6 Face feature detection

6 Face feature detection
- 6.1 experimental goals
- 6.2 Preparation before the experiment
- 6.3 experimental procedure
- 6.4 experimental results
- 6.5 experiment summary

6.1 experimental goals

This lesson mainly learns the face detection function, analyzes the picture collected by the camera, compares the model, frames the face, represents the feature points of the face by points, and prints the relevant information.

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

6.2 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.

6.3 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 = 100)
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/yolo_face_detect/face_detect_320x240.kmodel，并使用yolo2来计算是否符合模型要求。

```
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 landmark KPU model, and 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 to the KPU for calculation, and use the yolo 2 neural network algorithm to solve, first need to detect the face, then extract the position information detected by the face, and then pass it to the KPU to calculate and extract the 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):
```

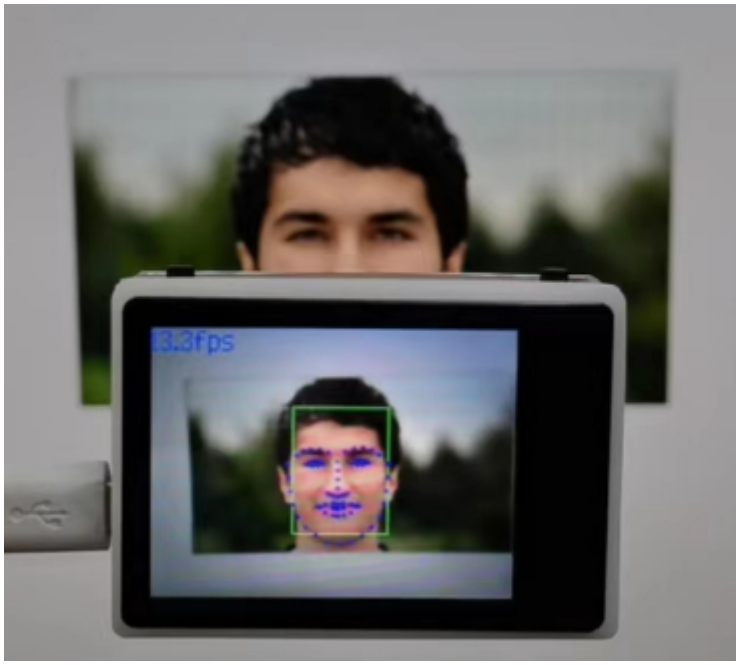
```
                x = int(KPU.sigmoid(out[2 * j])*cut_img_w + x1)
                y = int(KPU.sigmoid(out[2 * j + 1])*cut_img_h + y1)
                #a = img.draw_cross(x, y, size=1, color=(0, 0, 255))
                a = img.draw_circle(x, y, 2, color=(0, 0, 255), fill=True)
            del (face_cut_128)
            del (face_cut)

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

6.4 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 is completed, LCD display camera screen, use the camera to shoot the face, when the face is detected, the screen will appear green frame to frame the face, while framing the face, the outline of the face organ with symbols circled, print the relevant data on the IDE baseplate.



6.5 experiment summary

Face detection requires a 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 K 210 module, if the model file in the memory card cannot be read, an error will be reported.

At present, the threshold value for detecting faces is threshold=0.7, and if you need to detect faces more accurately, you can adjust the threshold appropriately.

The premise of facial organ detection is that the position of the face needs to be detected first, and then the position image information of the face is transferred to the KPU for another calculation, so as to obtain the position of the facial organ features, and finally the outline of the facial organ can be drawn with symbols.