

8.12、色块识别

8.12、色块识别

8.12.1、实验目标

8.12.2、实验说明

8.12.3、实验效果

8.12.4、实验总结

8.12.1、实验目标

本节课主要学习颜色识别功能，根据色块的LAB值，框出相同颜色的物品。

本次实验的参考代码路径为：K210_Broad\05-AI\color_reg.py

8.12.2、实验说明

模块的出厂固件已经集成AI视觉算法模块，如果下载过其他固件，请烧录回出厂固件再进行实验。

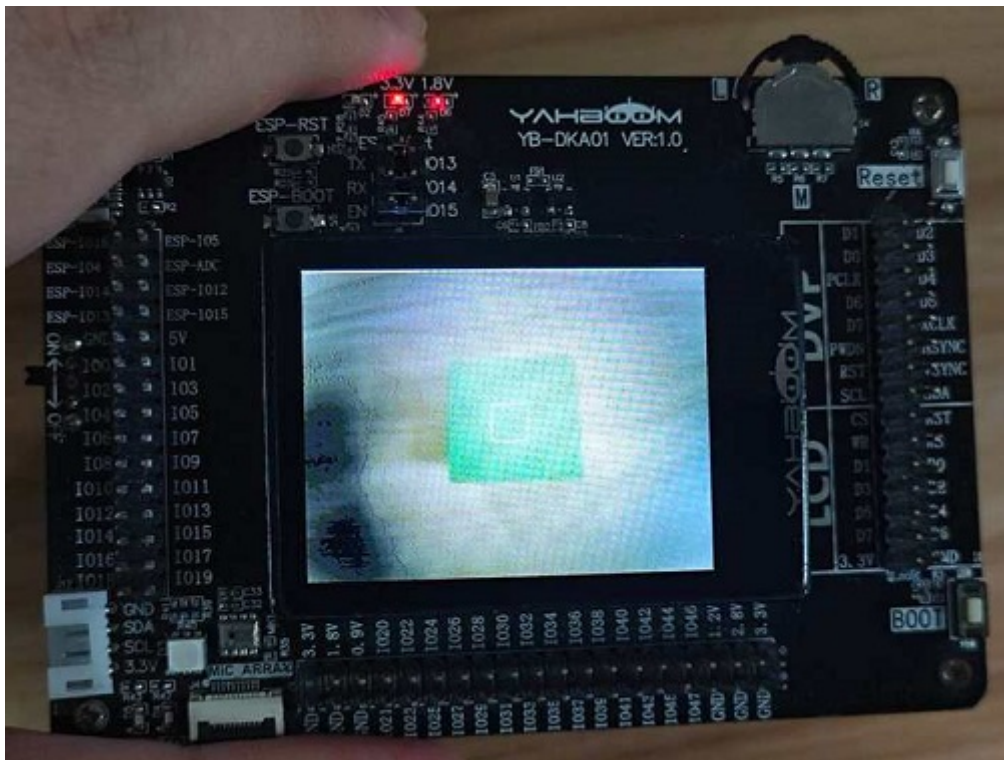
先打开本例程【color_reg.py】文件

修改下图中的地方，将learning_color()代码前面的#号给删除掉，然后在canmv里运行该程序

```
62
63 | #learning_color()
64
65
66 | #红色 color red
67 | threshold_red = [15, 50, 36, 73, 16, 49]
68 | #绿色 color green
69 | threshold_green = [25, 65, -47, -13, 10, 30]
70 | #蓝色 color blue
71 | threshold_blue = [24, 61, 17, 63, -78, -60]
72 | #黄色 color yellow
73 | threshold_yellow = [34, 68, -13, 6, 25, 48]
74
```

打开canmv里的串行终端，将K210中心的白色框框对准色块，等待一会儿就会在屏幕里输出该色块的LAB阈值

该过程最好在白色背景下、光线较好的环境下进行识别，识别过程中尽量保持不动让白框对准色块



同时串行终端也会打印出该颜色的LAB阈值

```

串行终端
[CANMV]: find ov9655
[CANMV]: find ov sensor
Hold the object you want to track in front of the camera in the box.
MAKE SURE THE COLOR OF THE OBJECT YOU WANT TO TRACK IS FULLY ENCLOSED BY THE BOX!
Learning thresholds...
threshold: [72, 77, -42, -27, 6, 23]
搜索结果  串行终端

