

# 11、手写数字识别

## 11、手写数字识别

### 11.1、实验目标

### 11.2、实验前准备

### 11.3、实验过程

### 11.4、实验效果

### 11.5、实验总结

## 11.1、实验目标

本节课主要学习K210识别数字的功能，能识别手写的数字和打印的数字。

本次实验的参考代码路径为：K210\_Broad\05-AI\mnist.py

## 11.2、实验前准备

请先将模型文件导入内存卡上，再将内存卡插入到K210开发板的内存卡插槽上。具体操作步骤请参考：

[附录：导入模型文件到内存卡](#)

## 11.3、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，加载模型文件：/sd/KPU/mnist/uint8\_mnist\_cnn\_model.kmodel。

```
kpu = KPU()
kpu.load_kmodel("/sd/KPU/mnist/uint8_mnist_cnn_model.kmodel")
```

2. 新建while循环读取摄像头画面，然后复制一个112\*112大小的画面，对像素进行取反等处理，再将图像传入KPU里进行计算，与模型文件做运算，最终得到最优识别结果和识别分数。

```
while True:
    gc.collect()
    img = sensor.snapshot()
    img_mnist1=img.to_grayscale(1)
    img_mnist2=img_mnist1.resize(112,112)
    img_mnist2.invert()
    img_mnist2.strech_char(1)
    img_mnist2.pix_to_ai()

    out = kpu.run_with_output(img_mnist2, getlist=True)
    max_mnist = max(out)
    index_mnist = out.index(max_mnist)
    score = KPU.sigmoid(max_mnist)
```

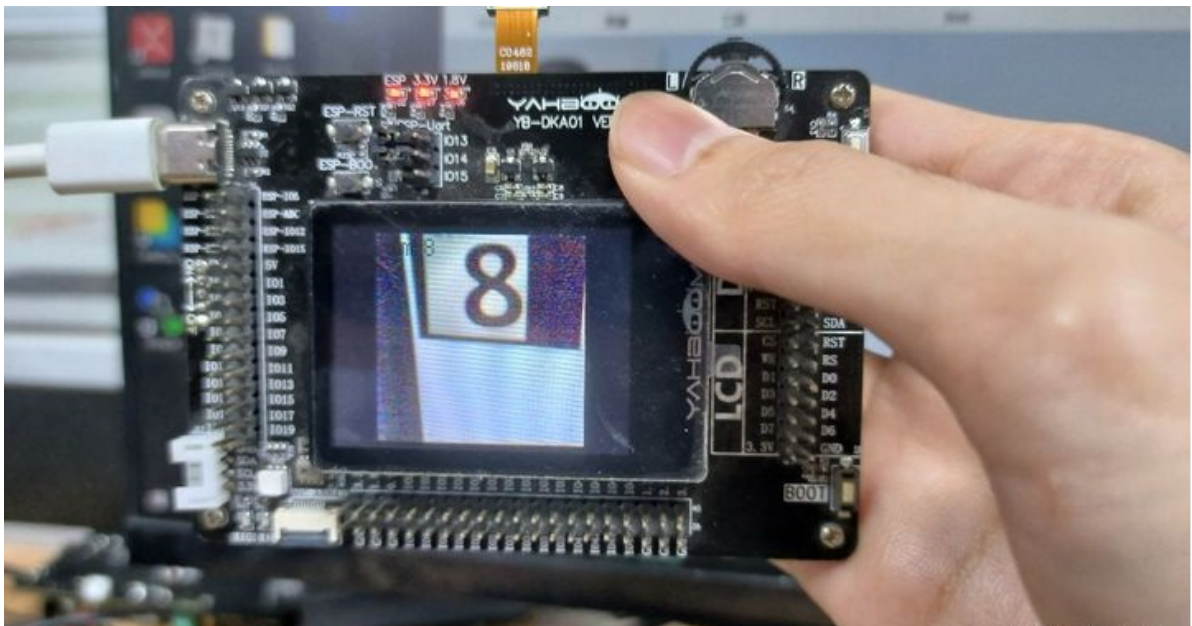
3. 由于识别的数字有可能会出现问题，请尽量在白色背景下识别。接下来进行过滤，根据实际测试出来的结果，背景全白的情况下，会误识别成数字1，背景全黑的情况下，会误识别成数字5。所以单独对识别到数字1和数字5进行额外分析，如果识别分数大于0.999才认为是对应的数字，反正就认为没有识别到数字。其中0.999可以根据实际识别效果做调整。最终将识别到的数字显示在屏幕上。

```
if index_mnist == 1:
    if score > 0.999:
        display_str = "num: %d" % index_mnist
        print(display_str, score)
        img.draw_string(4,3,display_str,color=(0,0,0),scale=2)
elif index_mnist == 5:
    if score > 0.999:
        display_str = "num: %d" % index_mnist
        print(display_str, score)
        img.draw_string(4,3,display_str,color=(0,0,0),scale=2)
else:
    display_str = "num: %d" % index_mnist
    print(display_str, score)
    img.draw_string(4,3,display_str,color=(0,0,0),scale=2)
lcd.display(img)
```

## 11.4、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，将摄像头拍摄自己手写的数字或打印的数字，屏幕左上角会显示当前识别到的数字。



## 11.5、实验总结

K210模块可以识别手写的数字和打印的数字，由于模型训练采用的是白底黑字，所以请使用白底黑字素材，如果背景太乱，或者数字太小，可能会出现误识别的情况。