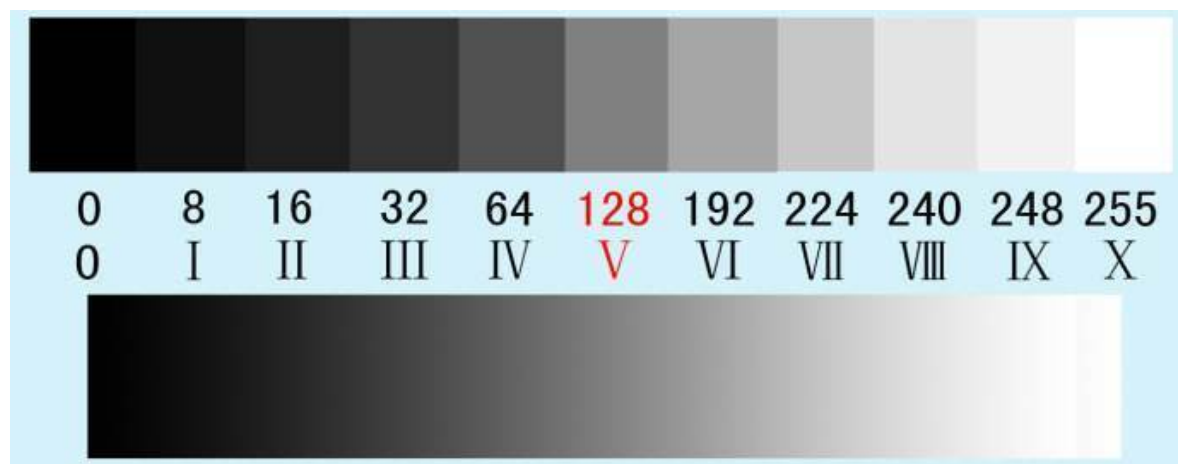


# TASK3 颜色空间转换

## 一、灰度图和RGB

图像是由很多个小格子组成的，每个小格子都只有一种颜色，这是构成图像的最小单元——像素（pixel）。不同的像素值代表了不同的颜色，像素值的值域一般在0到255（包括）之间，也就是256个整数，因此可以用完整个unsigned char类型的值域，所以像素值一般都是用unsigned char类型表示。

但是0-255不能映射到彩色，而只是对应黑色到白色之间的灰度值（grayscale），如图所示：



要表示彩色像素，可以使用RGB。因此，可以使用3个维度，也就是3个图像通道表示每个像素，即每个像素值用3个数字表示。如(255,255,255)表示白色，(255,0,0)表示红色，(255,255,0)表示黄色。

## 二、RGB转灰度图

RGB是三通道的，灰度只有一个通道，很自然的会联想到怎么把三个通道“融合”成一个通道。

对于RGB转灰度图，有一个心理学公式：

$$Gray = R * 0.299 + G * 0.587 + B * 0.114$$

实现代码：

```
1 import cv2
2 import numpy as np
3
4 def rgb2gray_mean(img):
5     # ratio = 1.0 / 3
6     # 转换类型
7     int_img = img.astype(np.int32)
8     print(int_img.shape)
9     result = 0.229*int_img[...,0] + 0.587*int_img[...,1] +
10 0.114*int_img[...,2]
11     return result.astype(np.uint8)
12
13 # 程序入口
14 def main():
15     # 读取lena图
16     color =
17     cv2.imread('E:/PythonProgram/opencv_study/fig_transaction/yoona.jpg')
```

```

16     # 转灰度
17     gray = rgb2gray_mean(color)
18     gray2 = cv2.cvtColor(color, cv2.COLOR_BGR2GRAY)
19
20     # 显示
21     cv2.imshow('color', color)
22     cv2.imshow('gray', gray)
23     cv2.imshow('gray2', gray2)
24     cv2.waitKey(0)
25
26     if __name__ == '__main__':
27         main()

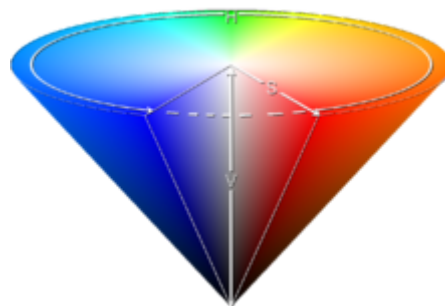
```

