色彩空间与通道

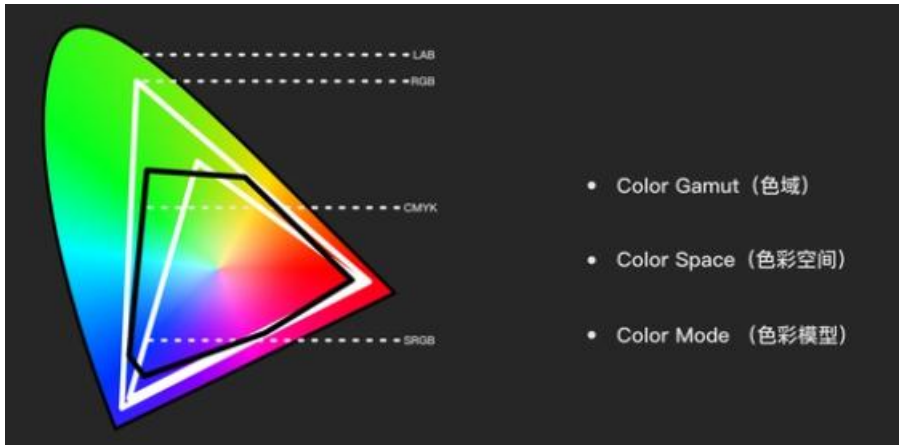
电子信息工程系

袁羽

目录

CONTENTS

一 色彩空间



- 色彩是人类的眼睛对于不同频率的光线的不同感受。
- 为了表示这些不同频率的光线的色彩，人类建立了多种色彩模型，把这些色彩模型称作色彩空间。
- 色彩空间分类：RGB、BGR、GRAY、HSV、CMYK、Lab等。

1 GRAY色彩空间

- GRAY色彩空间通常指的是灰度图像，灰度图像是一种每个像素都是从黑到白，被处理为256个灰度级别的单色图像。这256个灰度级别分别用区间[0，255]中的数值表示。其中，“0”表示纯黑色，“255”表示纯白色，0～255的数值表示。
- 不同亮度（即色彩的深浅程度）的深灰色或者浅灰色。因此，一幅灰度图像也能够展现丰富的细节信息。

2 从BGR色彩空间到GRAY色彩空间的转换

OpenCV可以实现从BGR色彩空间到其他色彩空间的转换。语法格式如下：

```
dst = cv2.cvtColor(src, code)
```

参数说明：

dst：转换后的图像。

src：转换前的初始图像。

code：色彩空间转换码。

说明：

当图像从BGR色彩空间转换到GRAY色彩空间时，常用的色彩空间转换码是cv2.COLOR_BGR2GRAY。

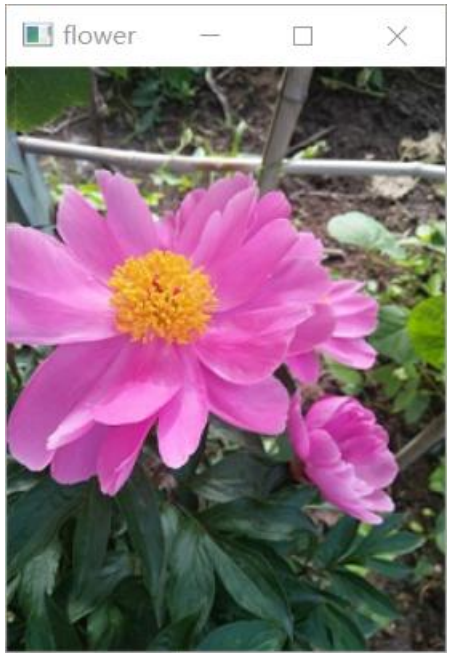
```
dst = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
```

例：编写一个程序，将图flower从BGR色彩空间转换到GRAY色彩空间

```
import cv2
image = cv2.imread("flower.jpg")
cv2.imshow("flower", image) # 显示图像
# 将图flower从BGR色彩空间转换到GRAY色彩空间
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imshow("GRAY", gray_image) # 显示灰度图像
cv2.waitKey()
cv2.destroyAllWindows()
```

