

7 Mask detection

7 Mask detection

7.1 experimental goals

7.2 Preparation before the experiment

7.3 experimental procedure

7.4 experimental results

7.5 experiment summary

7.1 experimental goals

This lesson mainly learns the mask detection function, analyzes the picture collected by the camera, compares the model, analyzes whether to wear a mask, and prints out the status of wearing a mask.

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

7.2 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.

7.3 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 libraries and initialize the camera and LCD display.

```
import sensor, image, time, lcd
from maix import KPU

lcd.init()
sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.skip_frames(time = 1000)
clock = time.clock()
```

2. To initialize the parameters related to KPU, kpu needs to load the kmodel file. The path of the model file required for this experiment is:/sd/KPU/face_mask_detect/detect_5.kmodel, And use yolo2 to calculate conformity to the model requirements. od_img is the image of the neural network, the size is 320*256, which is used to store the camera image and pass it to KPU for calculation.

```

od_img = image.Image(size=(320,256), copy_to_fb=False)

anchor = (0.156250, 0.222548, 0.361328, 0.489583, 0.781250, 0.983133, 1.621094,
1.964286, 3.574219, 3.94000)
kpu = KPU()
print("ready load model")
kpu.load_kmodel("/sd/KPU/face_mask_detect/detect_5.kmodel")
kpu.init_yolo2(anchor, anchor_num=5, img_w=320, img_h=240, net_w=320 , net_h=256
,layer_w=10 ,layer_h=8, threshold=0.7, nms_value=0.4, classes=2)

