8.9、物体检测

8.9.1、实验目标

本节课主要学习物体检测功能，将摄像头采集的画面进行分析出来，如果符合模型中的物体类型，则框出并显示物体类型名称。

本次实验的参考代码路径为：K210_Broad\05-AI\voc20_object_detect.py

8.9.2、实验前准备

请先将模型文件导入内存卡上，再将内存卡插入到K210开发板的内存卡插槽上。具体操作步骤请参考：

附录：导入模型文件到内存卡

8.9.3、实验过程

模块的出厂固件已经集成AI视觉算法模块，如果下载过其他固件，请烧录回出厂固件再进行实验。

1. 导入相关库，并初始化摄像头和LCD显示屏。

```
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. 新建物体类型名称变量`obj_name`，依次代表的物体为：0.飞机、1.自行车、2.鸟、3.船、4.瓶子、5.公交车、6.小汽车、7.猫、8.椅子、9.奶牛、10.餐桌、11.狗、12.马、13.摩托车、14.人、15.盆栽、16.绵羊、17.沙发、18.火车、19.显示器。

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

3. 初始化KPU相关的参数，kpu需要加载kmodel文件，本次实验需要的模型文件路径为：/sd/KPU/voc20_object_detect/voc20_detect.kmodel，并使用yolo2来计算是否符合模型要

求。od_img为神经网络的图像，尺寸为320*256，用于后续储存摄像头图像并传入KPU计算。

```
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. 新建while循环，将图像传入KPU进行计算，使用yolo2神经网络算法进行解算，最终得到相似度最高的类型名称，框出物体并显示物体名称。

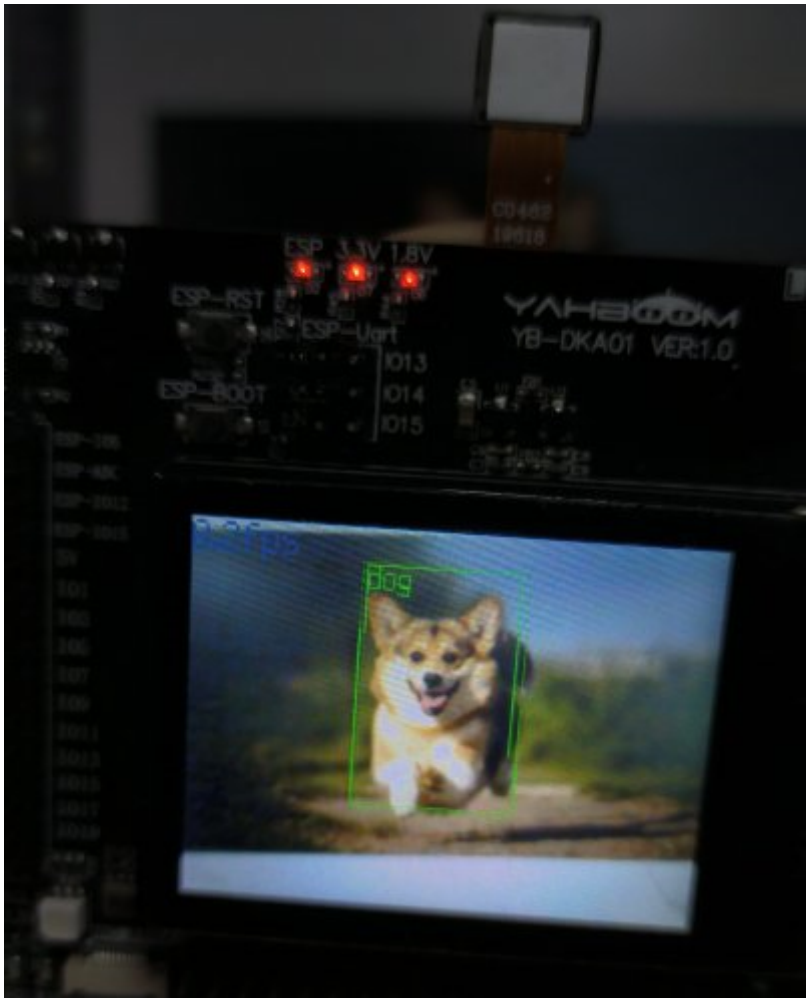
```
while True:
    clock.tick()
    img = sensor.snapshot()
    od_img.draw_image(img, 0,0)
    od_img.pix_to_ai()
    kpu.run_with_output(od_img)
    dect = kpu.regionlayer_yolo2()
    fps = clock.fps()
    if len(dect) > 0:
        print("dect:",dect)
        for l in dect :
            img.draw_rectangle(l[0],l[1],l[2],l[3], 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、实验效果

将K210开发板通过TYPE-C数据线连接到电脑上，CanMV IDE点击连接按钮，连接完成后点击运行按钮，运行例程代码。也可以将代码作为main.py下载到K210开发板上运行。

等待系统初始化完成后，LCD显示摄像头画面，用摄像头拍摄物体，屏幕会显示检测到的物体名称，同时IDE底部的串行终端会打印检测到的物体的相关信息。



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、实验总结

物体检测需要用的内存卡加载模型文件，所以需要提前将模型文件导入内存卡，再将内存卡插入K210开发板的内存卡卡槽里，如果无法读取到内存卡里的模型文件，则会报错。

目前检测物体的阈值为threshold=0.7，如果需要检测更加准确，可以适当调整阈值。

检测物体只能检测变量`obj_name`中的物体，其他物体暂时无法检测。