例：编写一个程序，将图flower从BGR色彩空间转换到GRAY色彩空间

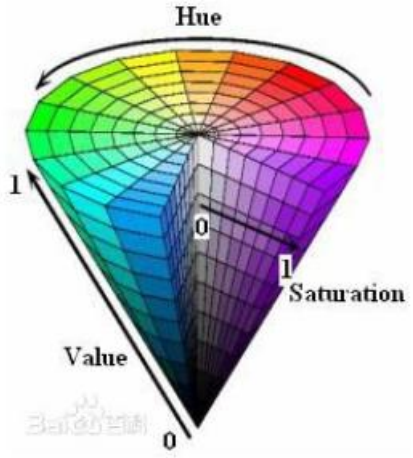
运行结果：



3 HSV色彩空间

HSV 色彩空间是一种面向视觉感知的颜色模型，从心理学和视觉的角度出发，指出人眼的色彩知觉主要包含三要素：

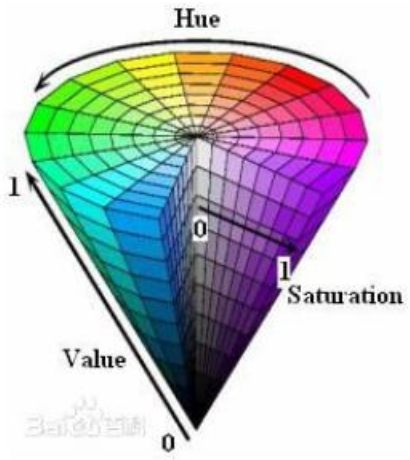
色调（Hue，也称为色相）、饱和度（Saturation）、亮度（Value）。



3 HSV色彩空间

色调H：

在HSV色彩空间中，色调H的取值范围是[0,360]。8位图像内每个像素点所能表示的灰度级有256个，所以在8位图像内表示HSV图像时，要把色调的角度映射到[0,255]范围内，所以利用OpenCV转换之后得到的H的范围为 0-180。

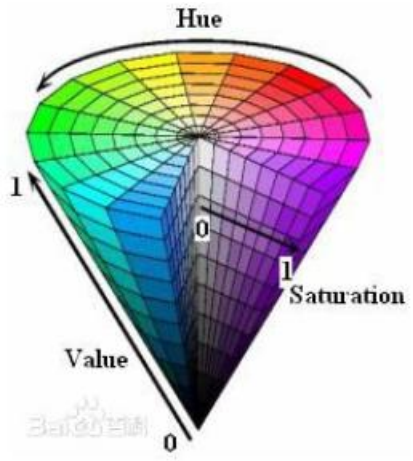


色调值	颜色
0	红色
30	黄色
60	绿色
90	青色
120	蓝色
150	品红色

3 HSV色彩空间

饱和度S：

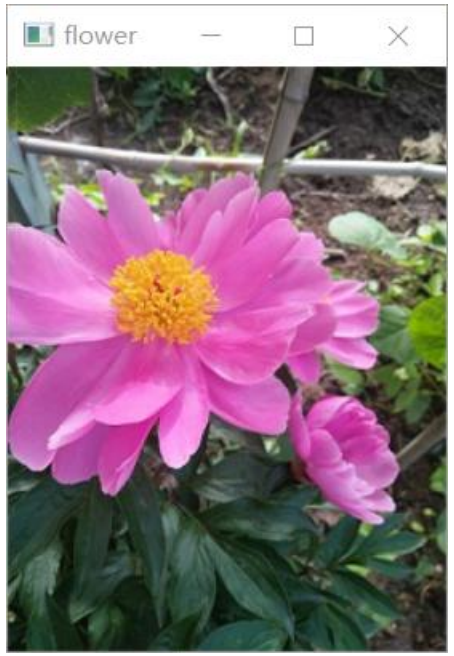
饱和度（S）是指色彩的深浅。在OpenCV中，饱和度在区间[0，255]内取值。



