

# 7 图像处理-颜色空间学习

## 1.颜色空间介绍

我们所看到的图像信息，其实每一帧都是由B、G、R三个颜色分量所构成的像素点排列而成。

颜色模型也被称为颜色空间，是用一组数值来描述颜色的数学模型。

RGB图像是一种比较常见的颜色空间类型，除此以外，还有一些还有一些其他的颜色空间，比较常见的包括 GRAY颜色空间（灰度图像）、Lab颜色空间、XYZ颜色空间、YCrCb颜色空间、HSV颜色空间、HLS颜色空间、CIEL\*a\*b\*颜色空间、CIEL\*u\*v\*颜色空间、Bayer颜色空间等。

每个颜色空间都有自己擅长的处理问题的领域，因此，为了更方便地处理某个具体问题，就要用到颜色空间类型转换。

颜色空间类型转换是指，将图像从一个颜色空间转换到另外一个颜色空间。例如，在使用OpenCV处理图像时，可能会在RGB颜色空间和Lab颜色空间之间进行转换。在进行图像的特征提取、距离计算时，往往先将图像从RGB颜色空间处理为灰度颜色空间。在一些应用中，可能需要将颜色空间的图像转换为二值图像。

下面将列举介绍几种常见的空间。

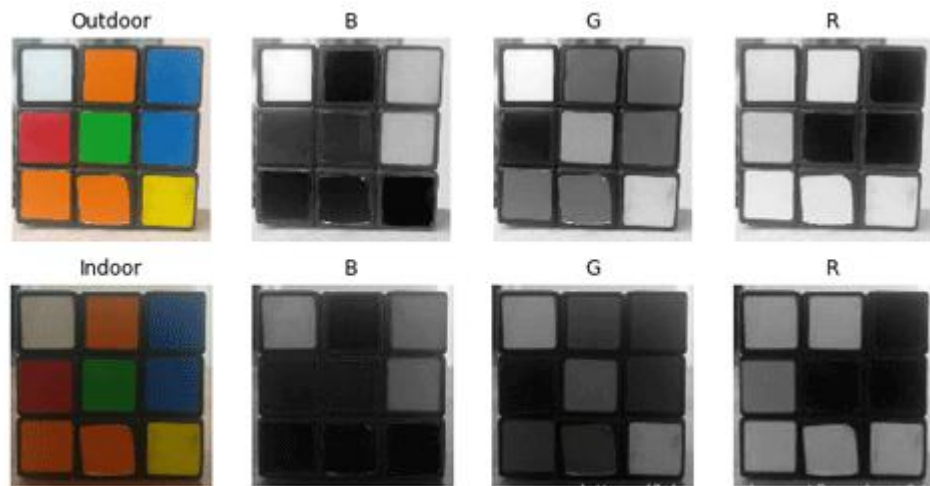
## 2.常用颜色空间类型学习

### 2.1 RGB颜色空间

RGB颜色空间具有以下属性：

- 1.它是一种加色空间，其中颜色通过红色R，绿色G和蓝色B的线性组合获得。
- 2.物体光照会影响该颜色空间各个通道值，三个颜色通道是具有相关性的。让我们将上面图像分成R，G和B分量并观察它们以更深入地了解颜色空间。

从下图可以看到如果你看蓝色通道,可以看到在室内光照条件下第二张图像中的魔方蓝色和白色部分看起来相似,但第一张图像有明显差异。这种不均匀性使得在该颜色空间中基于颜色的分割非常困难。此外,两个图像的值之间存在总体差异。因此RGB颜色空间存在颜色值分布不均匀以及色度和亮度混合在一起的问题。



