虚谷号结合摄像头制作智能变色灯

浙江省温州中学 谢作如

温州大学教师教育学院 林淼焱

涉及学科：物理、技术、艺术

DF论坛上有人发了一篇关于制作“变色龙”的文章，利用RGB全彩LED灯珠或灯带，结合颜色识别传感器，做一个能够根据环境颜色发出不同光芒的“变色龙”。这一作品激起很多创客的兴趣，于是大家就进一步改造，让这个“会变色”的作品越来越好玩。

这类变色主题的作品再多，也无非是利用了颜色传感器，控制的还是REG灯，在我们看来都不够好玩。在研究利用Wi-Fi控制互联网智能灯泡的过程中，我们发现只要遵循一定的协议，控制物联网家电并非难事。于是，本文继续研究用虚谷号来控制Yeelight灯泡，做一个智能变色灯。和其他作品不同的是，我们用摄像头或者高拍仪替代了颜色传感器，并且控制的是220V“真正灯泡”。



DF论坛上的“变色龙”作品

一、作品原理分析

之所以用摄像头来替代颜色传感器，是因为前者不仅便宜，而且随处可以获取。利用摄像头获取颜色，可以借助于Python中OpenCV库，调用摄像头拍照，并对拍到的图像进行处理分析，将图像中的主要颜色分解为RGB三种颜色数值，再通过局域网发送控制指令使智能灯泡变色。

用虚谷号控制智能灯泡的原理，在上一期文章《让创客空间的照明系统个性化起来》中已经详细介绍，这里不再详细介绍。Yeeligh灯泡支持局域网控制，能够接收到局域网内任一智能设备的控制指令，发送控制指令的设备可以是电脑、手机，也可以是任何一个支持网络功能的MCU，如掌控板、虚谷号和树莓派等。

这个智能变色灯需要的器材很少，虚谷号加yeelight智能灯泡、摄像头就可以了，作品的工作流程图如下。

虚谷号

摄像头

获取物体颜色

发送控制指令

智能灯泡的工作流程图

二、用摄像头获取物体颜色的代码实现

摄像头前物体是什么颜色？看起来问题很简单，但是让计算机来回答并不容易。当我们要判断某个物体是什么颜色的时候，往往会先找出这一物体的最主要颜色，再判断这一颜色和哪种颜色最接近。计算机虽然说不出天蓝、橘黄之类的色彩名词，但可以精确地分析出图像中某一个像素点的RGB值，即RGB三种颜色的具体组成，但一个物体往往不会是纯色的，所以要确定一张含有多种颜色的图像的“颜色”，需要确定一个算法。

算法一：将这个图像的所有像素点的RGB值分别相加，取出RGB三种颜色的平均值。如果担心计算的效率太低，可以均匀地取出部分像素，应该可以得到大致的颜色平均值。

算法二：利用滤镜功能，给这个图像加上RGB三种颜色遮罩，处理为黑白图片，然后通过计算分别得到图像的白色区域面积大小，换算为0-255之间的数值。

|  |  |
| --- | --- |
|  |  |
| 图像处理原图 | 图像处理1 |

其实，这两种算法的原理是差不多的，只不过实现的方式不一样。OpenCV库作为著名的跨平台计算机视觉库，提供了很多图像处理和计算机视觉方面的通用算法，可以很简单地帮我们完成颜色分析的功能。核心代码如下：

|  |
| --- |
| def get\_pic(): #调用摄像头拍摄图像  cap=cv2.VideoCapture(0)  sucess,img=cap.read()  cv2.imwrite("CachePhoto/image.jpg",img)  cap.release()  def get\_color(frame): #处理图像，返回颜色值  maxsum = -100  color = None  #将RGB图像转化为HSV图像，HSV空间比在BGR空间中更容易表示一个特定的颜色  hsv = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)  #导入预设的颜色字典，即RGB三种颜色  color\_dict = colorList.getColorList()  #遍历颜色字典，选取不同的颜色来与原图进行比较  for d in color\_dict:  #为当前颜色添加遮罩，变成白色，其他颜色变成黑色  mask = cv2.inRange(hsv, color\_dict[d][0], color\_dict[d][1])  #将这张黑白照片存入本地以便调试查看  cv2.imwrite('CachePhoto/' + d +'.jpg', mask)  #为了保证图像的质量，对这张黑白照片进行降噪处理，凸显轮廓  binary = cv2.threshold(mask, 127, 255, cv2.THRESH\_BINARY)[1]  binary = cv2.dilate(binary, None, iterations=2)  #寻找轮廓函数，使用countours得到每张图像白色部分的面积  img, countours, hierarchy = cv2.findContours(binary.copy(), cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)  sum = 0  # cv2.contourArea()用于计算轮廓的面积，我们用它在代表不同颜色的许多图像中，找到白色轮廓面积最大的图像  for cnts in countours:  sum += cv2.contourArea(cnts)  if sum > maxsum:  maxsum = sum  color = d  return color |

三、虚谷号控制物联网灯泡的代码实现

为方便用户控制灯泡，Yeelight公司提供的一个小型库，可让用户通过Wi-Fi发送指令来控制Yeelight灯泡。只要获取了设备的IP地址，通过ON、OFF、Set\_Color等简单的函数，就能调整灯泡的开关、颜色、亮度等参数。

需要强调的是，Yeelight灯泡要开启“局域网控制”功能，默认情况下这一功能是关闭的。另外，虚谷号和Yeelight灯泡一定要处于同一局域网，即连接同一个无线路由器，才能通过指令来找到设备的IP地址。