### 三、HSV空间

HSV是一种将RGB色彩空间中的点在倒圆锥体中的表示方法。HSV即色相(Hue)、饱和度(Saturation)、明度(Value)，又称HSB(B即Brightness)。色相是色彩的基本属性，就是平常说的颜色的名称，如红色、黄色等。饱和度(S)是指色彩的纯度，越高色彩越纯，低则逐渐变灰，取0-100%的数值。明度(V)，取0-max(计算机中HSV取值范围和存储的长度有关)。HSV颜色空间可以用一个圆锥空间模型来描述。圆锥的顶点处，V=0，H和S无定义，代表黑色。圆锥的顶面中心处V=max，S=0，H无定义，代表白色。

RGB颜色空间中，三种颜色分量的取值与所生成的颜色之间的联系并不直观。而HSV颜色空间，更类似于人类感觉颜色的方式，封装了关于颜色的信息：“这是什么颜色？深浅如何？明暗如何？”

#### HSV模型

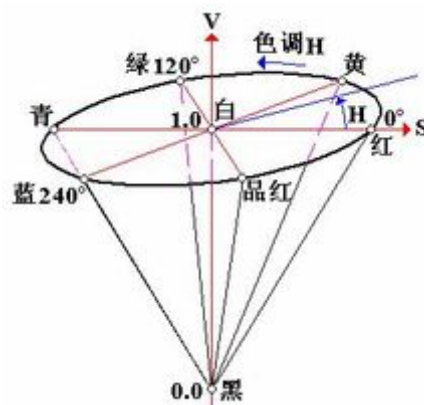


这个模型就是按色彩、深浅、明暗来描述的。

H是色彩；

S是深浅，S = 0时，只有灰度；

V是明暗，表示色彩的明亮程度，但与光强无直接联系。



应用：可以用于偏光矫正、去除阴影、图像分割等

## 四、RGB与HSV转换

### 1.RGB2HSV

$$V \leftarrow \max(R, G, B)$$
$$S \leftarrow \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases}$$
$$H \leftarrow \begin{cases} 60(G - B)/(V - \min(R, G, B)) & \text{if } V = R \\ 120 + 60(B - R)/(V - \min(R, G, B)) & \text{if } V = G \\ 240 + 60(R - G)/(V - \min(R, G, B)) & \text{if } V = B \end{cases}$$

If  $H < 0$  then  $H \leftarrow H + 360$ . On output  $0 \leq V \leq 1, 0 \leq S \leq 1, 0 \leq H \leq 360$ .

### 2.HSV2RGB

$$h_i \equiv \left\lfloor \frac{h}{60} \right\rfloor \pmod{6}$$

$$f = \frac{h}{60} - h_i$$

$$p = v \times (1 - s)$$

$$q = v \times (1 - f \times s)$$

$$t = v \times (1 - (1 - f) \times s)$$

对于每个颜色向量  $(r, g, b)$ ,

$$(r, g, b) = \begin{cases} (v, t, p), & \text{if } h_i = 0 \\ (q, v, p), & \text{if } h_i = 1 \\ (p, v, t), & \text{if } h_i = 2 \\ (p, q, v), & \text{if } h_i = 3 \\ (t, p, v), & \text{if } h_i = 4 \\ (v, p, q), & \text{if } h_i = 5 \end{cases}$$

代码实现：

```
1 import cv2
2 import numpy as np
3
4 def rgb2gray_mean(img):
5     # ratio = 1.0 / 3
6     # 转换类型
7     int_img = img.astype(np.int32)
8     print(int_img.shape)
9     result = 0.229*int_img[...,0] + 0.587*int_img[...,1] +
0.114*int_img[...,2]
10     return result.astype(np.uint8)
11
12 # 程序入口
13 def main():
```

```
14     # 读取lena图
15     color =
cv2.imread('E:/PythonProgram/opencv_study/fig_transaction/yoona.jpg')
16     # 转灰度
17     gray = rgb2gray_mean(color)
18     gray2 = cv2.cvtColor(color, cv2.COLOR_BGR2GRAY)
19     hsv = cv2.cvtColor(color, cv2.COLOR_BGR2HSV)
20
21     # 显示
22     cv2.imshow('color', color)
23     cv2.imshow('gray', gray)
24     cv2.imshow('gray2', gray2)
25     cv2.imshow('hsv', hsv)
26     cv2.waitKey(0)
27
28 if __name__ == '__main__':
29     main()
```