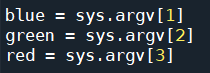
opencv-python颜色检测实验

一.预备知识介绍：

bgr\_hsv\_converter.py：

①sys.argv[]是一个从程序外部获取参数的桥梁，可以看作是一个列表，可以在命令行中为此列表赋值





如blue得到了列表的第一个值145

②opencv中处理图片颜色的格式通常为HSV格式，且opencv提供了BGR-HSV或BGR-Gray的转换函数cv2.cvtColor(img,flag)，flag分别对应于cv2.COLOR\_BGR2GRAY或cv2.COLOR\_BGR2HSV【见名知意】



③uint8是专门用于存储各种图像的格式（包括RGB，灰度图像等），范围是从0–255，np.uint8()函数会将数字转换为此格式，截断超过255的值



colorDetection.py：

①cv2.resize函数可缩放图像，如下代码将图像的x,y方向均缩小为原来的20%



②cv2.inRange (hsv, lower\_range, upper\_range)函数：

第二个参数：lower\_range指的是图像中低于这个lower\_range的值，图像值变为0（黑色）

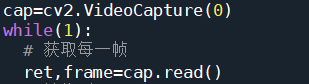
第三个参数：upper\_range指的是图像中高于这个upper\_range的值，图像值变为0（黑色）

而在lower\_red～upper\_red之间的值变成255



camera\_colorDetection.py：

①对于摄像头捕获的视频，都需要循环一帧一帧地来处理，相当于把视频分解成了图片



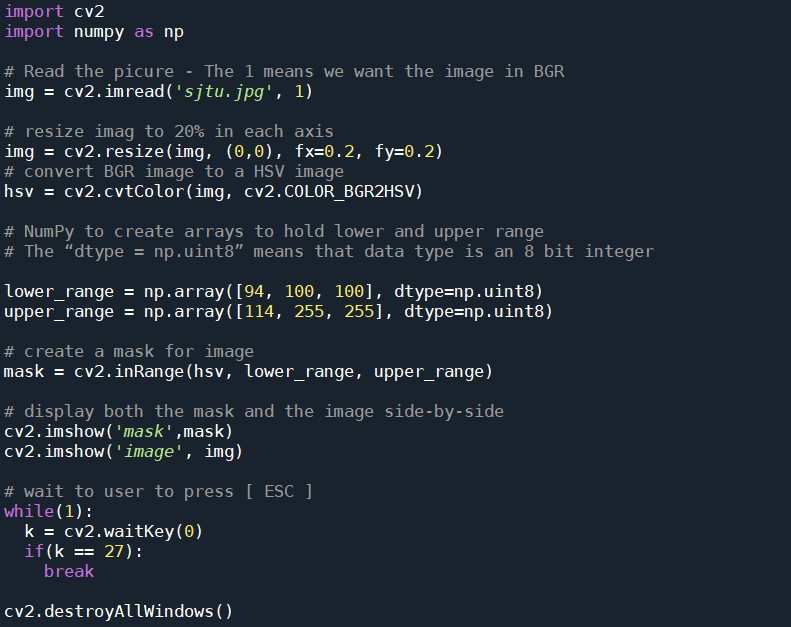
②cv2.bitwise\_and



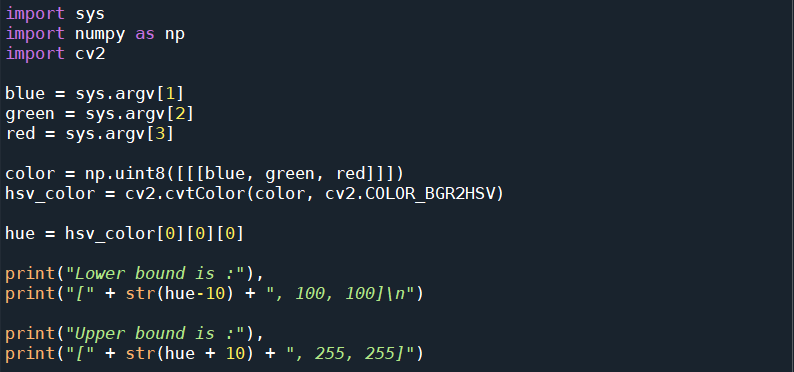
对原图像和掩模进行位运算，此处将原视频非检测颜色之外的内容都变为黑色

二.实验代码：

colorDetection.py：



bgr\_hsv\_converter.py：



camera\_colorDetection.py：

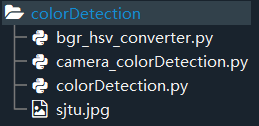


三.实验步骤：

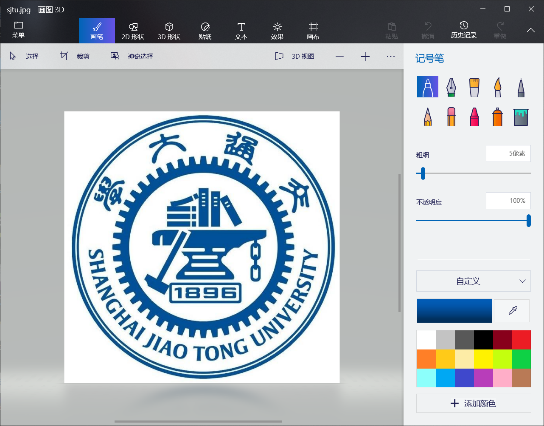
①我们找到一张图像，以上海交通大学校徽为例：



将其命名为’sjtu.jpg’并放在颜色检测实验的colorDetection文件夹下：

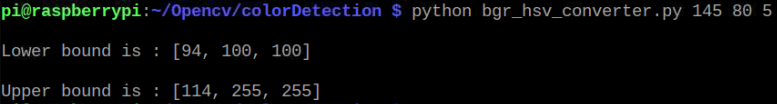


②利用windows系统自带的画图3D软件打开该图片，并使用取色器取得蓝色部分RGB值为[5,80,145]（不同区域会有一定差异）



③进入colorDetection所在文件夹后运行bgr\_hsv\_converter.py并将得到的BGR参数145 80 5输入（注意是BGR，可以看作RGB逆着写）

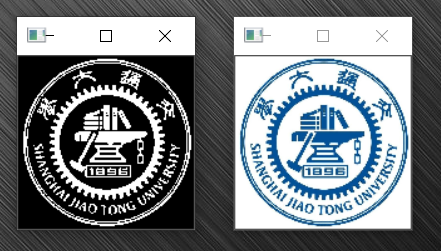
树莓派终端命令行如下：



得到HSV模型上界114，下界94，并将colorDetection.py中的相应项改为得到的值



④运行程序，得到颜色识别结果，非蓝色部分被黑色遮盖：



⑤运行camera\_colorDetection.py，对摄像头捕获到的视频进行颜色检测

注意需调整要识别颜色的阈值（此处为红色阈值）：



此阈值获得方法与获得图片颜色阈值的方法一致

检测效果如图：

