

第4章

認識色彩空間到藝術創作

4-1：BGR與RGB色彩空間的轉換

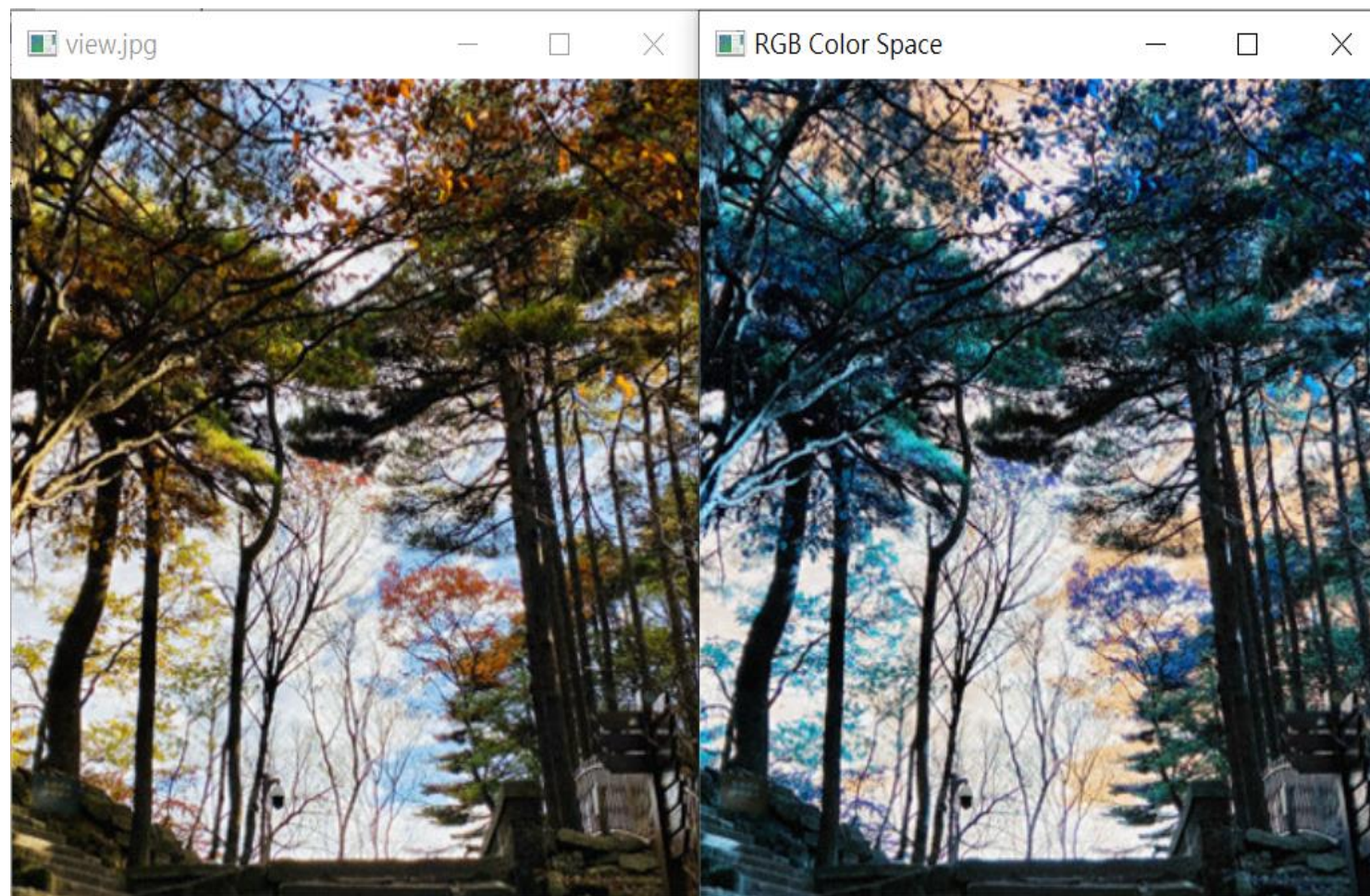
- `image = cv2.cvtColor(src, code)`

- `src`：要轉換的影像物件。

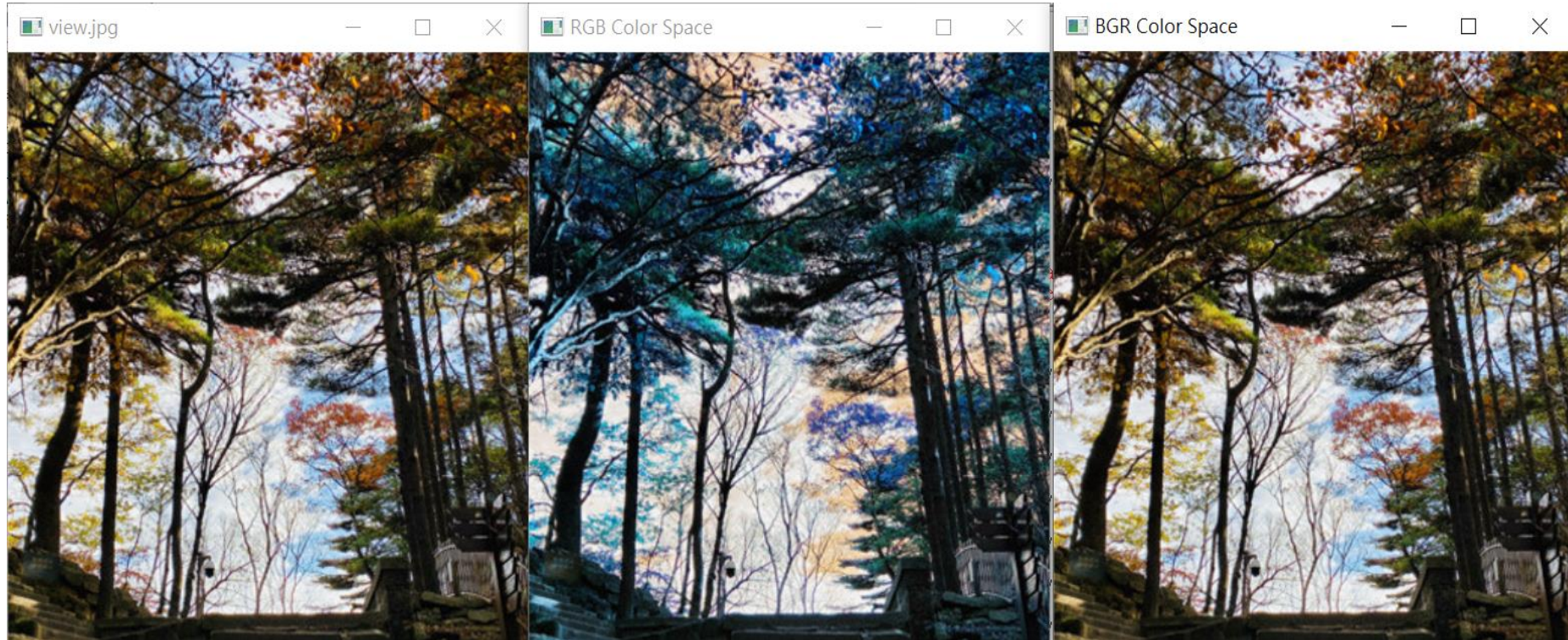
- `code`：色彩空間轉換具名參數，下列是常見的參數表。

| 具名參數 | 值 | 說明 |
|-----------------------|------------------|----------------------|
| COLOR_BGR2BGRA | 0 | 影像從 BGR 色彩轉為 BGRA 色彩 |
| COLOR_RGB2RGBA | =COLOR_BGR2BGRA | 與上一項相同 |
| COLOR_BGRA2BGR | 1 | 影像從 BGRA 色彩轉為 BGR 色彩 |
| COLOR_RBGA2RGB | =COLOR_BGRA2BGR | 與上一項相同 |
| COLOR_BGR2RGBA | 2 | 影像從 BGR 色彩轉為 RGBA 色彩 |
| COLOR_RGB2BGRA | =COLOR_BGR2RGBA | 與上一項相同 |
| COLOR_RGBA2BGR | 3 | 影像從 RGBA 色彩轉為 BGR 色彩 |
| COLOR_BGRA2RGB | =COLOR_RGBA2BGR | 影像從 BGRA 色彩轉為 BGR 色彩 |
| COLOR_BGR2RGB | 4 | 影像從 BGR 色彩轉為 RGB 色彩 |
| COLOR_RGB2BGR | =COLOR_BGR2RGB | 影像從 RGB 色彩轉為 BGR 色彩 |
| COLOR_BGR2GRAY | 6 | 影像從 BGR 色彩轉為 GRAY 色彩 |
| COLOR_RGB2GRAY | 7 | 影像從 RGB 色彩轉為 GRAY 色彩 |
| COLOR_GRAY2BGR | 8 | 影像從 GRAY 色彩轉為 BGR 色彩 |
| COLOR_GRAY2RGB | = COLOR_GRAY2BGR | 與上一項相同 |
| COLOR_BGR2HSV | 40 | 影像從 BGR 色彩轉為 HSV 色彩 |
| COLOR_RGB2HSV | 41 | 影像從 BGR 色彩轉為 HSV 色彩 |
| COLOR_HSV2BGR | 54 | 影像從 HSV 色彩轉為 BGR 色彩 |
| COLOR_HSV2RGB | 55 | 影像從 HSV 色彩轉為 RGB 色彩 |