3 HSV色彩空间

饱和度S：

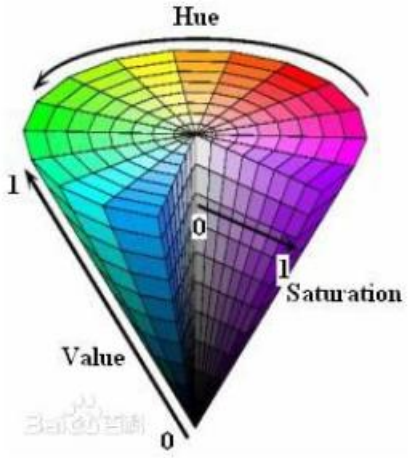
灰度颜色所包含R、G、B的成分是相当的，相当于一种极不饱和的颜色。所以，灰度颜色的饱和度是0。作为灰度图像显示时，较亮区域对应的颜色具有较高的饱和度。左是用手机拍摄的原图像，右图是把图flower的饱和度调为0时的效果。



3 HSV色彩空间

亮度V：

亮度值在OpenCV内也将值映射到[0,255]范围内。亮度值越大，图像越亮；亮度值越低，图像越暗。当亮度值为0时，图像为纯黑色



4 从BGR色彩空间转换到HSV色彩空间

OpenCV提供的cvtColor()方法不仅能将图像从BGR色彩空间转换到GRAY色彩空间，还能将图像从BGR色彩空间转换到HSV色彩空间。

语法格式如下：

```
dst = cv2.cvtColor(src,cv2.COLOR_BGR2HSV)
```

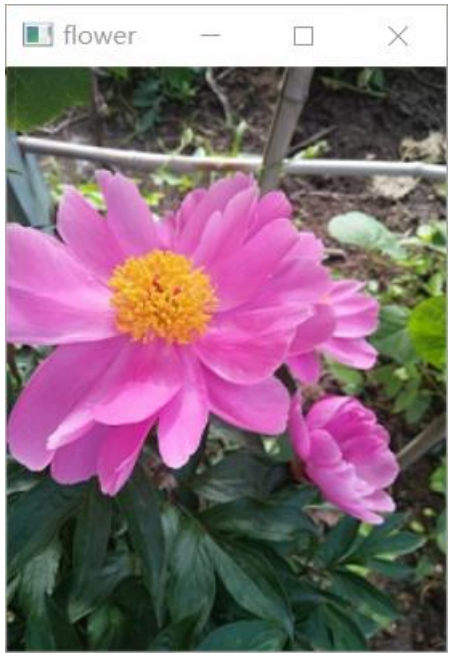
当图像在BGR色彩空间和HSV色彩空间之间转换时，常用的色彩空间转换码是cv2.COLOR_BGR2HSV和cv2.COLOR_HSV2BGR。

例：编写一个程序，将图flower从BGR色彩空间转换到HSV色彩空间

```
import cv2
image = cv2.imread("flower.jpg")
cv2.imshow("flower", image)                    # 显示图flower
# 将图flower从BGR色彩空间转换到HSV色彩空间
hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
cv2.imshow("HSV", hsv_image)                   # 用HSV色彩空间显示的图像
cv2.waitKey()
cv2.destroyAllWindows()
```

例：编写一个程序，将图flower从BGR色彩空间转换到HSV色彩空间

运行结果：



二 通道

在图像处理中，每一种基本颜色表示的是一个颜色通道（band）。对于常用的彩色图像，通常用三基色R（红色）、G（绿色）和B（蓝色）表示，即有3个颜色通道，R、G、B按照灰度等级的不同比例组合（调配），则会产生不同的色彩。

常见的图像格式包括:

- 单通道 （ Grayscale)
- 3 通道(Red, Green, and Blue)
- 4通道 (Red, Green, Blue, and “Alpha”, a.k.a. Opacity)