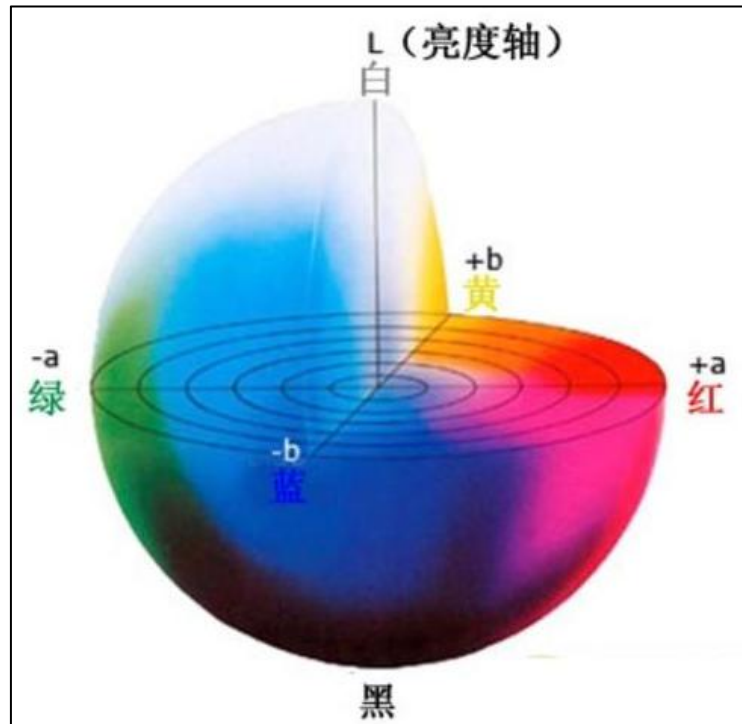
## 2.2 Lab颜色空间

类似RGB空间, Lab也有三个图像通道。

L: 亮度通道, 表亮度。

a: 颜色通道a, 表示从绿色到洋红色的颜色。

b: 颜色通道b, 表示从蓝色到黄色的颜色。



Lab颜色空间与RGB颜色空间完全不同。在RGB颜色空间中，颜色信息被分成三个通道，但是相同的三个通道也包含亮度信息。另一方面，在Lab颜色空间中，L通道独立于颜色信息并仅只含亮度信息。另外两个通道编码颜色。

