

## 8.5、 Face Detection

---

### 8.5、 Face Detection

#### 8.5.1、 Experimental goal

#### 8.5.2、 Preparation before experiment

#### 8.5.3、 experiment procedure

#### 8.5.4、 Experimental effect

#### 8.5.5、 Experiment summary

### 8.5.1、 Experimental goal

This lesson mainly learns the function of face detection, analyzes the pictures captured by the camera, compares them with models, frames them if there is a face, and prints the relevant information.

The reference code path for this experiment is: K210\_Broad\05-AI\yolo\_face\_detect.py

### 8.5.2、 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](#)

### 8.5.3、 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/yolo\_face\_detect.kmodel, and use yolo2 to calculate whether it meets the model requirements. od\_img is the image of the neural network, with a size of 320\*256, which is used for subsequent storage of the camera image and

passed to the KPU for calculation.

```
od_img = image.Image(size=(320,256))

anchor = (0.893, 1.463, 0.245, 0.389, 1.55, 2.58, 0.375, 0.594, 3.099, 5.038, 0.057,
0.090, 0.567, 0.904, 0.101, 0.160, 0.159, 0.255)
kpu = KPU()
kpu.load_kmodel("/sd/KPU/yolo_face_detect/yolo_face_detect.kmodel")
kpu.init_yolo2(anchor, anchor_num=9, img_w=320, img_h=240, net_w=320 , net_h=256
,layer_w=10 ,layer_h=8, threshold=0.7, nms_value=0.3, classes=1)
```

3. Create a new while loop, pass the image to the KPU for calculation, use the yolo2 neural network algorithm to solve, and finally get the position information of the face, and then frame the face.

```
while True:
    clock.tick()
    img = sensor.snapshot()
    a = 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 :
            a = img.draw_rectangle(l[0],l[1],l[2],l[3], color=(0, 255, 0))
        a = img.draw_string(0, 0, "%2.1ffps" %(fps), color=(0, 60, 128), scale=2.0)
    lcd.display(img)

kpu.deinit()
```

## 8.5.4、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 use the camera to take pictures of faces. When a face is detected, a green frame will appear on the screen to frame the face, and the detected face will be printed on the serial terminal at the bottom of the IDE. information.



```
串行终端 | 📄 📁
dect: [[142, 12, 116, 119, 0, 0.9562205]]
dect: [[146, 6, 116, 128, 0, 0.9562205]]
dect: [[148, 9, 107, 118, 0, 0.9562205]]
dect: [[145, 12, 107, 118, 0, 0.9218]]
dect: [[140, 8, 116, 110, 0, 0.9562205]]
dect: [[149, 9, 107, 110, 0, 0.9562205]]
dect: [[144, 8, 116, 110, 0, 0.8964768]]
dect: [[154, 0, 125, 93, 0, 0.9562205]]
dect: [[151, 0, 125, 99, 0, 0.9454496]]
```

### 8.5.5、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.