

## 8.2、条形码识别

### 8.2、条形码识别

#### 8.2.1、实验目标

#### 8.2.2、实验过程

#### 8.2.3、实验效果

#### 8.2.4、实验总结

### 8.2.1、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\find\_barcodes.py

### 8.2.2、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，这里把摄像头的颜色设置成灰度。

```
import sensor, image, time, math, lcd

lcd.init()
sensor.reset()
sensor.set_pixformat(sensor.RGB565) #GRAYSCALE
sensor.set_framesize(sensor.QVGA)
sensor.skip_frames(time = 100)
sensor.set_auto_gain(False)
sensor.set_auto_whitebal(False)
clock = time.clock()
```



2. 转化条形码的类型名称为字符串。

```
def barcode_name(code):
    if(code.type() == image.EAN2):
        return "EAN2"
    if(code.type() == image.EAN5):
        return "EAN5"
    if(code.type() == image.EAN8):
        return "EAN8"
    if(code.type() == image.UPCE):
        return "UPCE"
    if(code.type() == image.ISBN10):
        return "ISBN10"
    if(code.type() == image.UPCA):
        return "UPCA"
    if(code.type() == image.EAN13):
```



```

        return "EAN13"
    if(code.type() == image.ISBN13):
        return "ISBN13"
    if(code.type() == image.I25):
        return "I25"
    if(code.type() == image.DATABAR):
        return "DATABAR"
    if(code.type() == image.DATABAR_EXP):
        return "DATABAR_EXP"
    if(code.type() == image.CODABAR):
        return "CODABAR"
    if(code.type() == image.CODE39):
        return "CODE39"
    if(code.type() == image.PDF417):
        return "PDF417"
    if(code.type() == image.CODE93):
        return "CODE93"
    if(code.type() == image.CODE128):
        return "CODE128"

```

3. 新建while循环，将摄像头采集的画面进行条形码分析，如果图像中有条形码，则将条形码的信息打印出来，并用绿色框出条形码的位置。

```

while(True):
    clock.tick()
    img = sensor.snapshot()
    fps = clock.fps()
    codes = img.find_barcodes()
    for code in codes:
        img.draw_rectangle(code.rect())
        print_args = (barcode_name(code), code.payload(), (180 * code.rotation())
/ math.pi, code.quality(), fps)
        print("Barcode %s, Payload \"%s\", rotation %f (degrees), quality %d, FPS
%f" % print_args)
        img.draw_string(0, 0, "%2.1ffps" %(fps), color=(0, 60, 128), scale=2.0)
    lcd.display(img)

```



### 8.2.3、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，用摄像头拍摄条形码，则会将条形码框出来，并且在IDE底部的串行终端。



```
串行终端
Barcode I25, Payload "1234567890", rotation 0.000000 (degrees), quality 51
Barcode I25, Payload "1234567890", rotation 0.000000 (degrees), quality 46
Barcode I25, Payload "1234567890", rotation 0.000000 (degrees), quality 43
Barcode I25, Payload "1234567890", rotation 0.000000 (degrees), quality 50
Barcode I25, Payload "1234567890", rotation 0.000000 (degrees), quality 50
Barcode I25, Payload "1234567890", rotation 0.000000 (degrees), quality 43
Barcode I25, Payload "1234567890", rotation 0.000000 (degrees), quality 46
Barcode I25, Payload "1234567890", rotation 0.000000 (degrees), quality 45
Barcode I25, Payload "1234567890", rotation 0.000000 (degrees), quality 44
Barcode I25, Payload "1234567890", rotation 0.000000 (degrees), quality 46
Barcode I25, Payload "1234567890", rotation 0.000000 (degrees), quality 49
搜索结果 串行终端
```

## 8.2.4、实验总结

本实验可以识别多种类型的条形码，也可以自行到网上搜索条形码生成工具，生成条形码自定义信息的条形码，或者点击IDE的工具->机器视觉->条码生成器，可弹出浏览器页面搜索工具。

二维码图片：



1 2 3 4 5 6 7 8 9 0