```

3. Create a new while loop, pass the image to the KPU for calculation, use the yolo2 neural network algorithm to solve, if it is a face wearing a mask buttoned, it is framed in green and displayed "with mask", if it is a face without a mask, it is framed in red and displayed "without mask".

```

while True:
    clock.tick()
    img = sensor.snapshot()
    od_img.draw_image(img, 0,0)
    od_img.pix_to_ai()
    kpu.run_with_output(od_img)
    dect = kpu.regionlayer_yolo2()
    fps = clock.fps()
    if len(dect) > 0:
        print("dect:", dect)
        for l in dect :
            if l[4] :
                img.draw_rectangle(l[0],l[1],l[2],l[3], color=(0, 255, 0))
                img.draw_string(l[0],l[1]-24, "with mask", color=(0, 255, 0),
scale=2)
            else:
                img.draw_rectangle(l[0],l[1],l[2],l[3], color=(255, 0, 0))
                img.draw_string(l[0],l[1]-24, "without mask", color=(255, 0, 0),
scale=2)

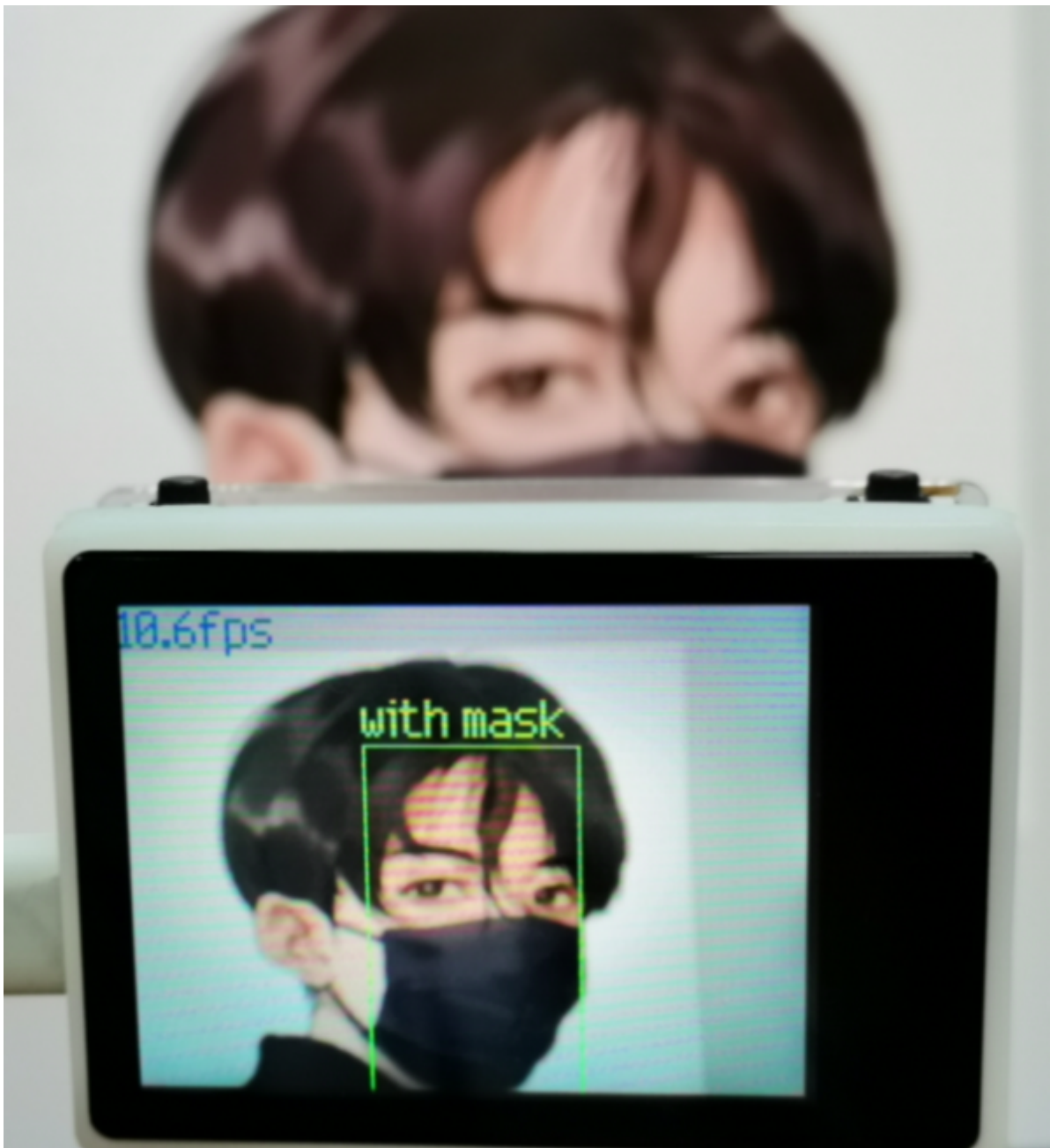
        img.draw_string(0, 0, "%2.1ffps" %(fps), color=(0, 60, 128), scale=2.0)
    lcd.display(img)

```

7.4 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, shoots the face with the camera, displays a green box and "with mask" when there is a mask, and displays a red box and "without mask" when no mask is worn. At the same time, the serial terminal at the bottom of the IDE will print information about the face.





```
dect: [[121, 76, 73, 88, 0, 0.9914771]]
dect: [[121, 77, 73, 88, 0, 0.9915406]]
dect: [[121, 72, 72, 88, 0, 0.9890528]]
dect: [[121, 77, 73, 88, 0, 0.9930605]]
dect: [[121, 77, 73, 88, 0, 0.986703]]
dect: [[121, 76, 73, 88, 0, 0.9868907]]
dect: [[121, 72, 73, 88, 0, 0.9872982]]
dect: [[121, 77, 73, 88, 0, 0.9894311]]
dect: [[121, 76, 73, 88, 0, 0.9894311]]
dect: [[109, 61, 82, 90, 0, 0.974972]]
```

7.5 experiment summary

The memory card required for mask detection needs to load the model file, so you need to import the model file into the memory card in advance, and then insert the memory card into the memory card slot of the K 210 module, if the model file in the memory card cannot be read, an error will be reported.

At present, the threshold of mask detection is threshold=0.7, and if you need to detect faces more accurately, you can adjust the threshold appropriately.