控制灯泡颜色的参考代码如下：

|  |
| --- |
| #从yeelight库中导入Bulb类  from yeelight import Bulb  #实例化Bulb类，灯泡的IP地址为192.168.31.39  bulb = Bulb("192.168.31.39")  #yeelight特定函数：开启灯泡  bulb.turn\_on()  #yeelight特定函数：通过设置不同的RGB值设置灯光颜色  bulb.set\_rgb(red,green,blue) |

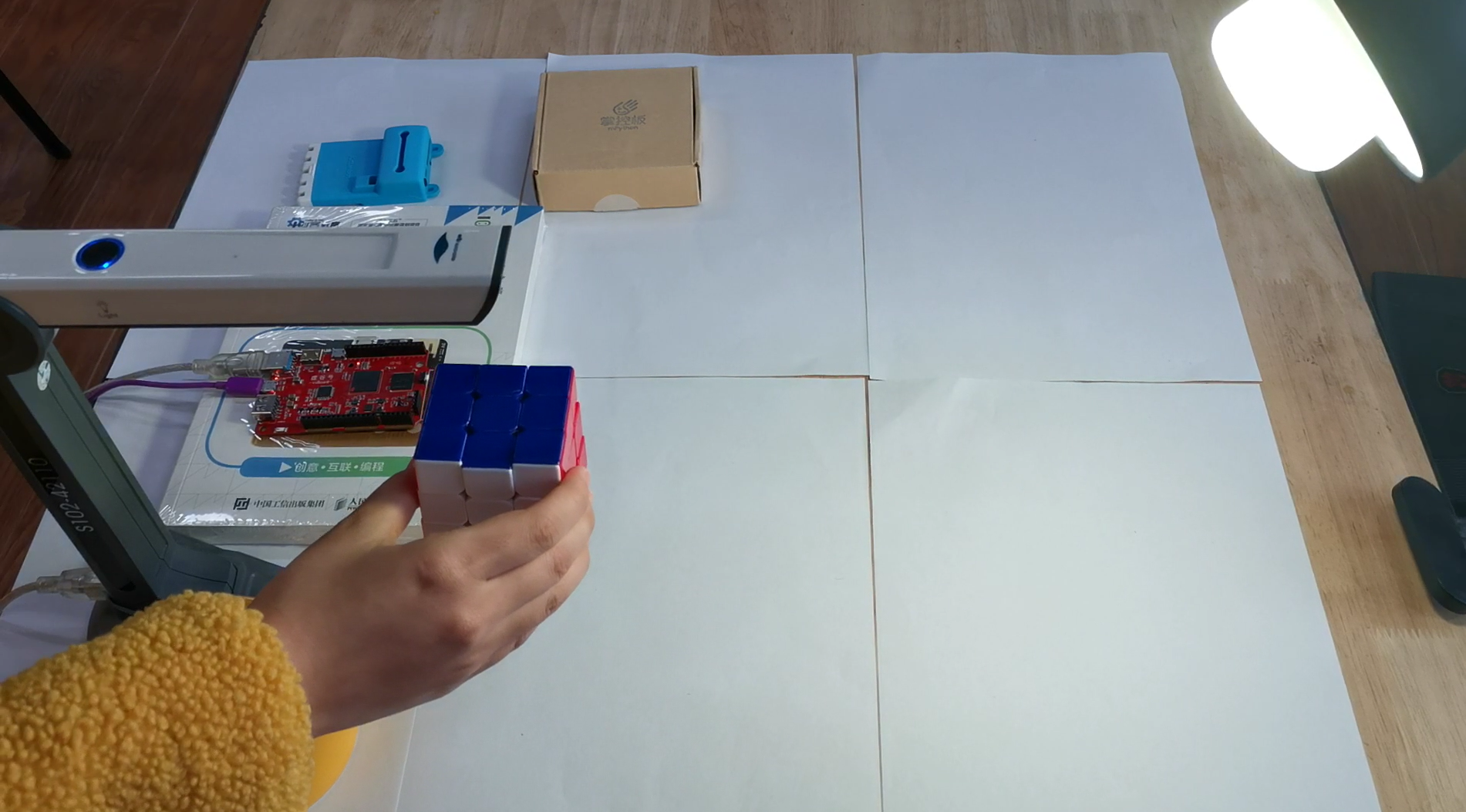
我们在虚谷号的论坛中，提供了更多关于Yeelight库的教程，地址如下：

http://vvboard.net.cn/bbs/

四、作品测试

为了让获取的物体颜色更加准确，我们用高拍仪替代了普通摄像头。因为高拍仪一般都自带了用来补光的LED灯，可以避免色差。虚谷号支持绝大多数的免驱摄像头和高拍仪，能够做到即插即用。我们使用的高拍仪是良田的S系列。

作品的演示效果如下图。默认情况下灯光是白色的，当在高拍仪下面放置不同的物体，灯光则会变成物体的颜色，识别速度很快。如果加上语音提示功能，这个作品就更加吸引人了。



从这个作品可以看出，借助Python的OpenCV库，摄像头获取物体颜色非常方便，代码简单易懂。开源硬件虚谷号，不仅能够用来编写Arduino、Python、Processing等应用，也可以用来做视觉识别、语言识别之类的人工智能应用。在制作这个作品的时候，我们并不需要给虚谷号接上显示器和鼠标、键盘，只需要把在电脑中调试成功的Python代码，复制到虚谷号的U盘中就可以正常执行，大大降低了应用门槛。

本文为2018 年度温州市中小学校科技创新项目《<物联网与科学探究创意实验>课程的开发》的阶段性成果，项目编号：2018ZXX01。