在BGR色彩空间中，图像的通道由B通道、G通道和R通道构成。

1 拆分通道

split()方法可以拆分一幅BGR图像中的通道，语法如下：

```
b, g, r = cv2.split(bgr_image)
```

参数说明：

b：B通道图像。

g：G通道图像。

r：R通道图像。

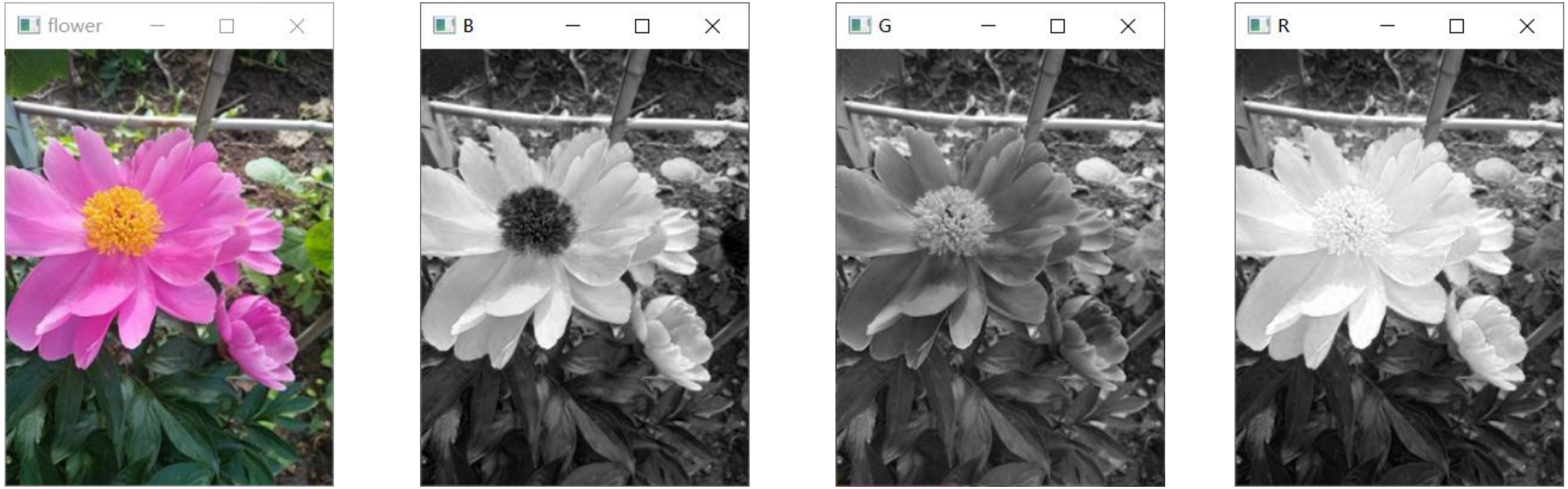
bgr_image：一幅BGR图像。

例 编写一个程序，先拆分一幅BGR图像中的通道，再显示拆分后的通道图像

```
import cv2
bgr_image = cv2.imread("flower.jpg")
cv2.imshow("flower", bgr_image) # 显示图flower
b, g, r = cv2.split(bgr_image) # 拆分图flower中的通道
cv2.imshow("B", b) # 显示B通道图像
cv2.imshow("G", g) # 显示G通道图像
cv2.imshow("R", r) # 显示R通道图像
cv2.waitKey()
cv2.destroyAllWindows()
```

