11、handwritten digit recognition

11.1、Experimental goal

This lesson mainly learns the function of K210 to recognize numbers, which can recognize handwritten numbers and printed numbers.

The reference code path for this experiment is: K210_Broad\05-AI\mnist.py

11.2、Preparation before experiment

Please import the model file to the memory card first, and then insert the memory card into the memory card slot of the K210 development board. Please refer to the specific operation steps：

Appendix: Import model files to memory card

11.3、The process of the experiment

The factory firmware of the module has integrated the AI vision algorithm module. If you have downloaded other firmware, please burn it back to the factory firmware and then conduct experiments.

1. Import related libraries, initialize camera and LCD display, load model files：/sd/KPU/mnist/uint8_mnist_cnn_model.kmodel。

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

2. Create a while loop to read the camera image, then copy a 112*112 image, invert the pixels, etc., then transfer the image to the KPU for calculation, and perform calculations with the model file to finally obtain the optimal recognition result and recognition Fraction.

```
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)
```

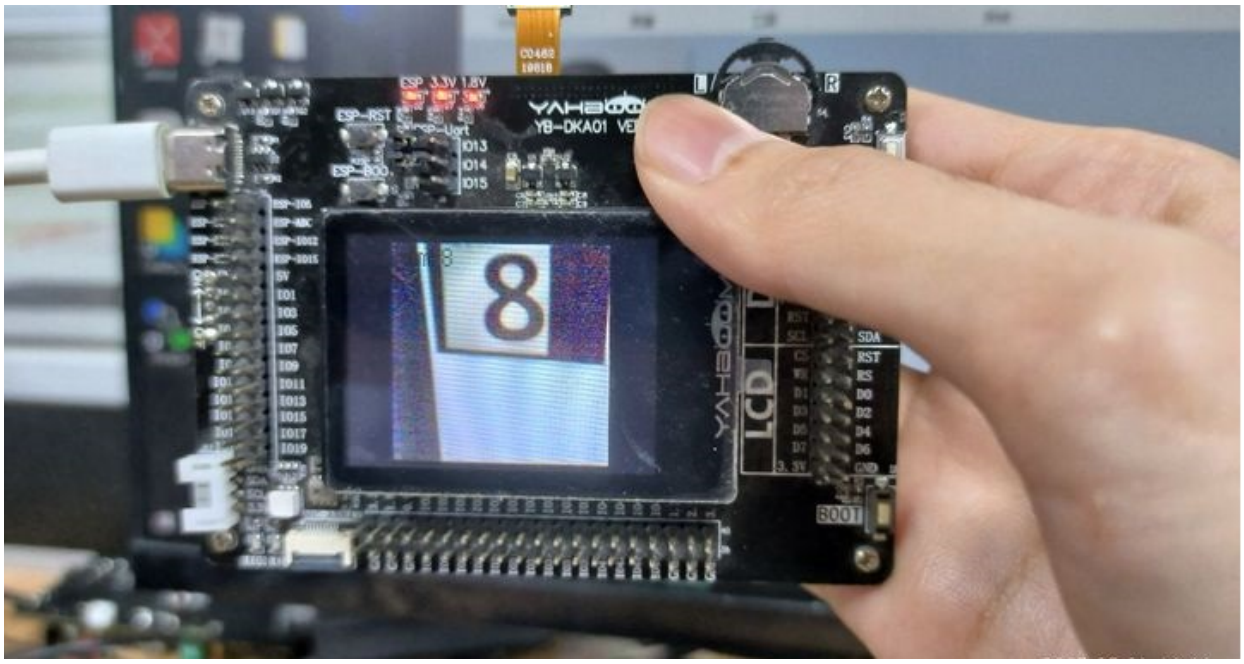
3. Since the recognized numbers may be misrecognized, please try to recognize them against a white background. Next, filter. According to the actual test results, if the background is completely white, it will be misrecognized as the number 1, and if the background is completely black, it will be mistakenly recognized as the number 5. Therefore, additional analysis is performed on the recognized numbers 1 and 5 separately. If the recognition score is greater than 0.999, it is considered to be the corresponding number. Anyway, it is considered that no number has been recognized. Among them, 0.999 can be adjusted according to the actual recognition effect. Finally, the recognized numbers are displayed on the screen.

```
    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、Experimental effect

Connect the K210 development board to the computer through the TYPE-C data cable, click the connect button in CanMV IDE, and click the run button after the connection is completed to run the routine code. You can also download the code as main.py to the K210 development board to run.

After the system initialization is completed, the LCD will display the camera screen, and the camera will take pictures of your handwritten or printed numbers, and the currently recognized number will be displayed in the upper left corner of the screen.



11.5、Experiment summary

The K210 module can recognize handwritten numbers and printed numbers. Since the model training uses black characters on a white background, please use black characters on a white background. If the background is too messy or the numbers are too small, misrecognition may occur.