11 Handwritten digit recognition

11.1 experimental goals

This lesson focuses on the K 210's function of recognizing numbers, and can recognize handwritten numbers and printed numbers.

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

11.2 Preparation before the experiment

Please first import the model file into the memory card, and then insert the memory card into the memory card slot of the K210 module.

11.3 experimental procedure

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 before doing the experiment.

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

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

2. Create a new while loop to read the camera picture, then copy a 112112-sized picture, invert the pixels, and then transfer the image to the KPU for calculation, and calculate with the model file, and finally obtain the optimal recognition result and recognition score.

```
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 identified numbers may be misidentified, please try to identify them against a white background. Next, filtering, according to the actual test results, if the background is all white, it will be misidentified as the number 1, and if the background is completely black, it will be misidentified as the number 5. Therefore, the number 1 and number 5 are recognized separately, and if the recognition score is greater than 0.999, it is considered to be the corresponding number, anyway, it is considered that the number is not recognized. Among them, 0.999 can be adjusted according to the actual recognition effect. The recognized number is finally 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)
```
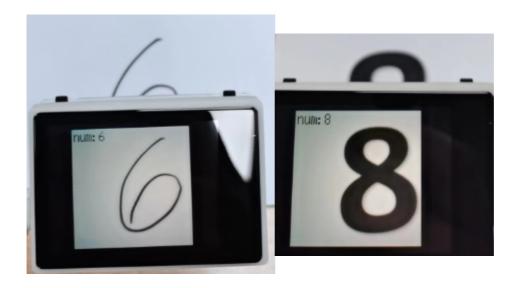
11.4 experimental results

Connect the K210 module to the computer through the microUSB data cable, CanMV IDE click the connect button, after the connection is completed click the Run button to run the routine code. You can also download the code as main.py and run it in the K210 module.

Wait for the system initialization to complete, the LCD displays the camera screen, the camera will take its own handwritten or printed numbers, and the upper left corner of the screen will display the currently recognized number.

11.5 experiment summary

The K 210 module can recognize handwritten numbers and printed numbers, because the model training uses black characters on a white background, so please use black characters on a white background, if the background is too messy, or the numbers are too small, there may be misidentification.