L分量：表示像素的亮度。L值越大，亮度越高。

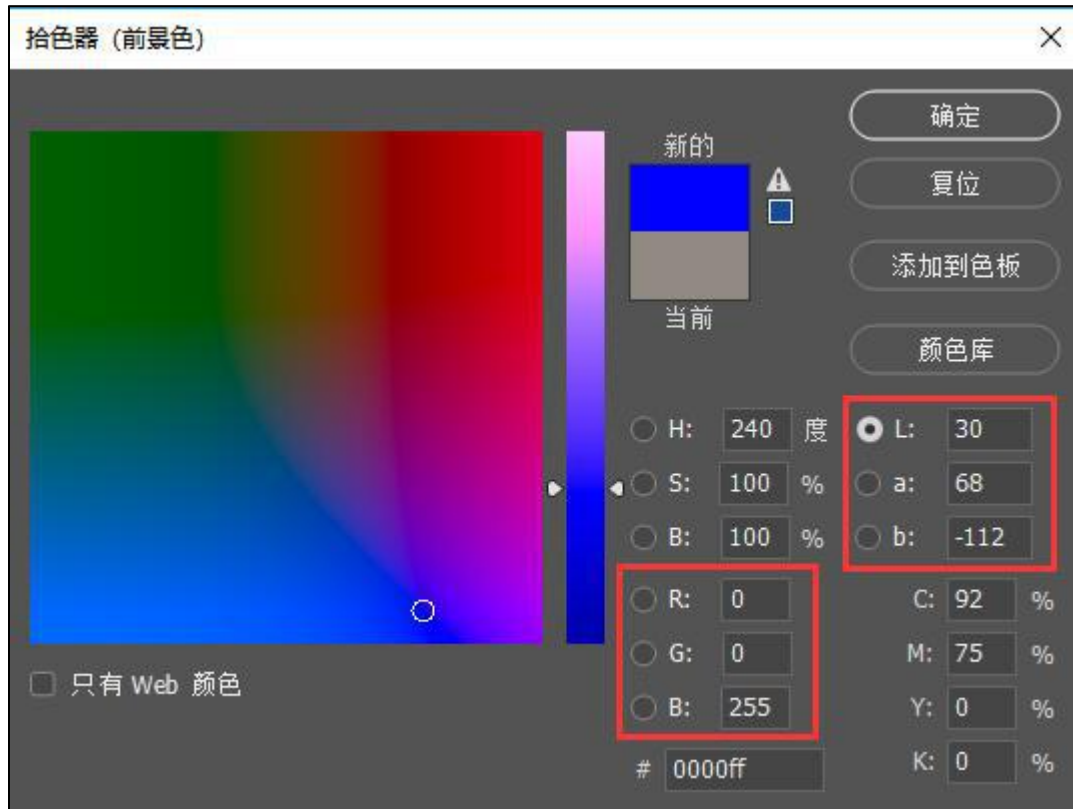
a分量：表示从红色到绿色的范围。

b分量：表示从黄色到蓝色的范围。

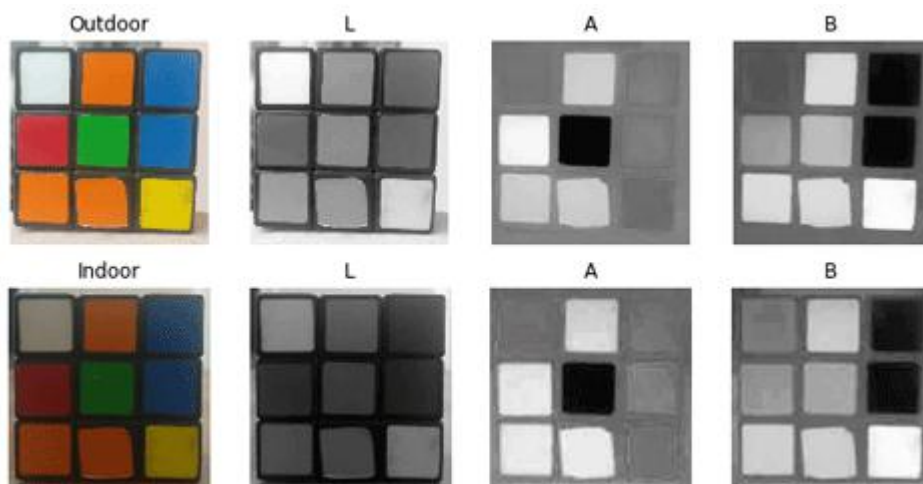
在OpenCV里，RGB颜色空间中R、G、B的数值范围均为[0-255]。在Lab颜色空间中的L数值范围是[0-100]，L为0时代表黑色，为100时代表白色。a和b的数值范围是[-128,127]，a和b为0时都代表灰色。