例 编写一个程序，先拆分一幅BGR图像中的通道，再显示拆分后的通道图像

运行结果：



1 拆分通道

split()方法可以拆分一幅HSV图像中的通道，语法如下：

```
h, s, v = cv2.split(hsv_image)
```

参数说明：

h：H通道图像。

s：S通道图像。

v：V通道图像。

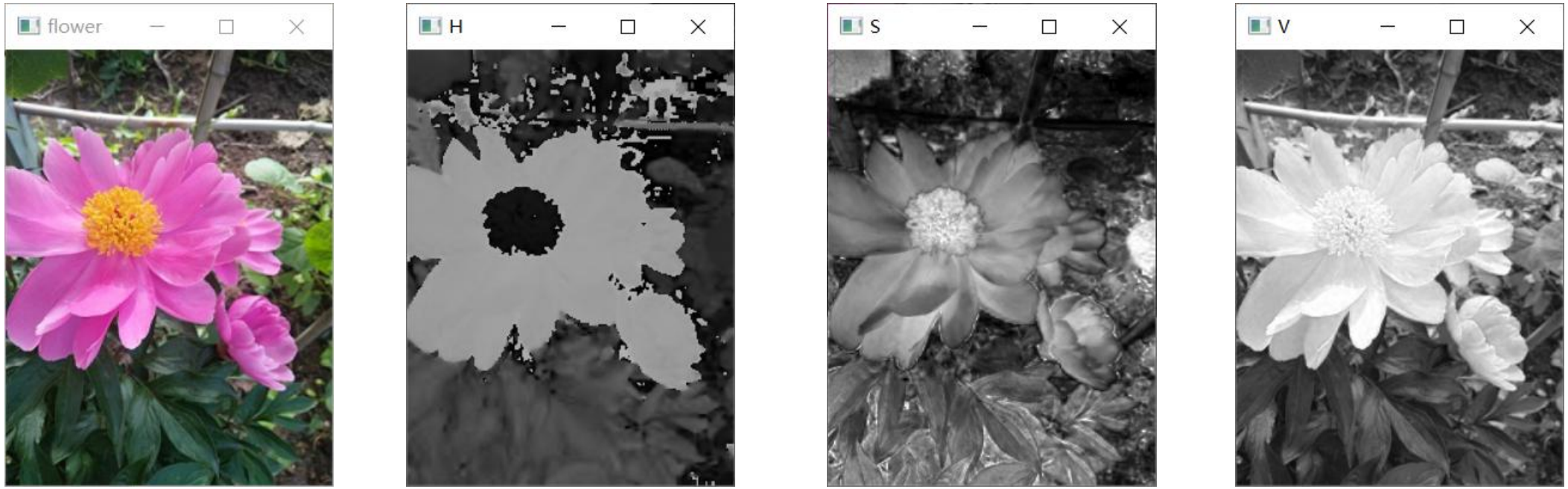
hsv_image：一幅HSV图像。

例 编写一个程序，先拆分一幅HSV图像中的通道，再显示拆分后的通道图像

```
import cv2
bgr_image = cv2.imread("flower.jpg")
cv2.imshow("flower", bgr_image) # 显示图flower
# 图flower从BGR色彩空间转换到HSV色彩空间
hsv_image = cv2.cvtColor(bgr_image, cv2.COLOR_BGR2HSV)
h, s, v = cv2.split(hsv_image) # 拆分HSV图像中的通道
cv2.imshow("H", h) # 显示HSV图像中的H通道图像
cv2.imshow("S", s) # 显示HSV图像中的S通道图像
cv2.imshow("V", v) # 显示HSV图像中的V通道图像
cv2.waitKey()
cv2.destroyAllWindows()
```

例 编写一个程序，先拆分一幅HSV图像中的通道，再显示拆分后的通道图像

运行结果：



2 合并通道

merge()方法可以按B→G→R的顺序合并通道，语法如下：

```
bgr = cv2.merge([b, g, r])
```

参数说明：

bgr：按B→G→R的顺序合并通道后得到的图像。

b：B通道图像。

g：G通道图像。

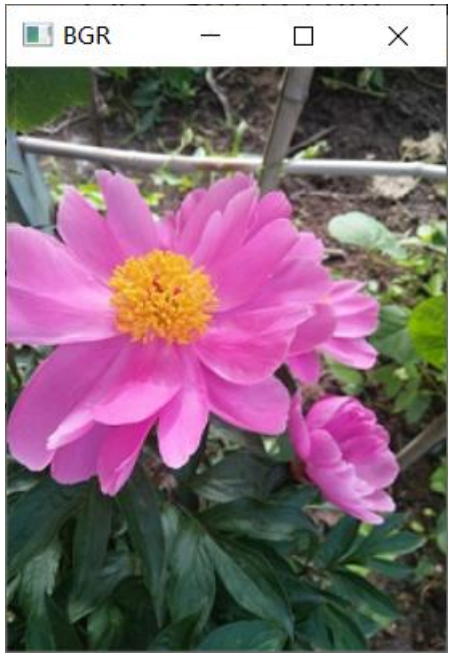
r：R通道图像。

例 编写一个程序，按B→G→R的顺序对图像执行先拆分通道，再分别按照bgr和rgb的顺序合并通道

```
import cv2
bgr_image = cv2.imread("flower.jpg")
b, g, r = cv2.split(bgr_image) # 拆分图flower中的通道
bgr = cv2.merge([b, g, r]) # 按B→G→R的顺序合并通道
cv2.imshow("BGR", bgr)
rgb = cv2.merge([r, g, b]) # 按R→G→B的顺序合并通道
cv2.imshow("RGB", rgb)
cv2.waitKey()
cv2.destroyAllWindows()
```