- 程式實例 `ch4_1.py`：讀取彩色影像 `view.jpg`，然後將此影像轉成 RGB 影像。



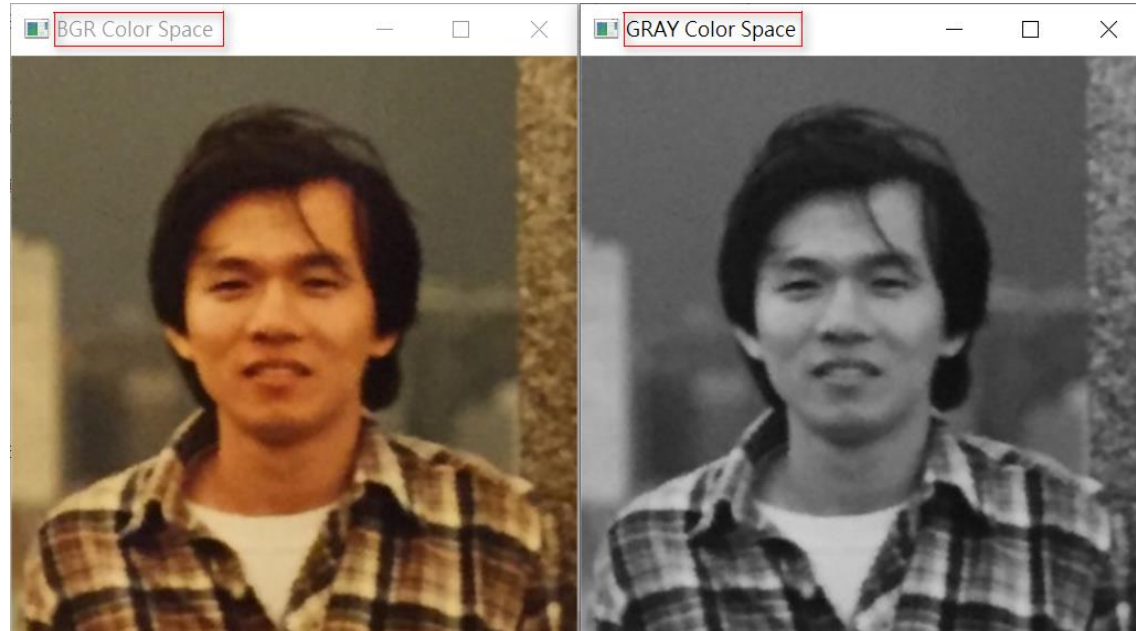
- 程式實例ch4_2.py：繼續ch4_1.py將RBG影像轉回BGR影像。



- 程式實例 ch4_3.py：重新設計 ch4_2.py，將第 8 列的 `COLOR_RGB2BGR` 參數改為 `COLOR_BGR2RGB`。

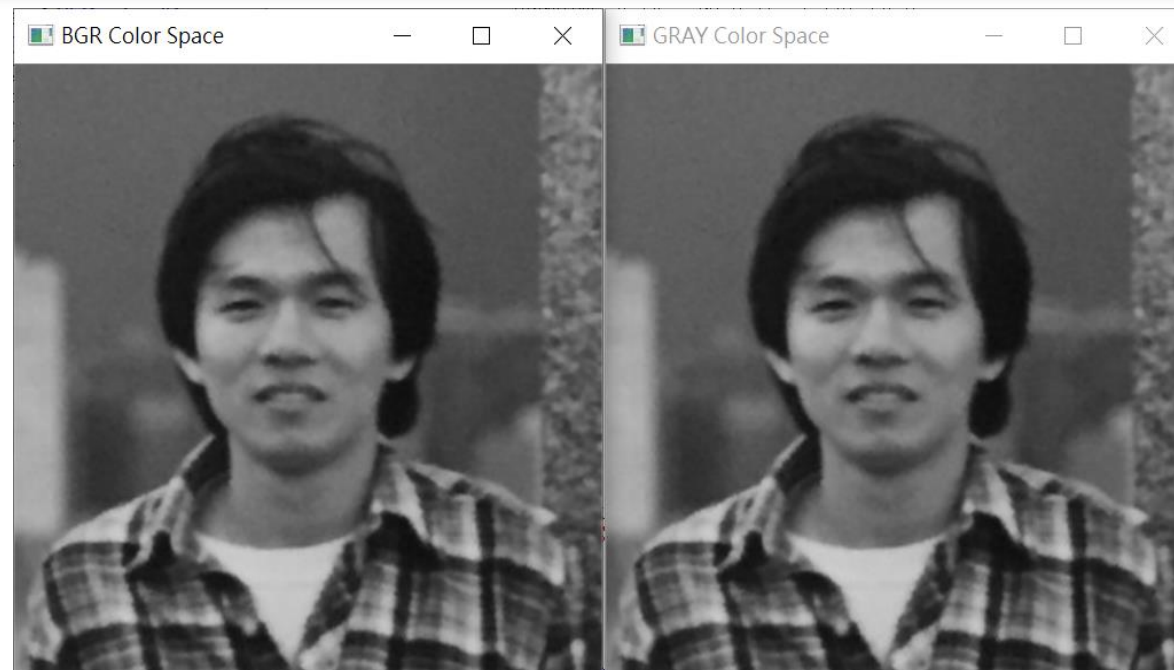
4-2：BGR色彩空間轉換至GRAY色彩空間

- 4-2-1：使用`cvtColor()`函數
- 程式實例`ch4_4.py`：讀取彩色影像`jk.jpg`，然後將此影像轉成灰階影像。



- 程式實例ch4_5.py：讀取彩色影像，將BGR色彩轉成GRAY色彩，然後顯示GRAY色彩的影像與此影像的通道值。然後將GRAY色彩轉為BGR色彩，然後顯示BGR色彩的影像與此影像的通道值。

```
===== RESTART: D:\OpenCV_Python\ch4\ch4_5.py =====  
Gray Color 通道值 = 128  
BGR Color 通道值 = [128 128 128]
```