为了进一步帮助大家了解RGB和Lab的对照关系，这里以PS软件为例进行说明：

- 1) 在 PS 软件中使用“吸管工具”，选择需要吸取的颜色；
- 2) 点击左下方“拾色器”，可以看到 Lab 与 RGB 的对应关系如下图所示：



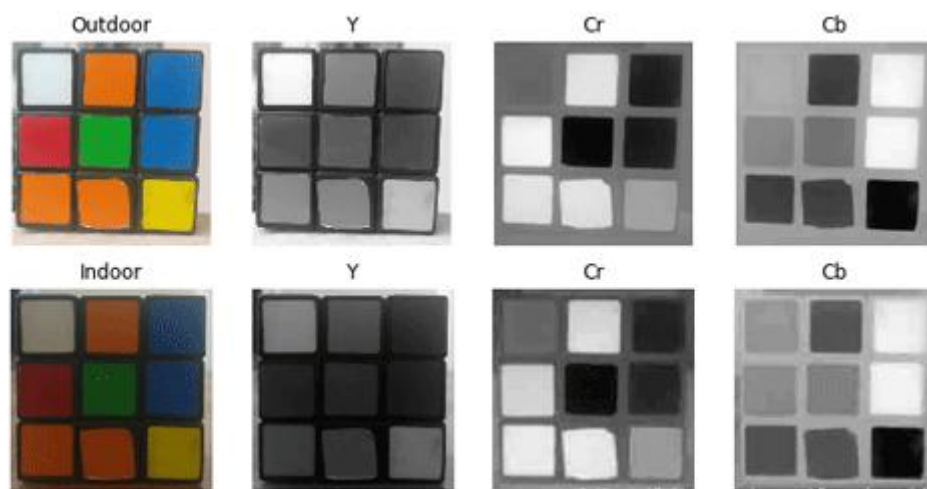
Lab颜色空间还具有以下特性：1.感知上均匀的颜色空间近似于我们如何感知颜色。2. 独立于设备（捕获或显示）。3.广泛用于Adobe Photoshop。4.通过复数变换方程与RGB颜色空间相关。 OpenCV中读取图像，转换为Lab空间图像结果如下图所示：



## 2.3 YcrCb颜色空间

人眼视觉系统（HVS，Human Visual System）对颜色的敏感度要低于对亮度的敏感度。在传统的 RGB 颜色空间内，RGB 三原色具有相同的重要性，但是忽略了亮度信息。

在YCrCb颜色空间中，Y代表光源的亮度，色度信息保存在Cr和Cb中，其中，Cr表示红色分量信息，Cb表示蓝色分量信息。亮度给出了颜色亮或暗的程度信息，该信息可以通过照明中强度成分的加权和来计算。在RGB光源中，绿色分量的影响最大，蓝色分量的影响最小。

