

8.9、 object detection

8.9、 object detection

8.9.1、 Experimental goal

8.9.2、 Preparation before experiment

8.9.3、 The process of the experiment

8.9.4、 Experimental effect

8.9.5、 Experiment summary

8.9.1、 Experimental goal

This lesson mainly learns the object detection function, and analyzes the images captured by the camera. If it matches the object type in the model, it will frame and display the object type name.

The reference code path for this experiment is: K210_Broad\05-AI\voc20_object_detect.py

8.9.2、 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](#)

8.9.3、 The process of the experiment

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. Create a new object type name variable `obj_name`, and the objects represented in turn are: 0. Aeroplane, 1. Bicycle, 2. Bird, 3. Boat, 4. Bottle, 5. Bus, 6. Car, 7. Cat, 8. chair, 9. cow, 10. dining table, 11. dog, 12. horse, 13. motorcycle, 14. person, 15. potted plant, 16. sheep, 17. sofa, 18. train, 19. tvmonitor.

```
obj_name = ("aeroplane", "bicycle", "bird", "boat", "bottle", "bus", "car", "cat",
"chair", "cow", "diningtable", "dog", "horse", "motorbike", "person", "pottedplant",
"sheep", "sofa", "train", "tvmonitor")
```

3. Initialize KPU related parameters, kpu needs to load the kmodel file, the model file path required for this experiment is: /sd/KPU/voc20_object_detect/voc20_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 to store the camera image later and pass it to the KPU for calculation

```
od_img = image.Image(size=(320,256))
anchor = (1.3221, 1.73145, 3.19275, 4.00944, 5.05587, 8.09892, 9.47112, 4.84053,
11.2364, 10.0071)
kpu = KPU()
print("ready load model")
kpu.load_kmodel("/sd/KPU/voc20_object_detect/voc20_detect.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=0.2, classes=20)
```

4. Create a new while loop, pass the image to the KPU for calculation, use the yolo2 neural network algorithm for calculation, and finally get the type name with the highest similarity, frame the object and display the object name.

```
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 :
            img.draw_rectangle(l[0],l[1],l[2],l[3], color=(0, 255, 0))
            img.draw_string(l[0],l[1], obj_name[l[4]], color=(0, 255, 0), scale=1.5)

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

8.9.4、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, use the camera to shoot the object, the screen will display the name of the detected object, and the serial terminal at the bottom of the IDE will print the relevant information of the detected object.



串行终端

```
dect: [[108, 27, 112, 141, 11, 0.708204]]
dect: [[108, 28, 112, 141, 11, 0.7796531]]
dect: [[108, 29, 112, 141, 11, 0.7831535]]
dect: [[108, 22, 112, 154, 11, 0.7235578]]
dect: [[103, 22, 102, 154, 11, 0.7130483]]
dect: [[108, 27, 112, 141, 11, 0.8131722]]
dect: [[112, 27, 102, 141, 11, 0.8284641]]
dect: [[107, 24, 112, 154, 11, 0.7739974]]
```

8.9.5、 Experiment summary

The object 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, it will error.

At present, the threshold for detecting objects is threshold=0.7. If you need to detect more accurately, you can adjust the threshold appropriately.

Detecting objects can only detect objects in the variable `obj_name`, other objects cannot be detected temporarily.