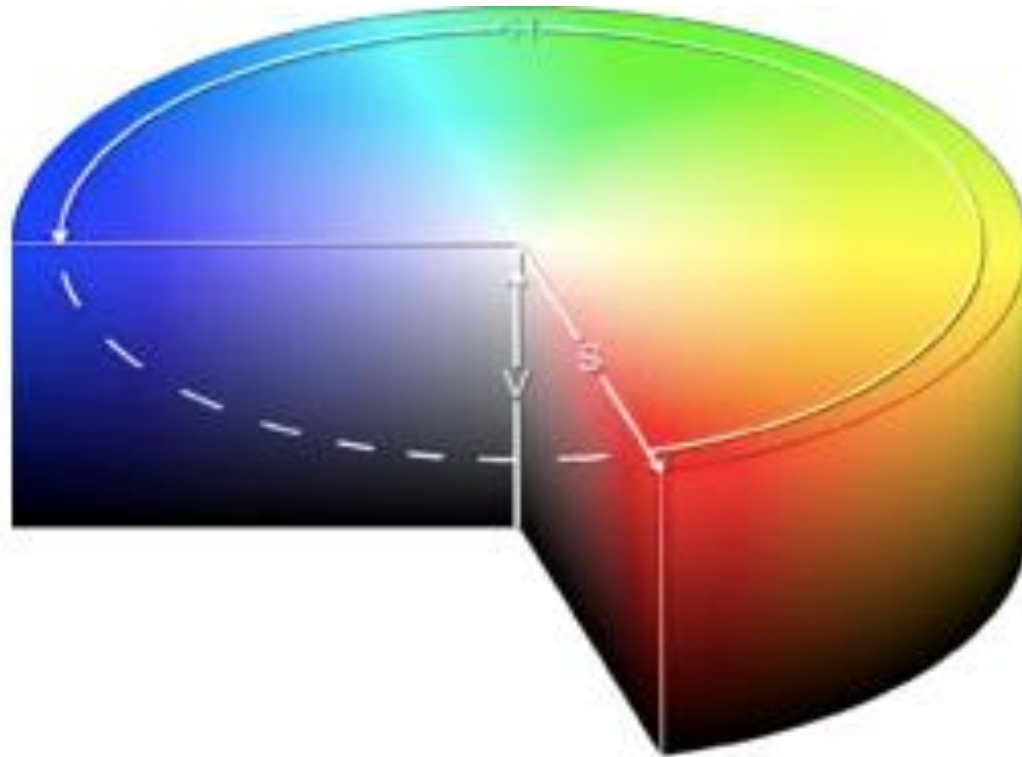


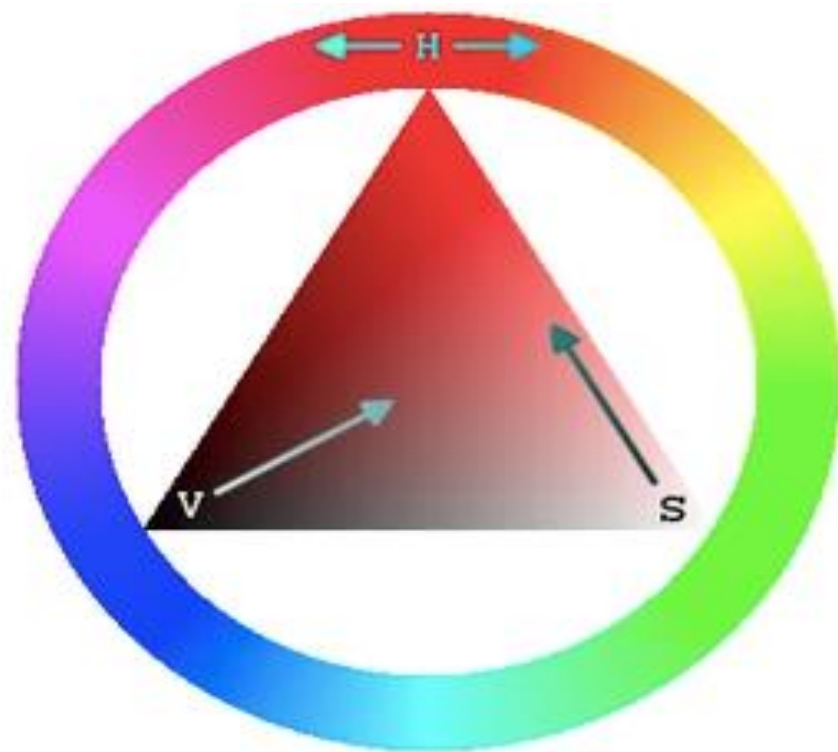
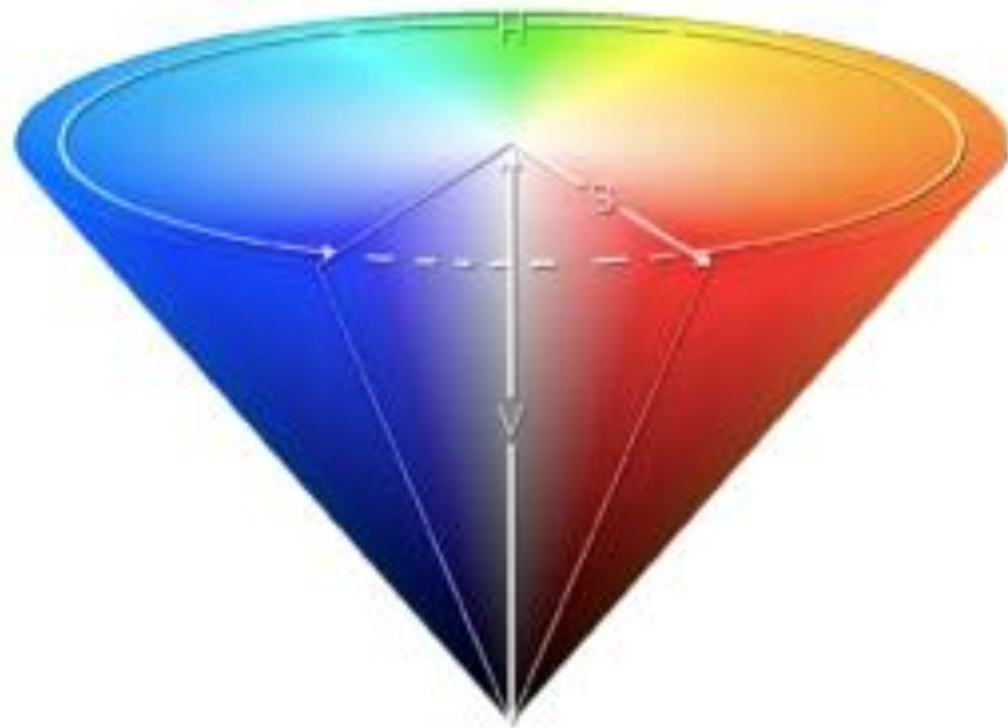
4-2-2 : OpenCV內部轉換公式

- $\text{Gray} = 0.2989 * R + 0.5870 * G + 0.1140 * B$

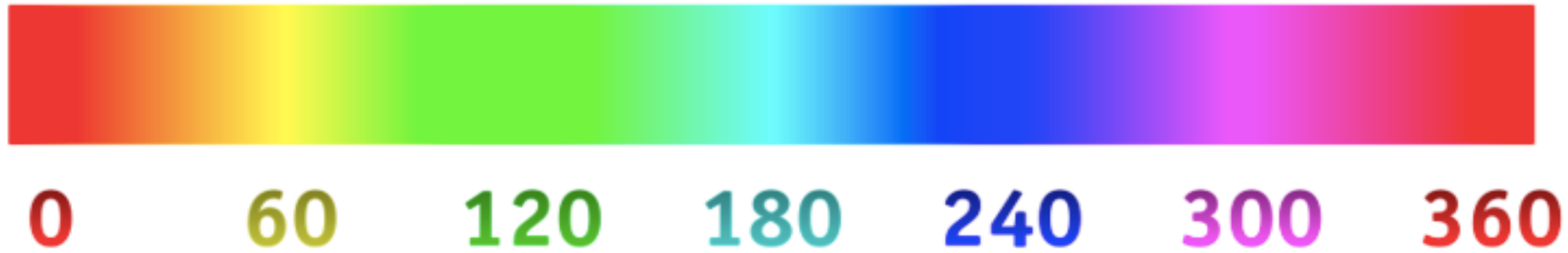
4-3：HSV色彩空間

- 4-3-1：認識HSV色彩空間





- 色調H(Hue)：是指色彩的基本屬性，也就是我們日常生活所謂的紅色、黃色、綠色、藍色、... 等。此值的範圍是0 ~ 360度之間，不過OpenCV依公式處理成0 ~ 180之間。



