

8.1、颜色识别

8.1、颜色识别

8.1.1、实验目标

8.1.2、实验过程

8.1.3、实验效果

8.1.4、实验总结

8.1.1、实验目标

本节课主要学习颜色识别功能，根据颜色的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_recognition.py

8.1.2、实验过程

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。

1. 导入相关库，并初始化摄像头和LCD显示屏。

```
import sensor, image, time, lcd

lcd.init()
sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.skip_frames(time = 2000)
sensor.set_auto_gain(False)
sensor.set_auto_whitebal(False)
clock = time.clock()
```



2. 在摄像头画面中绘制一个50*50大小的白色方框，此时的功能为提示用户将要识别的颜色放入方框内。

```
r = [(320//2)-(50//2), (240//2)-(50//2), 50, 50]
for i in range(50):
    img = sensor.snapshot()
    img.draw_rectangle(r)
    lcd.display(img)
```



3. 等到方框又白色变成绿色，则开始学习颜色的LAB值，多次读取数值，并取平均值作为学习的LAB值结果。

```
print("Learning thresholds...")
threshold = [50, 50, 0, 0, 0, 0] # Middle L, A, B values.
for i in range(50):
    img = sensor.snapshot()
    hist = img.get_histogram(roi=r)
```



```

    lo = hist.get_percentile(0.01) # Get the CDF of the histogram at the 1% range
    (ADJUST AS NECESSARY)!
    hi = hist.get_percentile(0.99) # Get the CDF of the histogram at the 99%
    range (ADJUST AS NECESSARY)!
    # Average in percentile values.
    threshold[0] = (threshold[0] + lo.l_value()) // 2
    threshold[1] = (threshold[1] + hi.l_value()) // 2
    threshold[2] = (threshold[2] + lo.a_value()) // 2
    threshold[3] = (threshold[3] + hi.a_value()) // 2
    threshold[4] = (threshold[4] + lo.b_value()) // 2
    threshold[5] = (threshold[5] + hi.b_value()) // 2
    for blob in img.find_blobs([threshold], pixels_threshold=100,
    area_threshold=100, merge=True, margin=10):
        img.draw_rectangle(blob.rect())
        img.draw_cross(blob.cx(), blob.cy())
        img.draw_rectangle(r, color=(0,255,0))
    lcd.display(img)

```

4. 当颜色学习完成，新建一个while循环，开始识别摄像头画面中的颜色，分析与上一步学习到的颜色的LAB值是否相符合，如果符合则框出来对应的颜色块。

```

print("Thresholds learned...")
print("Start Color Recognition...")
while(True):
    clock.tick()
    img = sensor.snapshot()
    for blob in img.find_blobs([threshold], pixels_threshold=100,
    area_threshold=100, merge=True, margin=10):
        img.draw_rectangle(blob.rect())
        img.draw_cross(blob.cx(), blob.cy())
    lcd.display(img)
    print(clock.fps())

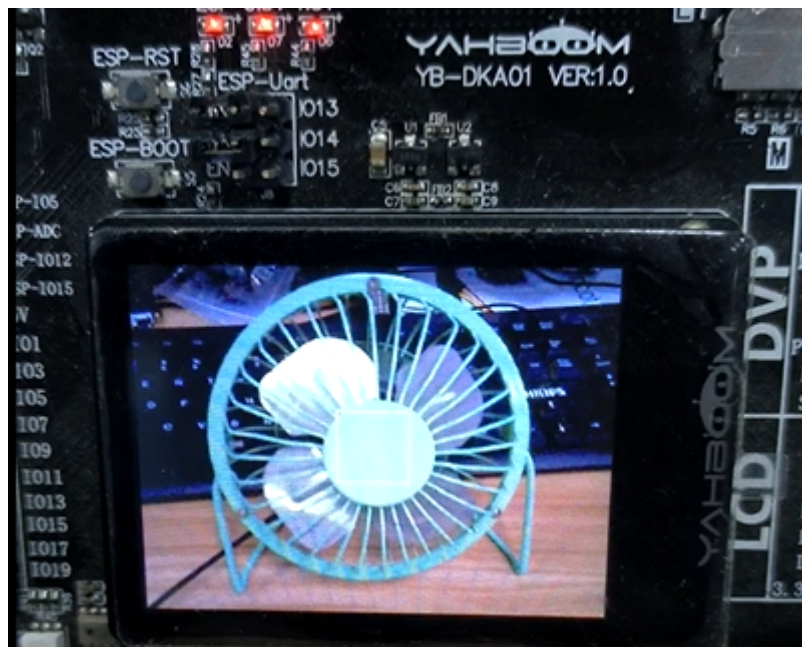
```



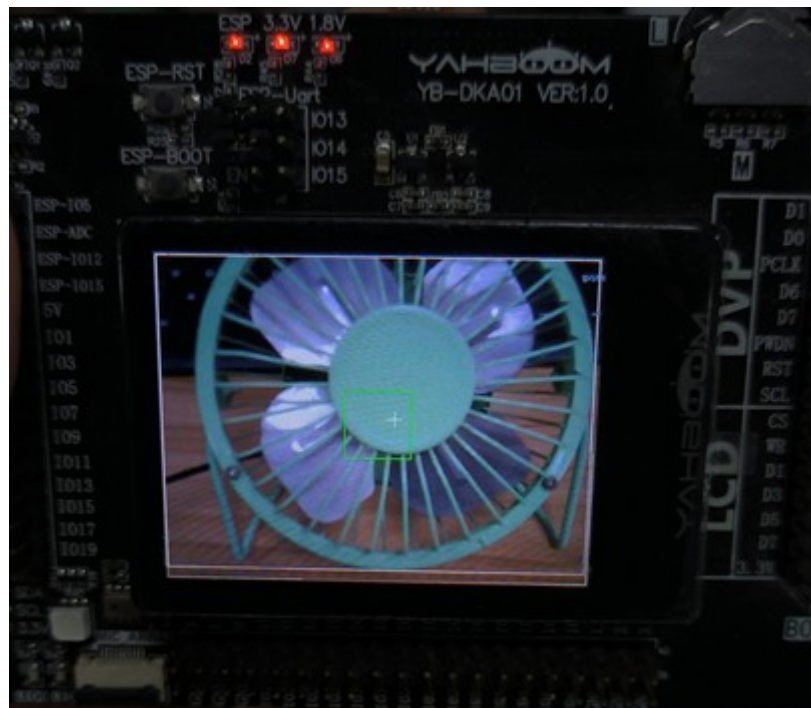
8.1.3、实验效果

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。

等待系统初始化完成后，LCD显示摄像头画面，并且屏幕中间有一个白色的方框，请将要识别的颜色放到白色方框内，白色方框持续时间大约3秒。



等到白色方框变为绿色方框时，此时系统开始学习绿色方框内的颜色LAB值，还会出现其他白色的方框作为预览效果，大约5秒后，绿色方框消失，则表示学习完成。



此时将摄像头朝向要识别的颜色，系统会自动框出识别的颜色。



8.1.4、实验总结

颜色识别的功能主要是分析颜色的LAB值，先把要识别的颜色放方框内，然后系统会根据方框内读取到的颜色的LAB值，再与摄像头采集到的颜色的LAB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。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。