```

同理，把四个颜色的色块都按照以上方法就可以获得四种颜色的阈值

最后将四个阈值都填写到此对应颜色的代码里，再将前面的learning_color代码加上#号注释掉，即可完成阈值设置

```

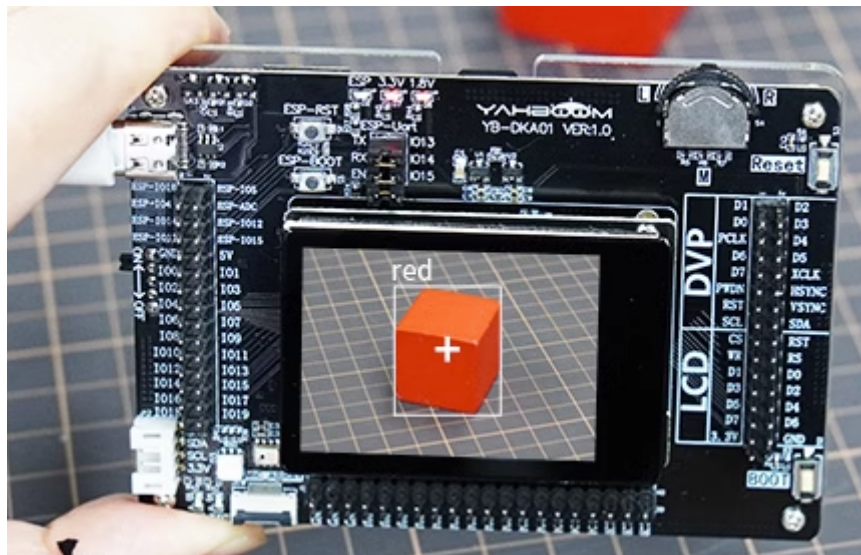
63 | #learning_color()
64 |
65 |
66 | #红色 color red
67 | threshold_red = [15, 50, 36, 73, 16, 49]
68 | #绿色 color green
69 | threshold_green = [25, 65, -47, -13, 10, 30]
70 | #蓝色 color blue
71 | threshold_blue = [24, 61, 17, 63, -78, -60]
72 | #黄色 color yellow
73 | threshold_yellow = [34, 68, -13, 6, 25, 48]
74 |

```

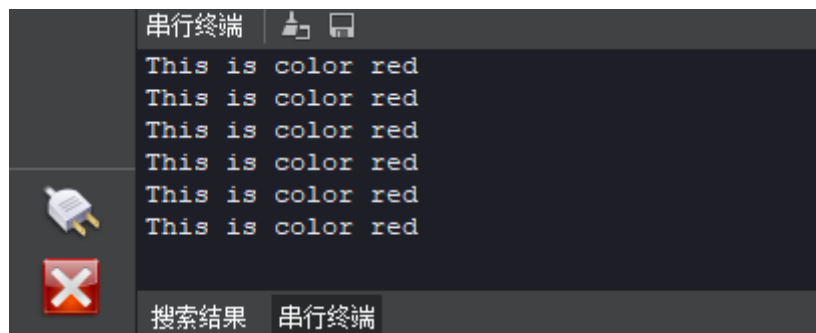
8.12.3、实验效果

将K210开发板通过TYPE-C数据线连接到电脑上，CanMV IDE点击连接按钮，连接完成后点击运行按钮，运行例程代码。也可以将代码作为main.py下载到K210开发板上运行。

运行后显示如下效果，同时主板上的RGB灯也会根据识别到的颜色发出不同颜色的光



如果是在canmv里运行可以打开串行终端看到所识别出来的颜色信息



8.12.4、实验总结

色块识别的功能主要是分析颜色的LAB值，将系统设定好的LAB阈值与摄像头采集到的颜色的LAB阈值作为分析对比，如果符合要求则画出方框，表示识别到该颜色，同时RGB灯也会显示该颜色的光。由于识别颜色存在误差，最好是在识别颜色与背景颜色差距较大情况下识别，如果背景颜色与识别的颜色相近，误识别的几率会增加。