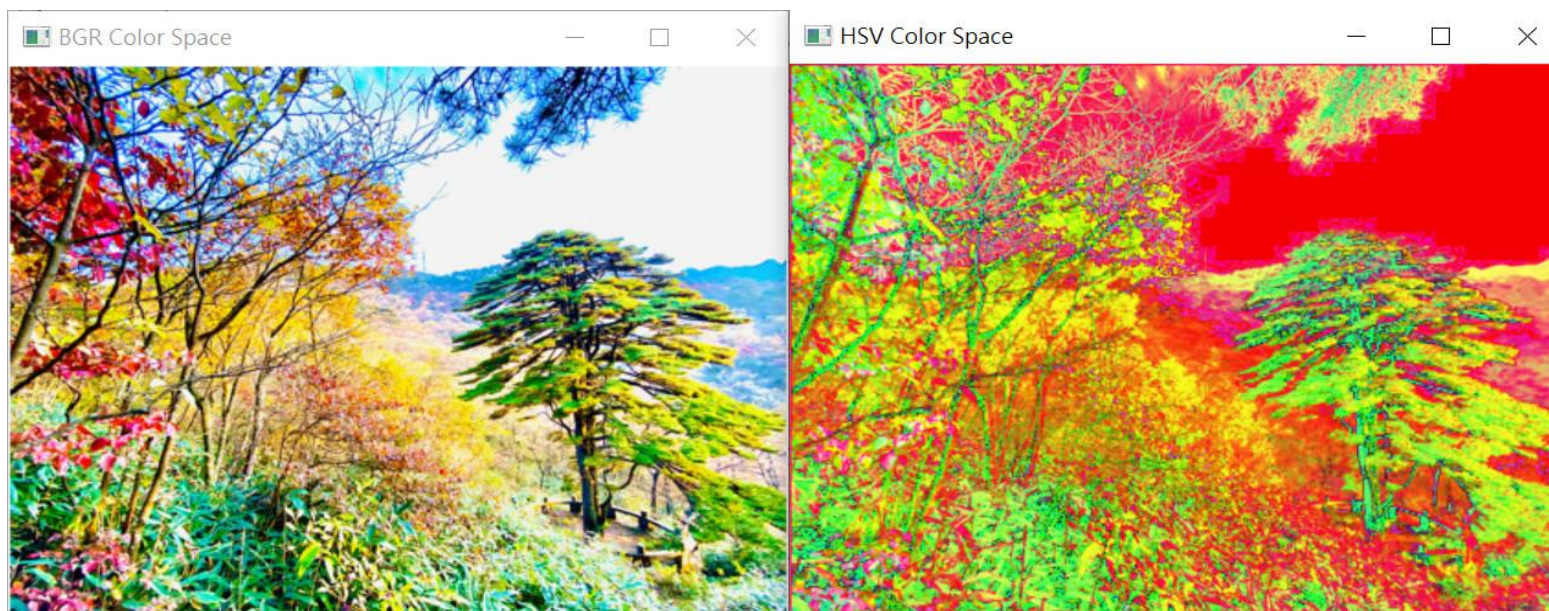
□ **飽和度S(Saturation)**：是指色彩的純度，數值越高彩純度越高，數值越低則逐漸變灰。此值範圍是0 ~ 100%，不過OpenCV也是依公式處理成0 ~ 255之間。下列左邊是原影像與右邊色彩飽和度是0%的比較。



□ 明度V(Value)：其實就是顏色的亮度，此值範圍是0 ~ 100%，不過OpenCV也是依公式處理成0 ~ 255之間，當明度是0時影像呈現黑色。

4-3-2：將影像由BGR色彩空間轉為HSV色彩空間

- `image = cv2.cvtColor(src, code)`
- 程式實例`ch4_6.py`：將影像由BGR色彩空間轉為HSV色彩空間，然後分別顯示原影像與HSV色彩空間影像。



4-3-3：將RGB色彩轉換成HSV色彩公式

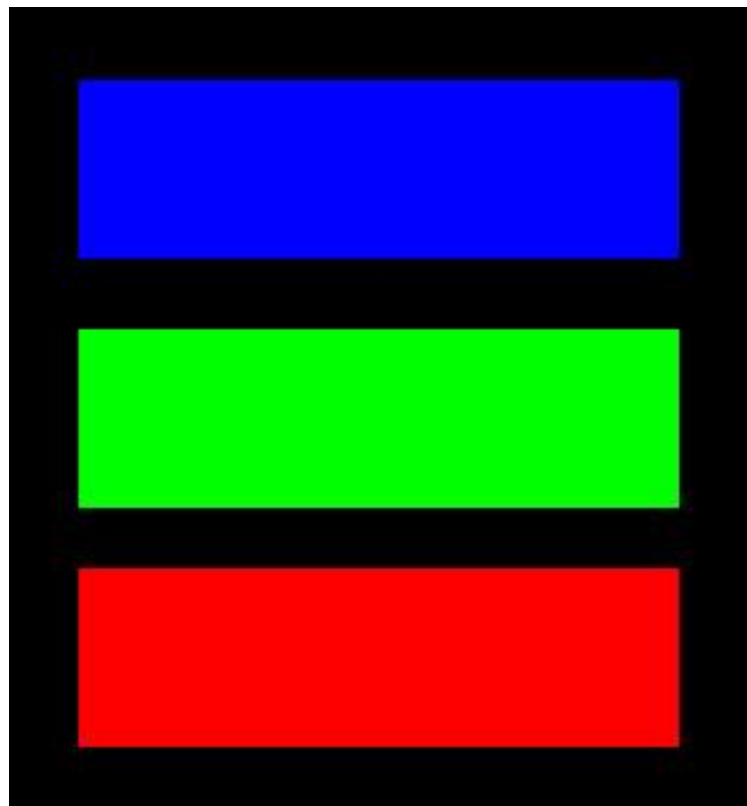
$$H = \begin{cases} \text{沒有定義} & \text{if } MAX = MIN \\ 60^\circ \times \frac{G-B}{MAX-MIN} + 0^\circ, & \text{if } MAX = R \\ & \text{and } G \geq B \\ 60^\circ \times \frac{G-B}{MAX-MIN} + 360^\circ, & \text{if } MAX = R \\ & \text{and } G < B \\ 60^\circ \times \frac{B-R}{MAX-MIN} + 120^\circ, & \text{if } MAX = G \\ 60^\circ \times \frac{R-G}{MAX-MIN} + 240^\circ, & \text{if } MAX = B \end{cases}$$

$$S = \begin{cases} 0, & \text{if } MAX = 0 \\ 1 - \frac{MIN}{MAX}, & \text{其他} \end{cases}$$

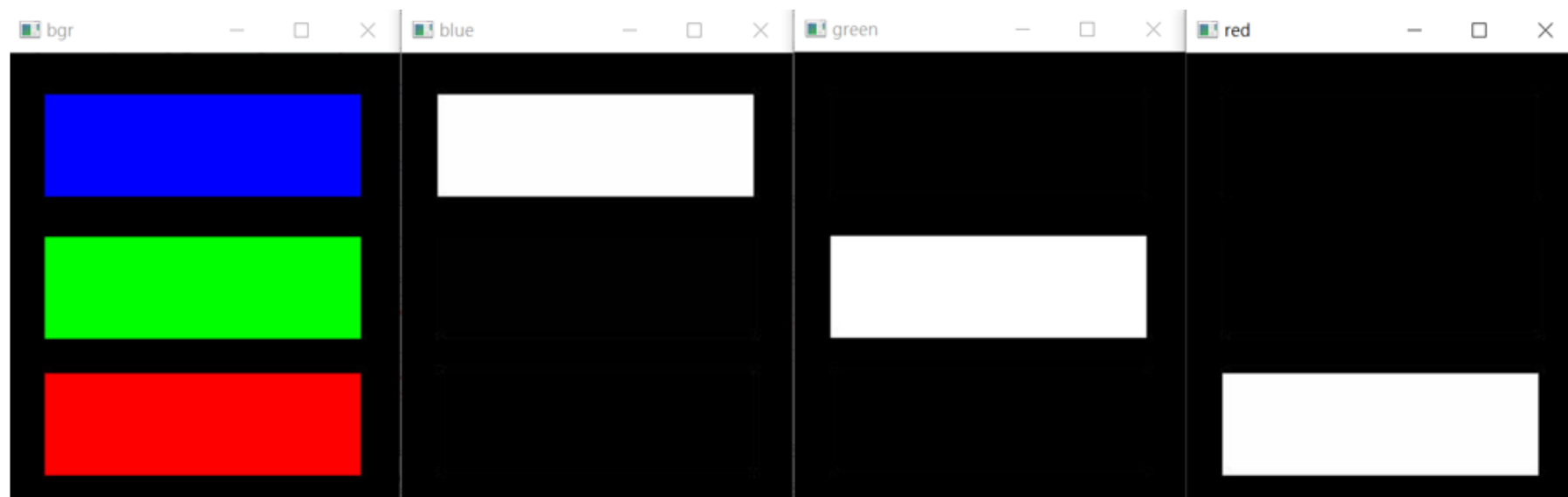
4-4：拆分色彩通道

- 4-4-1：拆分BGR影像的通道
- `blue, green, red = cv2.split(bgr_image)`
- 上述image是BGR影像物件，等號左邊blue, green, red內容如下：
 - blue：回傳B通道影像物件。
 - green：回傳G通道影像物件。
 - red：回傳R通道影像物件。