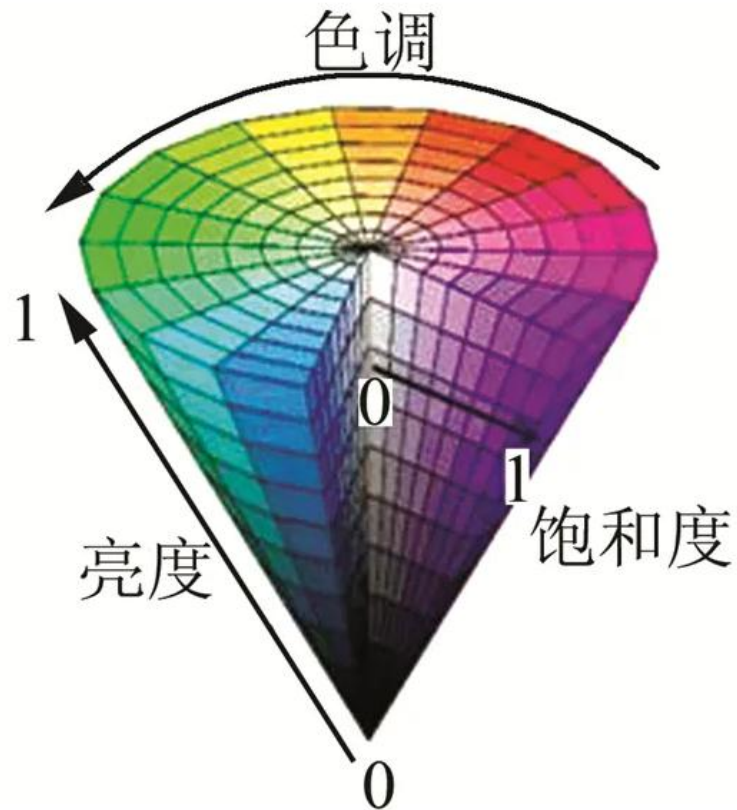


对于照度变化，可以针对强度和颜色分量对LAB进行类似的观察。与LAB相比，室外图像中红色和橙色之间的感知差异较小，白色在所有3个组件中发生了变化。

## 2.4 HSV颜色空间

HSV颜色空间是一种面向视觉感知的颜色模型，具有以下三个成分：

H色调，S饱和度，V明度。



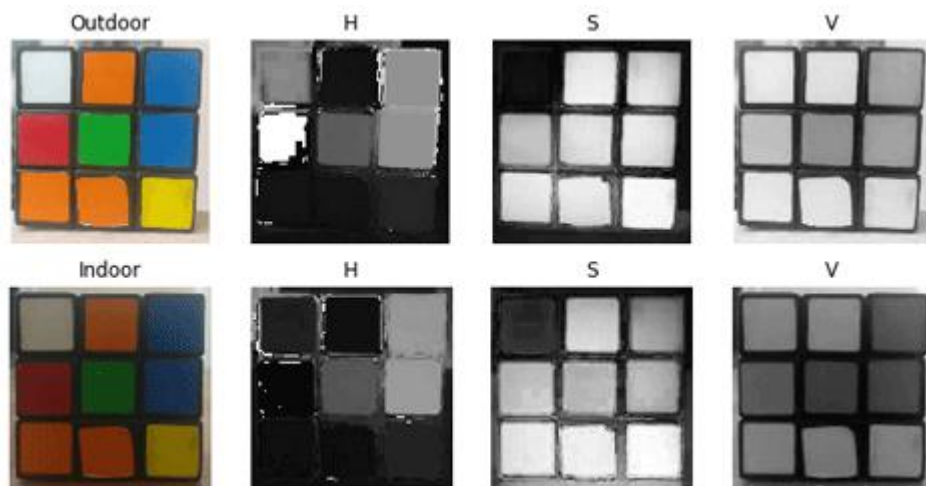
色调：色调与混合光谱中的主要光波长相关，例如“赤橙黄绿青蓝紫”分别表示不同的色调。如果从波长的角度考虑，不同波长的光表现为不同的颜色，实际上它们体现的是色调的差异。

饱和度：指相对纯净度，或一种颜色混合白光的数量。纯谱色是全饱和的，像深红色（红加白）和淡紫色（紫加白）这样的彩色是欠饱和的，饱和度与所加白光的数量成反比。

亮度：反映的是人眼感受到的光的明暗程度，该指标与物体的反射度有关。对于颜色来讲，如果在其中掺入的白色越多，则其亮度越高；如果在其中掺入的黑色越多，则其亮度越低。

HSV最大的特点是它只使用一个通道来描述颜色（H），这使得指定颜色变得非常直观。但是HSV颜色取决于设备。

从下图可以看到：



H分量在两个图像中非常相似，这表明即使在光照变化下颜色信息也是完整的。

两个图像中的S分量也非常相似，V分量表示亮度，因此它会因照明变化而发生变化。

红色室外和室内图像的值之间存在巨大差异。这是因为H值是以角度表示红色表示起始角度。因此它可能会取角度  $[300, 360]$  和  $[0, 60]$  之间的值。

## 2.5 Gray颜色空间

GRAY颜色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”表示白色，0~255之间的数值表示不同的亮度（即颜色的深浅程度）的深灰色或者浅灰色。

## 3.颜色转换实战

在了解Lab颜色模型后，接下来看下OpenCV中颜色转换的函数：

```
dst = cv2.cvtColor( src, code [, dstCn] )
```

dst表示输出图像，与原始输入图像具有同样的数据类型和深度。src表示原始输入图像。

code是颜色空间转换的标志。

dstCn是目标图像的通道数，默认为0。

标记	简记	作用
cv.COLOR_BGR2BGRA	0	为RGB添加alpha通道
cv.COLOR_BGR2RGB	4	更改彩色图像通道颜色的顺序
cv.COLOR_BGR2GRAY	10	把彩色图像转成灰度图像
cv.COLOR_GRAY2BGR	8	把灰度图像转成彩色图像(伪彩色)
cv.COLOR_BGR2YUV	82	从RGB颜色空间转成YUV颜色空间
cv.COLOR_YUV2BGR	84	从YUV颜色空间转成RGB颜色空间
cv.COLOR_BGR2HSV	40	从RGB颜色空间转成HSV颜色空间
cv.COLOR_HSV2BGR	54	从HSV颜色空间转成RGB颜色空间
cv.COLOR_BGR2Lab	44	从RGB颜色空间转成Lab颜色空间
cv.COLOR_Lab2BGR	56	从Lab颜色空间转成RGB颜色空间

