官網 Sample Code & 作業開始解題