- 程式實例ch4_7.py：有一個影像colorbar.jpg，內容如下，請分別顯示此影像以及所拆分的通道影像。



```
===== RESTART: D:\OpenCV_Python\ch4\ch4_7.py =====  
B通道影像屬性 shape = (319, 279)  
列印B通道內容  
[[0 0 0 ... 0 0 0]  
 [0 0 0 ... 0 0 0]  
 [0 0 0 ... 0 0 0]  
 ...  
 [0 0 0 ... 0 0 0]  
 [0 0 0 ... 0 0 0]  
 [0 0 0 ... 0 0 0]]
```



colorbar.jpg

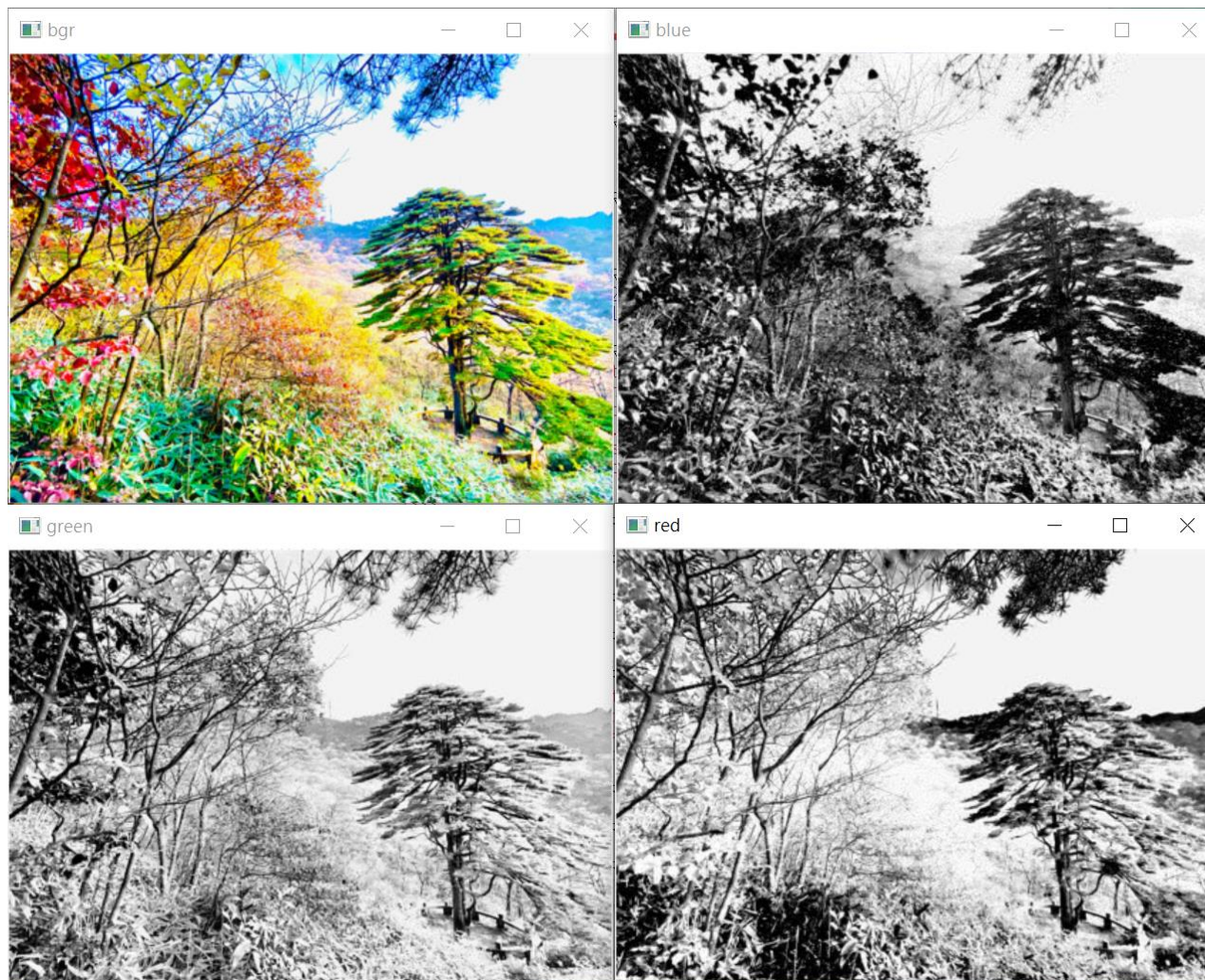
B通道影像

G通道影像

R通道影像

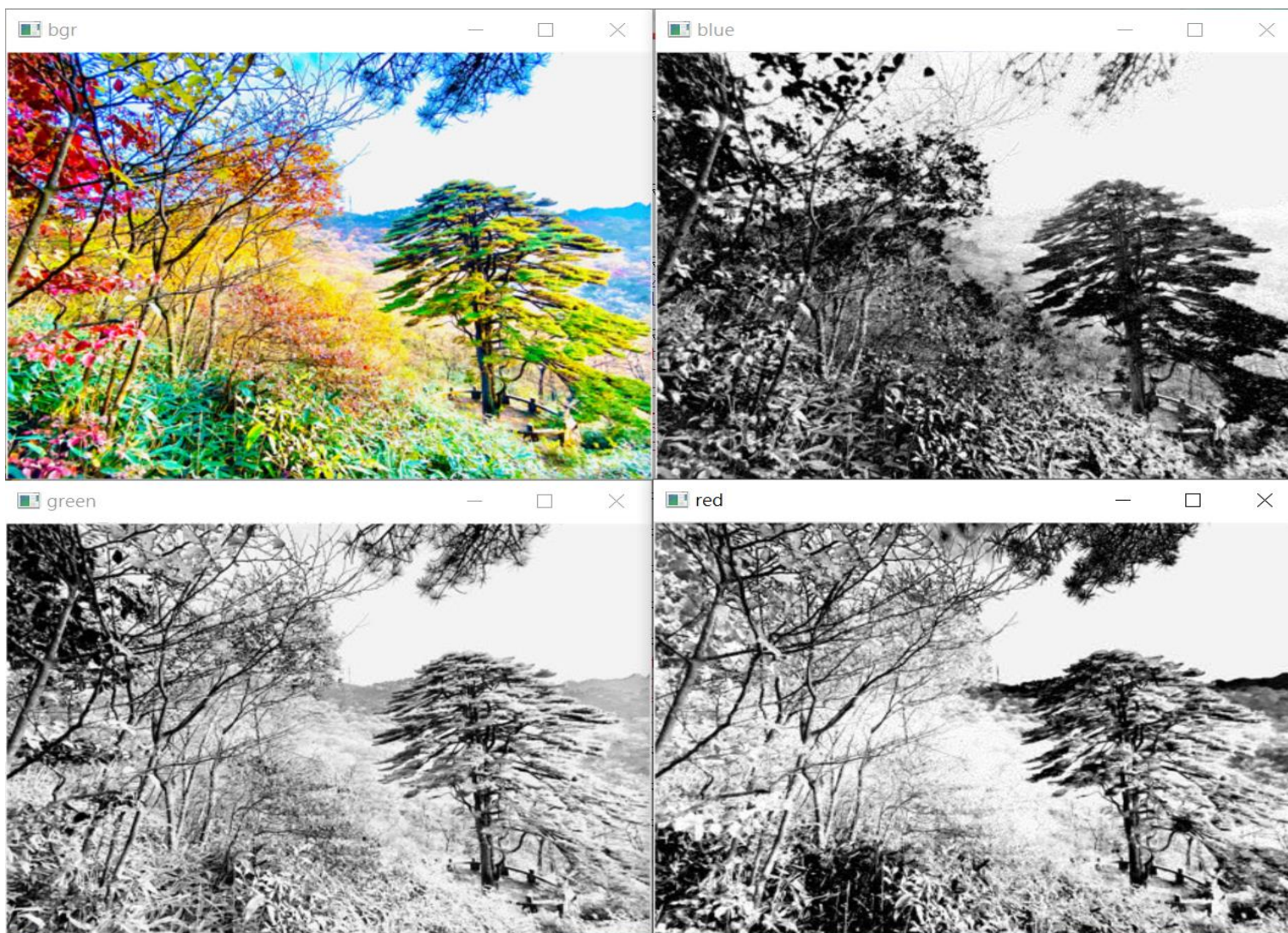
- 程式實例ch4_8.py：使用mountain.jpg取代colorbar.jpg，重新設計ch4_7.py，這一個實例同時驗證所拆分的影像是單通道，所得影像是以灰階顯示。

```
===== RESTART: D:\OpenCV_Python\ch4\ch4_8.py =====  
BGR 影像 : (314, 425, 3)  
B通道影像 : (314, 425)  
G通道影像 : (314, 425)  
R通道影像 : (314, 425)
```