例如：cv2.cvtColor(frame, cv2.COLOR\_RGB2LAB)

其中“frame”为需要进行处理的图片；“cv2.COLOR\_RGB2LAB”为指定颜色转换模



式，这里是将RGB颜色空间的frame信息转换为LAB颜色空间。

下面我们就以一张图片为例，通过转换几种常用的颜色空间来查看效果。

### 3.1 实验步骤



开始操作前，需要先将目录“第4章 OpenCV计算机视觉学习->图像处理进阶篇->第7课 图像处理—颜色空间转换->例程源码”下的例程“color\_conversion.py”和示例图片“img1.jpg”复制到共享文件夹。

关于共享文件夹的配置方法，可参考目录“第2章 Linux系统简介及使用入门->Linux基础课程->第3课 Linux系统安装及换源方法”下的文档。

---

注意：输入指令时需严格区分大小写，且可以使用“Tab”键补齐关键字。

---

1) 打开虚拟机，启动系统。点击系统任务栏的图标，并点击图标，或使用快捷键“Ctrl+Alt+T”，打开命令行终端。

2) 输入指令“cd /mnt/hgfs/Share/Image”，并按下回车，进入共享文件夹。

```
hiwonder@ubuntu:~$ cd /mnt/hgfs/Share/Image/
```

3) 输入指令“python3 color\_conversion.py”，并按下回车，运行例程。

```
hiwonder@ubuntu:/mnt/hgfs/Share/Image$ python3 color_conversion.py
```

### 3.2 实验效果

执行程序后，图像处理的结果如下图所示：



### 3.3 实验分析

- 1) 首先导入模块，通过import语句导入所需模块。

```
1 import numpy as np
2 import cv2 as cv
```

- 2) 通过调用cv2模块中的imread()函数，读取用于缩放处理的图像。

```
4 src = cv2.imread("img1.jpg")
```

- 3) 函数括号内的参数是图像名称，然后设置插入图片的长宽大小。如下图所示：

```
5 src = cv2.resize(src, (int(src.shape[1] / 2), int(src.shape[0] / 2)))
```

- 4) 下面依次创建四个函数，分别来转换Gray、Lab、Ycrcb、HSV的颜色空间。

```
7 GRAY = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
8 Lab = cv2.cvtColor(src, cv2.COLOR_BGR2LAB)
9 YCrCb = cv2.cvtColor(src, cv2.COLOR_BGR2YCrCb)
10 HSV = cv2.cvtColor(src, cv2.COLOR_BGR2HSV)
```

- 5) 接着将转换前及转换后的效果分别显示出来。

```
12 cv2.imshow("src", src)
13 cv2.imshow("GRAY", GRAY)
14 cv2.imshow("Lab", Lab)
15 cv2.imshow("YCrCb", YCrCb)
16 cv2.imshow("HSV", HSV)
```

6) 最后通过函数来关闭窗口。

```
18 cv2.waitKey(0)
19 cv2.destroyAllWindows()
```

`cv2.waitKey()`是一个键盘绑定函数。它的时间量度是毫秒ms。函数会等待（n）里面的n毫秒，看是否有键盘输入。若有键盘输入，则返回按键的ASCII值。没有键盘输入，则返回-1。一般设置为0，将无线等待键盘的输入。

`cv2.destroyAllWindows()` 用来删除窗口的，()里不指定任何参数，则删除所有窗口，删除特定的窗口，往()输入特定的窗口值。