例 编写一个程序，按B→G→R的顺序对图像执行先拆分通道，再分别按照bgr和rgb的顺序合并通道

运行结果：



以上两幅图虽然是同一幅图，但是显示效果却不同，使用OpenCV按b g r的顺序合并通道可以得到原图，按照r g b的顺序合并通道无法得到原图。

2 合并通道

merge()方法合并H通道图像、S通道图像和V通道图像时，merge()方法的语法如下：

```
hsv = cv2.merge([h, s, v])
```

参数说明：

hsv：合并H通道图像、S通道图像和V通道图像后得到的图像。

h：H通道图像。

s：S通道图像。

v：V通道图像。

例 编写一个程序，首先将图像从BGR色彩空间转换到HSV色彩空间，然后拆分得到的HSV图像中的通道，接着合并拆分后的通道图像，最后将合并通道后的图像从HSV色彩空间转换到BGR色彩空间。

```
import cv2
bgr_image = cv2.imread("flower.jpg")
# 图flower从RGB色彩空间转换到HSV色彩空间
hsv_image = cv2.cvtColor(bgr_image, cv2.COLOR_BGR2HSV)
h, s, v = cv2.split(hsv_image) # 拆分HSV图像中的通道
hsv = cv2.merge([h, s, v]) # 合并拆分后的通道图像
cv2.imshow("HSV", hsv) # 显示合并通道的HSV图像
cv2.waitKey()
cv2.destroyAllWindows()
```

例 编写一个程序，首先将图像从BGR色彩空间转换到HSV色彩空间，然后拆分得到的HSV图像中的通道，接着合并拆分后的通道图像，最后将合并通道后的图像从HSV色彩空间转换到BGR色彩空间。

运行结果：

3 综合运用拆分通道和合并通道

在HSV色彩空间内，如果保持其中两个通道的值不变，调整第3个通道的值，会得到相应的艺术效果。

练习 编写一个程序，实现以下功能。

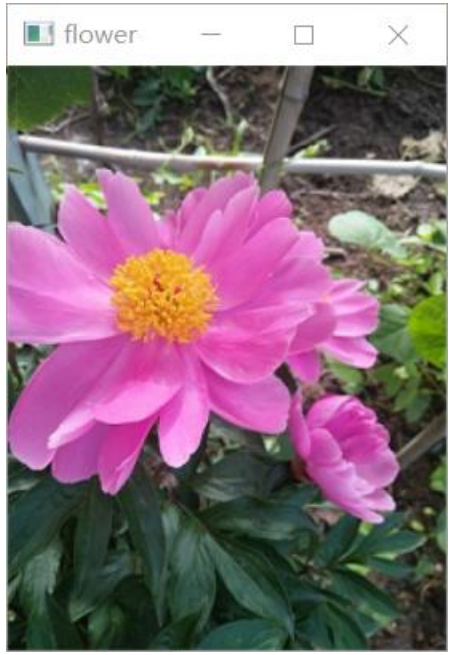
读取图像，展示图像；
将图像从BGR色彩空间转换到HSV色彩空间；
拆分HSV图像中的通道；
让饱和度S通道和亮度V通道的值保持不变，把色调H通道的值调整为30；
合并拆分后的通道图像；
把这个图像从HSV色彩空间转换到BGR色彩空间；
显示得到的BGR图像；

案例代码

```
import cv2

bgr_image = cv2.imread("flower.jpg")
cv2.imshow("flower", bgr_image)
# 图flower从BGR色彩空间转换到HSV色彩空间
hsv_image = cv2.cvtColor(bgr_image,
cv2.COLOR_BGR2HSV)
h, s, v = cv2.split(hsv_image) # 拆分HSV图像中的通道
h[:, :] = 30 # 将H通道的值调整为30
hsv = cv2.merge([h, s, v]) # 合并拆分后的通道图像
# 合并通道后的图像从HSV色彩空间转换到BGR色彩
空间
new_Image = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
cv2.imshow("NEW",new_Image)
cv2.waitKey()
cv2.destroyAllWindows()
```