- 程式實例ch4_8.py：使用mountain.jpg取代colorbar.jpg，重新設計ch4_7.py，這一個實例同時驗證所拆分的影像是單通道，所得影像是以灰階顯示。
- 執行結果：從下列可以看到BGR影像是3個通道，其他皆是1個通道。

```
===== RESTART: D:\OpenCV_Python\ch4\ch4_8.py =====  
BGR 影像 : (314, 425, 3)  
B通道影像 : (314, 425)  
G通道影像 : (314, 425)  
R通道影像 : (314, 425)
```

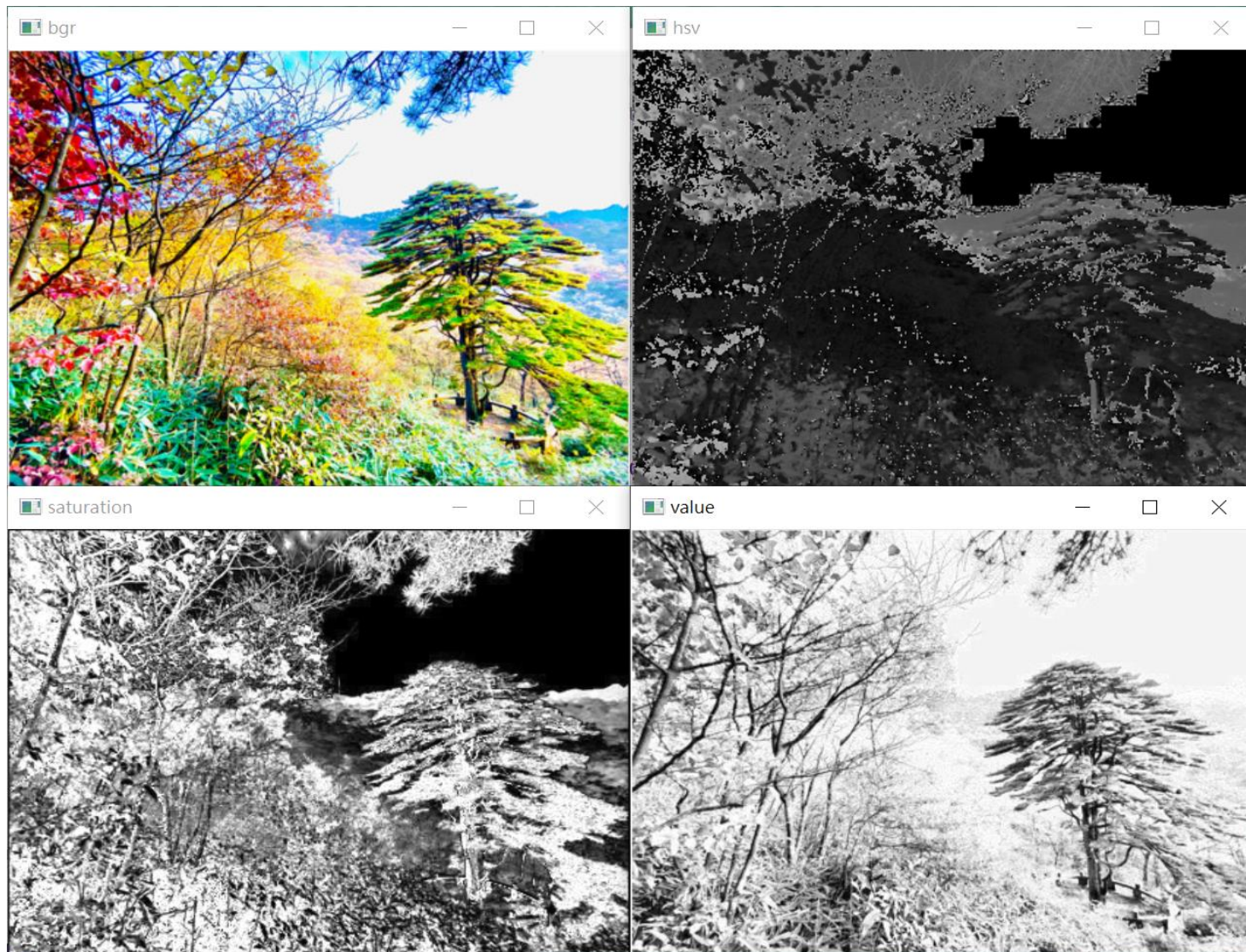



- 程式實例ch4_8_1.py：驗證BGR通道有不同的內容，因此影像呈現不同的灰階效果。
- 執行結果：省略繪製影像，只顯示B、G、R通道內容。


```
===== RESTART: D:\OpenCV_Python\ch4\ch4_8_1.py =====
BGR 影像 : (314, 425, 3)
B通道内容 :
[[250 250 252 ... 255 255 255]
 [248 34 27 ... 245 245 255]
 [246 27 12 ... 246 247 255]
 ...
 [254 57 23 ... 175 78 255]
 [248 61 55 ... 183 218 255]
 [255 78 65 ... 210 241 255]]
G通道内容 :
[[252 255 254 ... 255 255 255]
 [254 51 41 ... 245 245 255]
 [246 27 30 ... 246 247 255]
 ...
 [248 37 0 ... 187 56 255]
 [245 33 9 ... 209 222 255]
 [255 57 24 ... 235 244 255]]
R通道内容 :
[[252 251 254 ... 255 255 255]
 [255 124 129 ... 245 245 255]
 [255 133 137 ... 246 247 255]
 ...
 [255 6 7 ... 69 31 255]
 [255 22 1 ... 72 103 255]
 [251 25 0 ... 107 134 255]]
```

4-4-2：拆分HSV影像的通道

- 程式實例ch4_9.py：列印mountain.jpg，然後將此BGR影像物件轉成HSV影像物件，然後拆分HSV影像物件，最後列出拆分後的Hue通道影像物件、Saturation通道影像物件、Value通道影像物件。



