8.6、facial feature detection

8.6.1、Experimental goal

This lesson mainly learns the function of face detection, analyzes the screen captured by the camera, compares the model, and frames the face at the same time, expresses the feature points of the face by points, and prints the relevant information.

The reference code path for this experiment is：K210_Broad\05-AI\face_detect_68lm.py

8.6.2、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. For specific steps, please refer to:

Appendix: Import model files to memory card

8.6.3、experiment 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 and then conduct experiments.

1. Import related libraries, and initialize 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 = 100)
clock = time.clock()
```

2. Initialize KPU related parameters, kpu needs to load the kmodel file, the model file path required for this experiment is: /sd/KPU/yolo_face_detect/face_detect_320x240.kmodel, and use yolo2 to calculate whether it meets the model requirements.

```
anchor = (0.1075, 0.126875, 0.126875, 0.175, 0.1465625, 0.2246875, 0.1953125,
0.25375, 0.2440625, 0.351875, 0.341875, 0.4721875, 0.5078125, 0.6696875, 0.8984375,
1.099687, 2.129062, 2.425937)
kpu = KPU()
kpu.load_kmodel("/sd/KPU/yolo_face_detect/face_detect_320x240.kmodel")
kpu.init_yolo2(anchor, anchor_num=9, img_w=320, img_h=240, net_w=320 , net_h=240
,layer_w=10 ,layer_h=8, threshold=0.7, nms_value=0.2, classes=1)
```

3. Initialize the face feature point KPU model, the model file path is: /sd/KPU/face_detect_with_68landmark/landmark68.kmodel.

```
lm68_kpu = KPU()
print("ready load model")
lm68_kpu.load_kmodel("/sd/KPU/face_detect_with_68landmark/landmark68.kmodel")
```

4. Extract information about detected faces.

```
def extend_box(x, y, w, h, scale):
    x1_t = x - scale*w
    x2_t = x + w + scale*w
    y1_t = y - scale*h
    y2_t = y + h + scale*h
    x1 = int(x1_t) if x1_t>1 else 1
    x2 = int(x2_t) if x2_t<320 else 319
    y1 = int(y1_t) if y1_t>1 else 1
    y2 = int(y2_t) if y2_t<240 else 239
    cut_img_w = x2-x1+1
    cut_img_h = y2-y1+1
    return x1, y1, cut_img_w, cut_img_h
```

5. Create a new while loop, pass the image into the KPU for calculation, and use the yolo2 neural network algorithm for calculation. First, you need to detect the face, then extract the position information detected by the face, and then pass it into the KPU for calculation and extract facial feature points , and circle the facial organs with symbols.

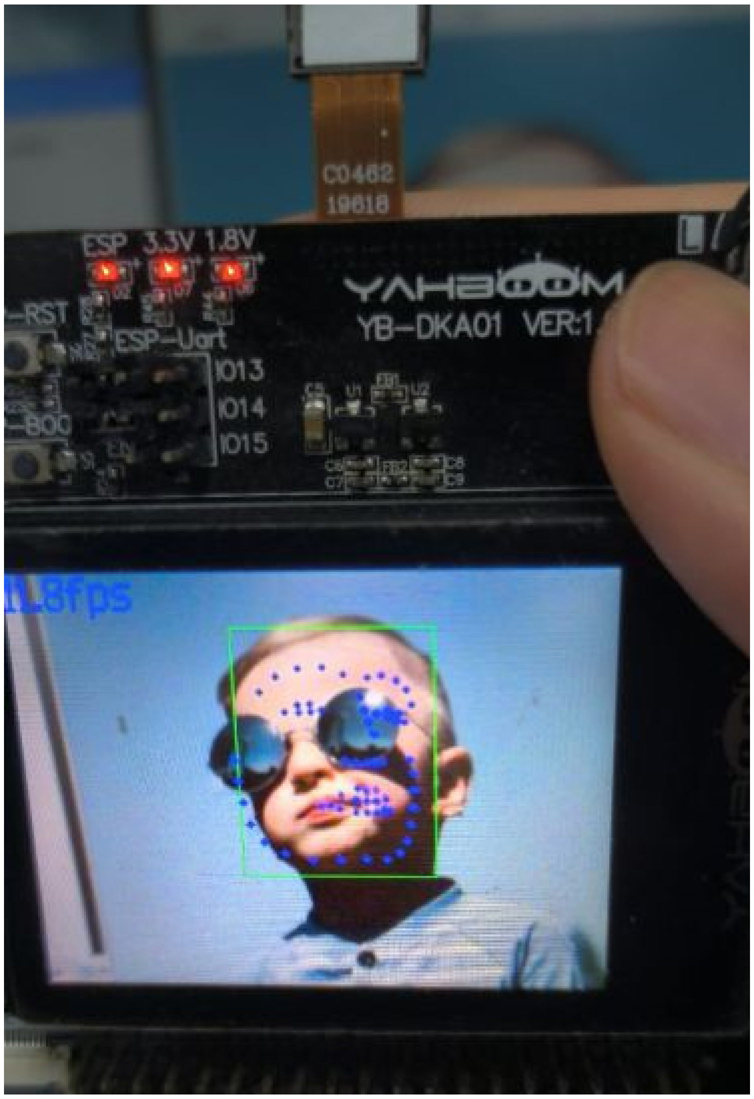
```
while True:
    clock.tick()
    img = sensor.snapshot()
    kpu.run_with_output(img)
    dect = kpu.regionlayer_yolo2()
    fps = clock.fps()
    if len(dect) > 0:
        print("dect:",dect)
        for l in dect :
            x1, y1, cut_img_w, cut_img_h = extend_box(l[0], l[1], l[2], l[3],
scale=0.08)
            face_cut = img.cut(x1, y1, cut_img_w, cut_img_h)
            a = img.draw_rectangle(l[0],l[1],l[2],l[3], color=(0, 255, 0))
            face_cut_128 = face_cut.resize(128, 128)
            face_cut_128.pix_to_ai()
            out = lm68_kpu.run_with_output(face_cut_128, getlist=True)
            #print("out:",len(out))
            for j in range(68):
                x = int(KPU.sigmoid(out[2 * j])*cut_img_w + x1)
                y = int(KPU.sigmoid(out[2 * j + 1])*cut_img_h + y1)
                #a = img.draw_cross(x, y, size=1, color=(0, 0, 255))
                a = img.draw_circle(x, y, 2, color=(0, 0, 255), fill=True)
            del (face_cut_128)
            del (face_cut)

    a = img.draw_string(0, 0, "%2.1ffps" %(fps), color=(0, 60, 255), scale=2.0)
    lcd.display(img)
```

8.6.4、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 image, and use the camera to capture the face. When the face is detected, a green frame will appear on the screen to frame the face. When the face is framed, the outline of the facial organs will be used Circle the symbol and print the relevant data on the IDE backplane.



8.6.5、Experiment summary

Face detection needs to use the memory card 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 K210 development board. If the model file in the memory card cannot be read, then Will report an error.

The current threshold for face detection is threshold=0.7. If you need to detect faces more accurately, you can adjust the threshold appropriately.

The premise of facial organ detection is to first detect the position of the face, and then transfer the position image information of the face to the KPU for another calculation to obtain the position of the facial organ features, and finally draw the face with symbols Organ silhouettes.