运行结果：



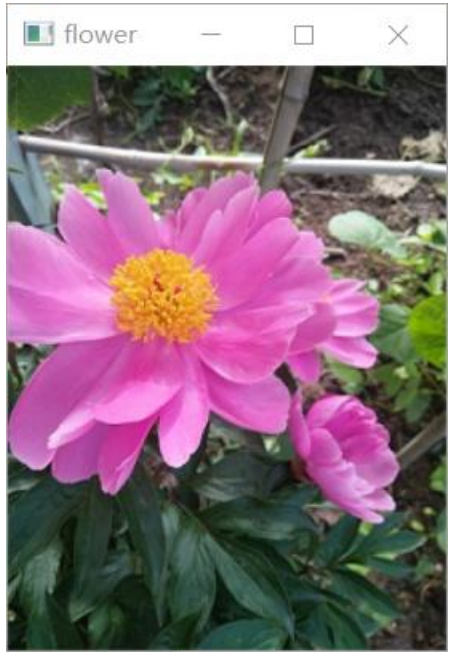
练习 编写一个程序，实现以下功能。

读取图像，展示图像；
将图像从BGR色彩空间转换到HSV色彩空间；
拆分HSV图像中的通道；
让色调H通道和饱和度S通道的值保持不变，把亮度V通道的值调整为255；
合并拆分后的通道图像；
把这个图像从HSV色彩空间转换到BGR色彩空间；
显示得到的BGR图像；

案例代码

```
import cv2
bgr_image = cv2.imread("flower.jpg")
cv2.imshow("flower", bgr_image)
# 图flower从BGR色彩空间转换到HSV色彩空间
hsv_image = cv2.cvtColor(bgr_image,
cv2.COLOR_BGR2HSV)
h, s, v = cv2.split(hsv_image) # 拆分HSV图像中的通道
v[:, :] = 255 # 将V通道的值调整为255
hsv = cv2.merge([h, s, v]) # 合并拆分后的通道图像
# 合并通道后的图像从HSV色彩空间转换到BGR色彩
空间
new_Image = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
cv2.imshow("NEW",new_Image)
cv2.waitKey()
cv2.destroyAllWindows()
```

运行结果：



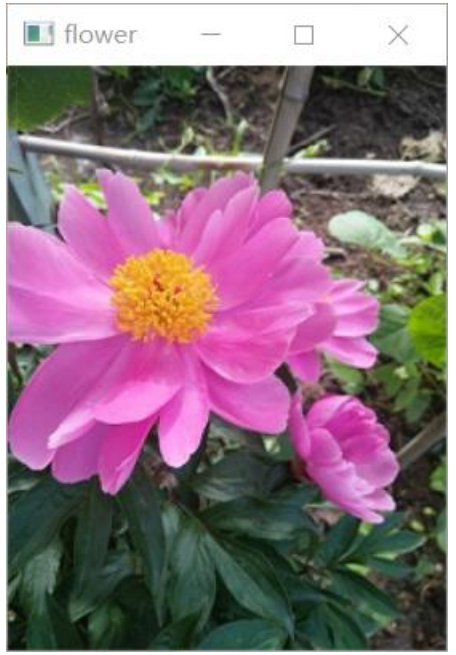
练习 编写一个程序，实现以下功能。

读取图像，展示图像；
将图像从BGR色彩空间转换到HSV色彩空间；
拆分HSV图像中的通道；
让色调H通道和亮度V通道的值保持不变，把饱和度S通道的值调整为255；
合并拆分后的通道图像；
把这个图像从HSV色彩空间转换到BGR色彩空间；
显示得到的BGR图像；

案例代码

```
import cv2
bgr_image = cv2.imread("flower.jpg")
cv2.imshow("flower", bgr_image)
# 图flower从BGR色彩空间转换到HSV色彩空间
hsv_image = cv2.cvtColor(bgr_image, cv2.COLOR_BGR2HSV)
h, s, v = cv2.split(hsv_image) # 拆分HSV图像中的通道
 s[:, :] = 255 # 将S通道的值调整为255
hsv = cv2.merge([h, s, v]) # 合并拆分后的通道图像
# 合并通道后的图像从HSV色彩空间转换到BGR色彩空间
new_Image = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
cv2.imshow("NEW",new_Image)
cv2.waitKey()
cv2.destroyAllWindows()
```