4-5：合併色彩通道

- 4-5-1：合併B、G、R通道的影像

- `bgr_image = cv2.merge([blue, green, red])` # 合併通道的影像物件

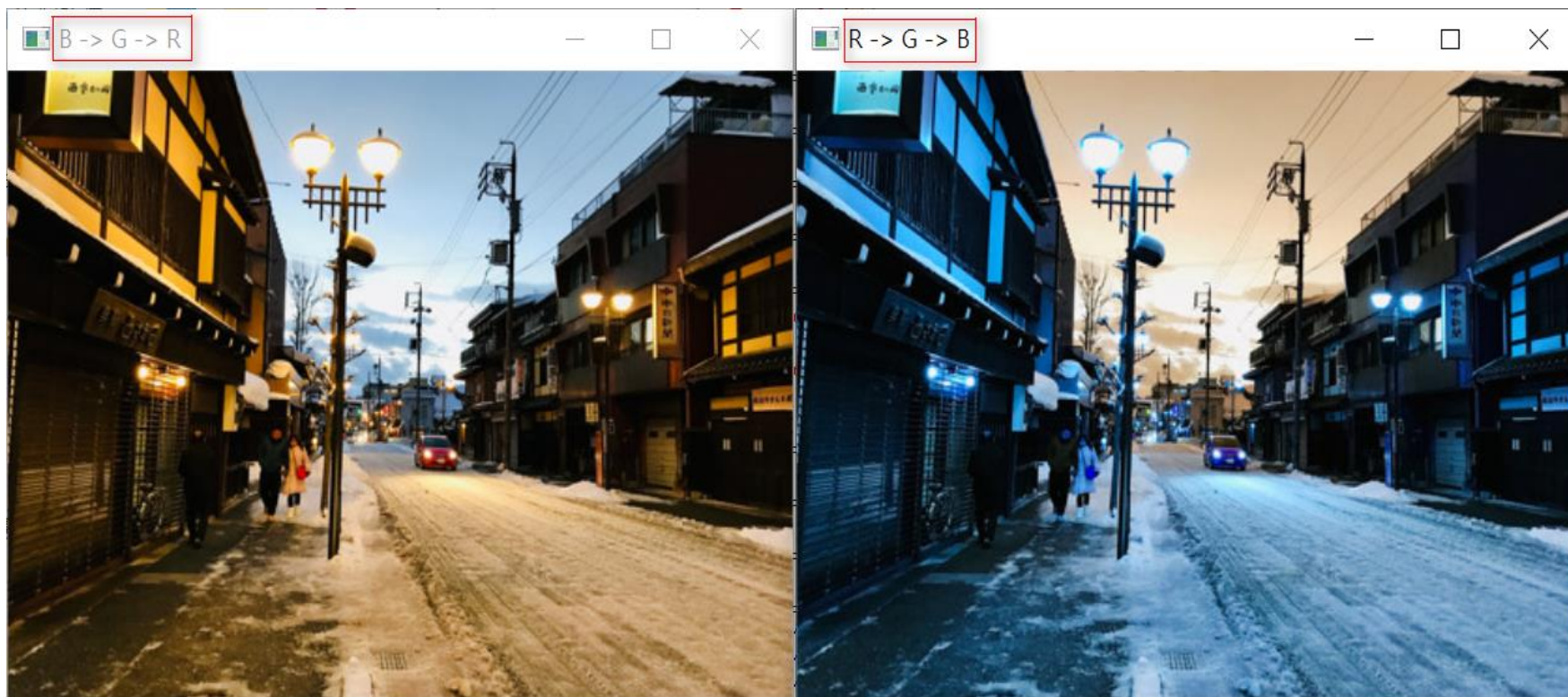
□ `blue`：B通道影像物件。

□ `green`：G通道影像物件。

□ `red`：R通道影像物件。

註：合併順序若是不同，所得的結果也會不同。

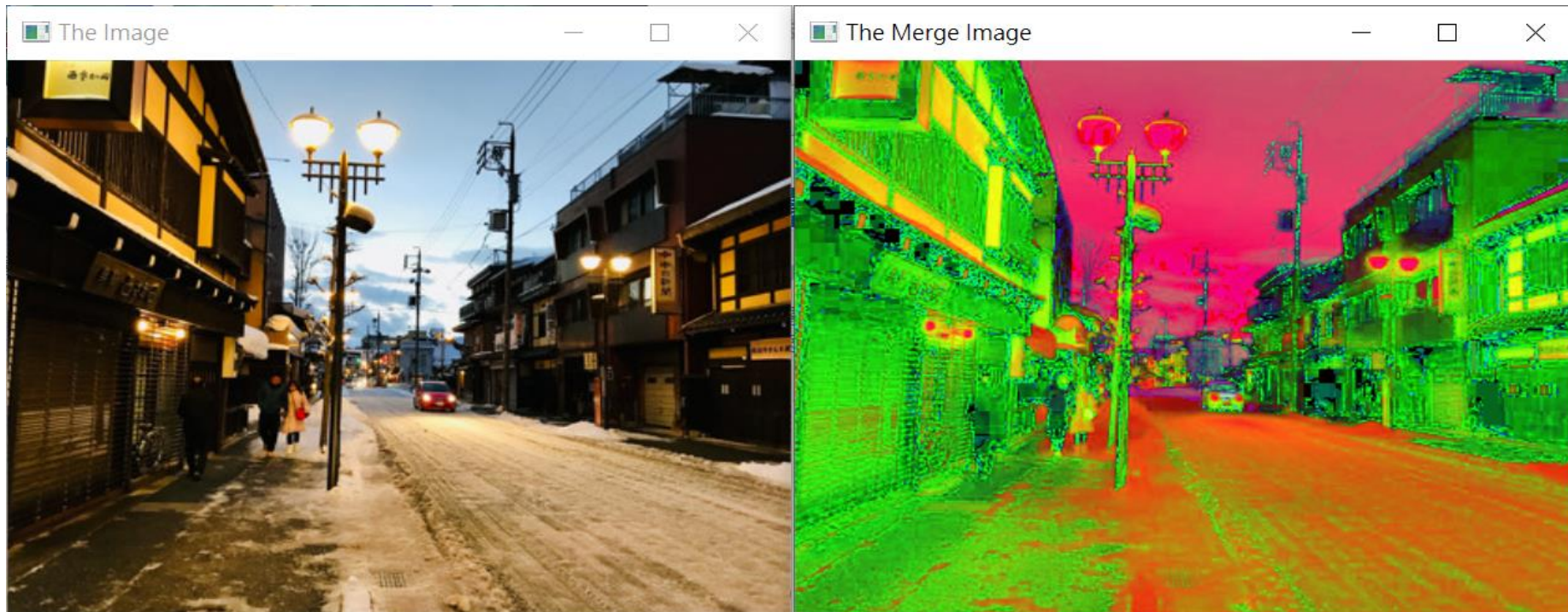
- 程式實例ch4_10.py：使用B -> G -> R順序合併，然後使用R -> G -> B順序合併，最後列出2個結果，視窗標題會顯示合併順序。



4-5-2：合併H、S、V通道的影像

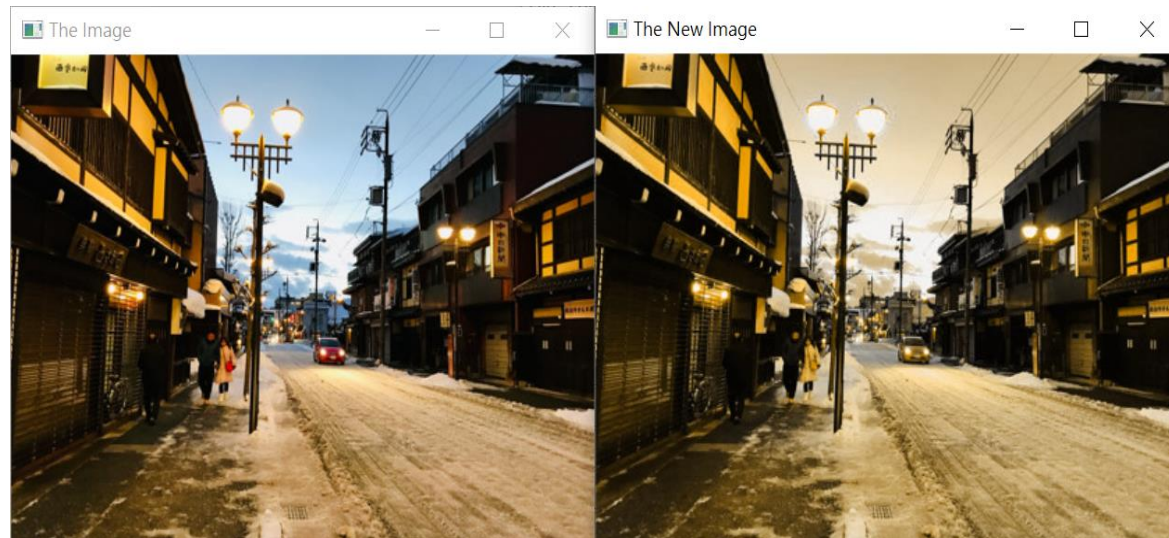
- `hsv_image = cv2.merge([hue, saturation, value])` # 合併通道的影像物件
- 上述hsv_image是HSV影像物件，merge()參數內容如下：
 - `hsv`：H通道影像物件。
 - `saturation`：S通道影像物件。
 - `value`：V通道影像物件。
- 註：合併順序若是不同，所得的結果也會不同。

- 程式實例ch4_11.py：顯示原影像，接著將BGR影像轉成HSV影像，然後將HSV影像的通道拆分，再使用H -> S -> V順序合併，最後列出合併結果。



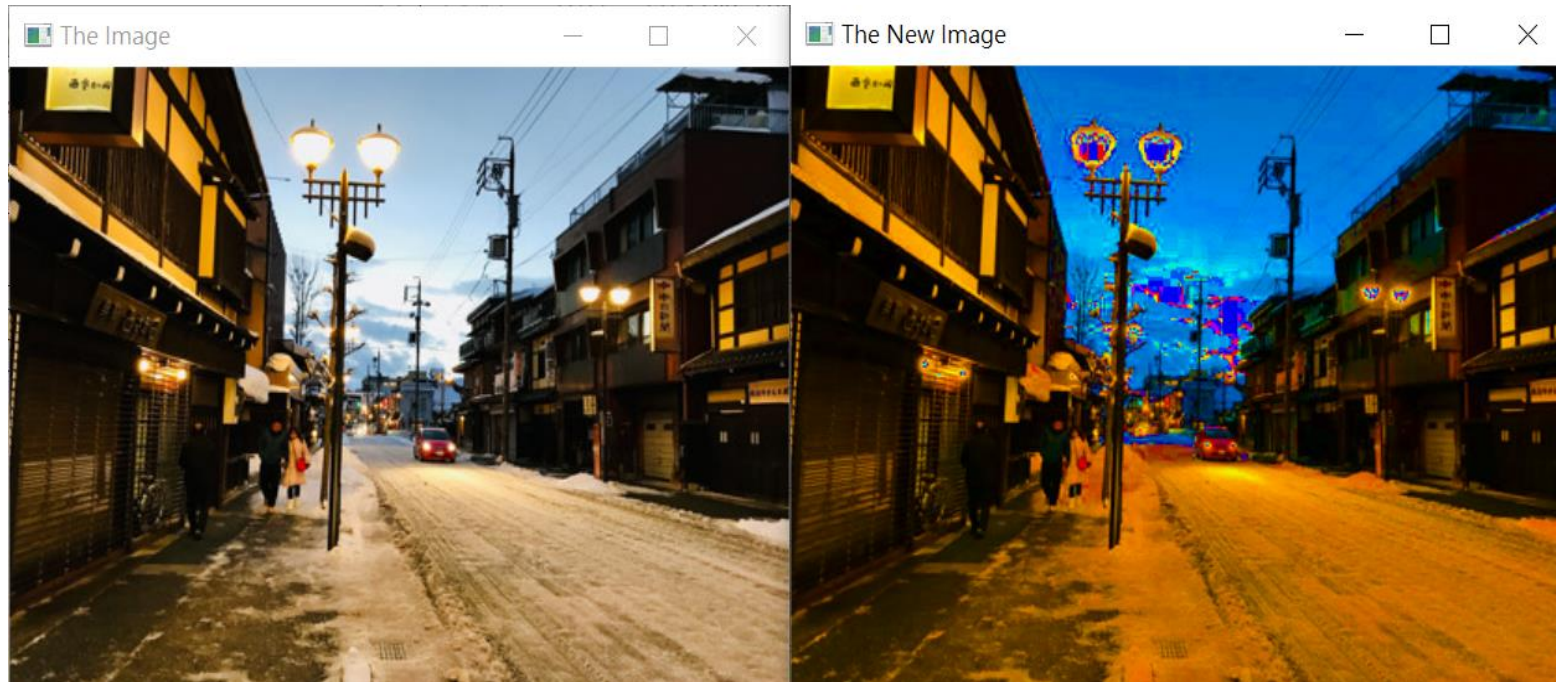
4-6：拆分與合併色彩通道的應用

- 4-6-1：色調Hue調整
- 程式實例ch4_12.py：將BGR影像轉成HSV，然後拆分，第8列修訂色調(hsv)為200，再將所拆分的hue、saturation、value通道合併，接著將HSV色彩轉回BGR色彩，然後顯示原影像，和修訂後的BGR色彩影像。



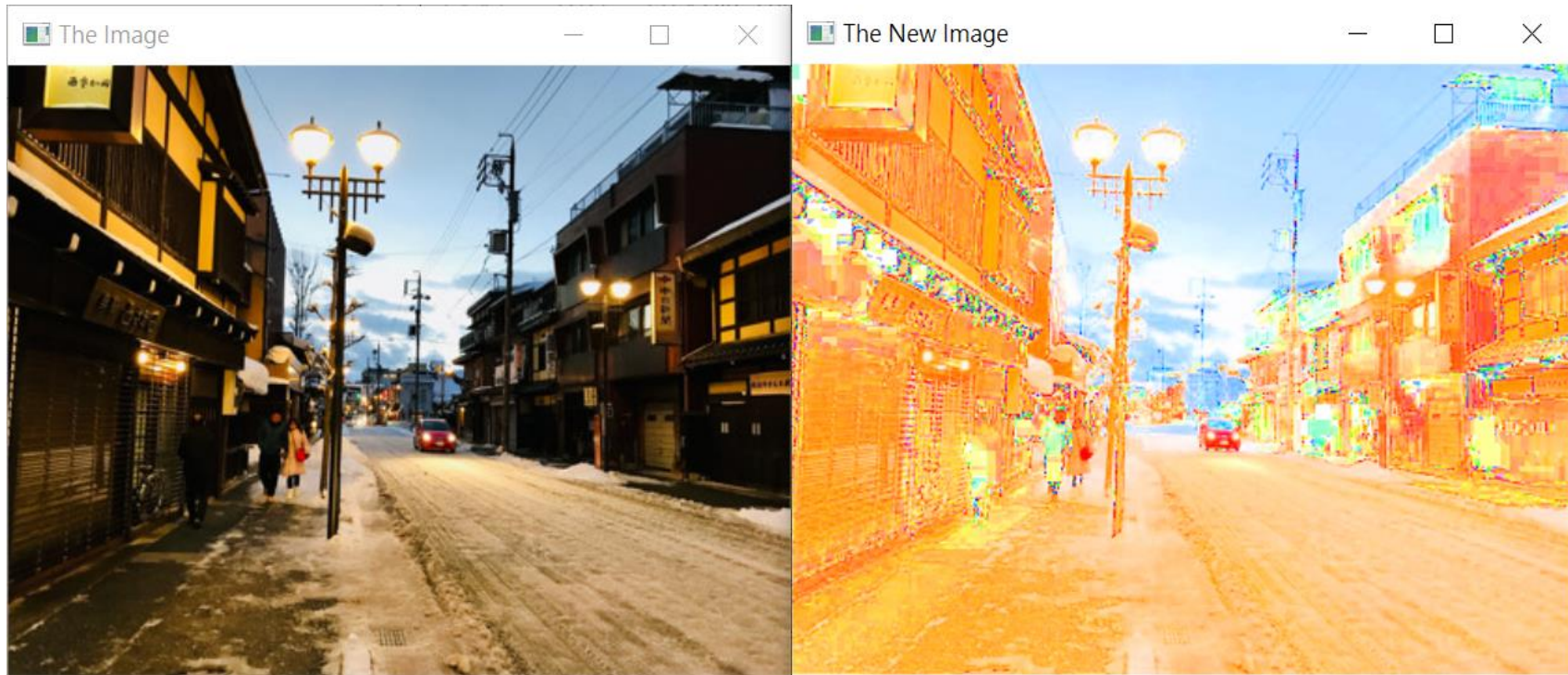
4-6-2：飽和度Saturation調整

- 程式實例ch4_13.py：重新設計ch4_12_1.py，將飽和度(saturation)設為255，然後列出結果做比對。



4-6-3：明度Value調整

- 程式實例ch4_14.py：重新設計ch4_12_1.py，將亮度(Value)設為255，然後列出結果做比對。



4-7 : alpha通道

- `bgra_image = cv2.cvtColor(image, cv2.COLOR_BGR2BGRA)`
- 程式實例`ch4_15.py`：顯示原影像，將BGR影像轉為BGRA影像同時顯示alpha通道值和影像，接著分別將BGRA影像轉為alpha=32和alpha=128，然後顯示以及儲存至a32_image和a128_image。

```
===== RESTART: D:/OpenCV_Python/ch4/ch4_15.py =====
列出轉成含A通道影像物件後的alpha值
[[255 255 255 ... 255 255 255]
 [255 255 255 ... 255 255 255]
 [255 255 255 ... 255 255 255]
 ...
 [255 255 255 ... 255 255 255]
 [255 255 255 ... 255 255 255]
 [255 255 255 ... 255 255 255]]
```


The Image



The a32 Image



The a128 Image



a32.png



a128.png