运行结果：



4 alpha通道

BGR色彩空间包含了3个通道，即B通道、G通道和R通道。OpenCV在BGR色彩空间的基础上，又增加了一个用于设置图像透明度的A通道，即alpha通道。这样，形成一个由B通道、G通道、R通道和A通道4个通道构成的色彩空间，即BGRA色彩空间。在BGRA色彩空间中，alpha通道在区间[0, 255]内取值；其中，0表示透明，255表示不透明。

例 调整A通道的值

编写一个程序，首先将图像从BGR色彩空间转换到BGRA色彩空间；然后拆分BGRA图像中的通道；接着把BGRA图像的透明度调整为172后，合并拆分后的通道图像；再接着把BGRA图像的透明度调整为0后，合并拆分后的通道图像；最后分别显示BGRA图像、透明度为172的BGRA图像和透明度为0的BGRA图像。

案例代码：

```
import cv2
bgr_image = cv2.imread("flower.jpg")
cv2.imshow("flower", bgr_image)
# 图flower从BGR色彩空间转换到BGRA色彩空间
bgra_image = cv2.cvtColor(bgr_image, cv2.COLOR_BGR2BGRA)
cv2.imshow("bgra", bgra_image)
b, g, r,a = cv2.split(bgra_image) # 拆分BGRA图像中的通道
a[:, :] = 172 # 将透明度调整为172
bgra_172 = cv2.merge([b, g, r,a]) # 合并拆分后的通道图像
cv2.imshow("bgra_172", bgra_172)
a[:, :] = 0 # 将透明度调整为0
bgra_0 = cv2.merge([b, g, r,a]) # 合并拆分后的通道图像
cv2.imshow("bgra_0", bgra_0)
cv2.waitKey()
cv2.destroyAllWindows()
```

以上运行结果显示图像的效果是一样的。为了显示3幅图像的不同效果，需要使用imwrite()方法将3幅图像保存下来，代码修改如下：

```
import cv2
bgr_image = cv2.imread("flower.jpg")
# 图3.1从BGR色彩空间转换到BGRA色彩空间
bgra_image = cv2.cvtColor(bgr_image, cv2.COLOR_BGR2BGRA)
b, g, r,a = cv2.split(bgra_image) # 拆分BGRA图像中的通道
a[:, :] = 172 # 将透明度调整为172
bgra_172 = cv2.merge([b, g, r,a]) # 合并拆分后的通道图像
a[:, :] = 0 # 将透明度调整为172
bgra_0 = cv2.merge([b, g, r,a]) # 合并拆分后的通道图像
cv2.imwrite("bgra.png",bgra_image)
cv2.imwrite("bgra_172.png",bgra_172)
cv2.imwrite("bgra_0.png",bgra_0)
cv2.waitKey()
cv2.destroyAllWindows()
```

运行结果：

这个位置是一幅
完全透明的图像

小结

本节内容：

1. 色彩空间:

   GRAY

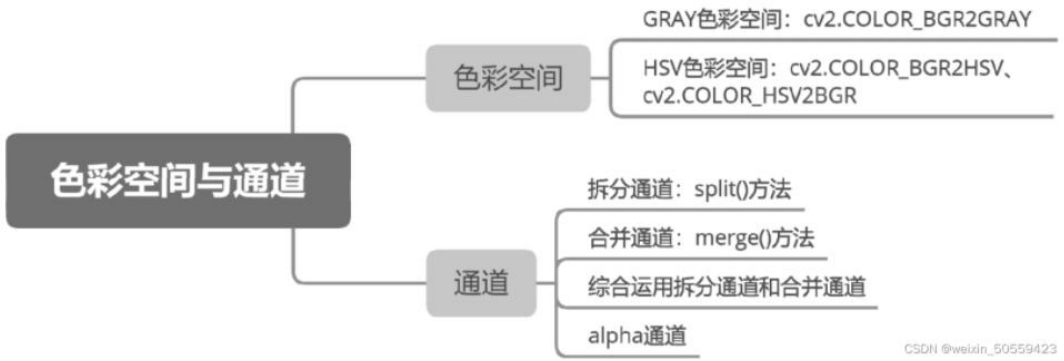
   HSV

2. 通道：

   拆分通道

   合并通道

   alpha通道



THANKS