

## 5.7 物体检测

一、实验目的(以 OV2640 为例, OV9655/GC2145 思路一样)

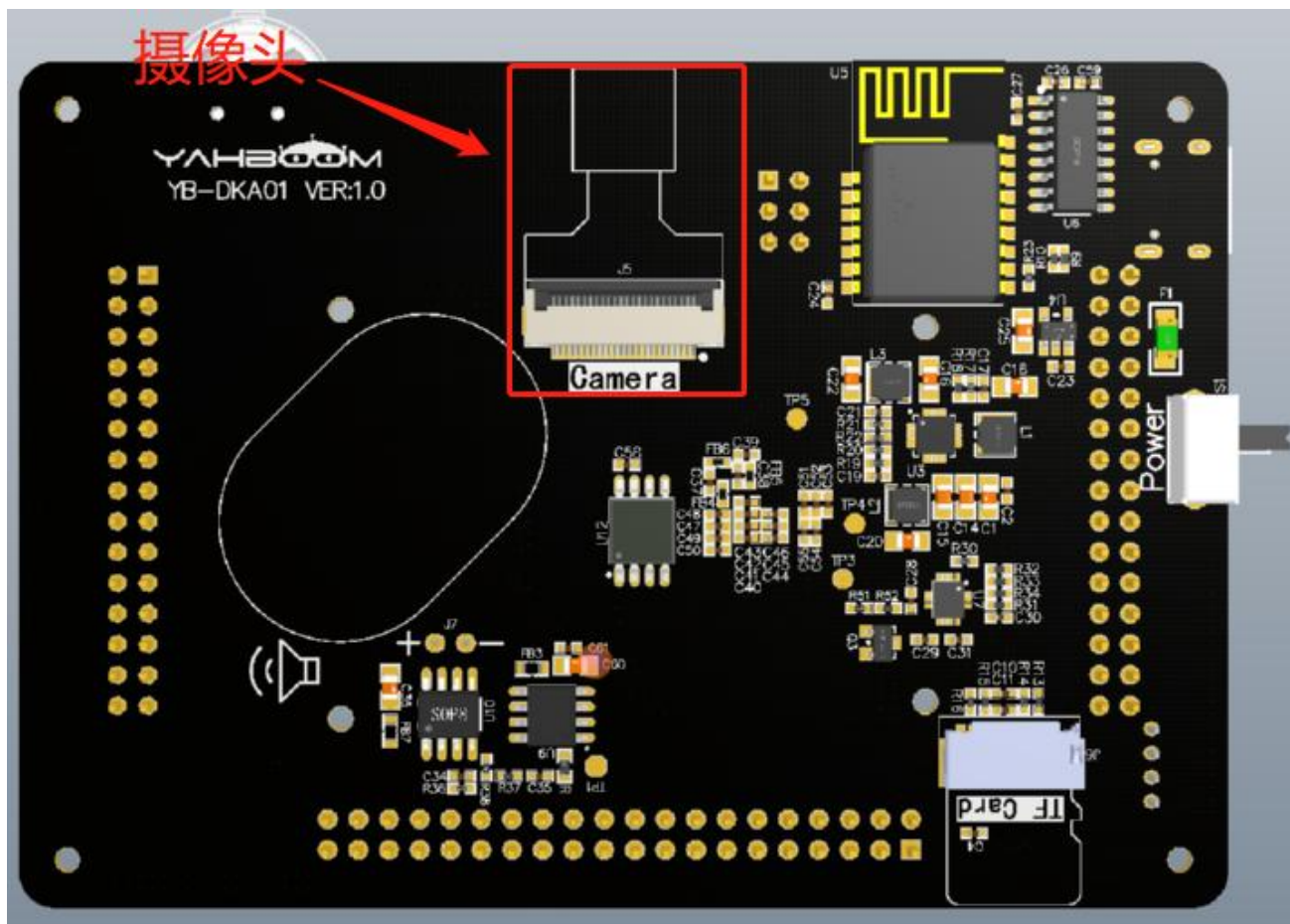
本节课主要学习 K210 如何物体检测, 然后通过 LCD 显示屏实时框出检测物体然后以不同颜色标记名称。

二、实验准备

### 1. 实验元件

OV2640 摄像头/OV9655 摄像头/GC2145 摄像头

LCD 显示屏



### 2. 硬件连接

K210 开发板出厂默认已经安装好摄像头和显示器, 只需要使用 Type-C 数

据线连接 K210 开发板与电脑即可。

### 三、实验原理

Kendryte K210 具备机器视觉能力，是零门槛机器视觉嵌入式解决方案。它可以在低功耗情况下进行卷积神经网络计算。相关介绍请看前面所述。

物体检测是计算机视觉中的经典问题之一，其任务是用框去标出图像中物体的位置，并给出物体的类别。从传统的人工设计特征加浅层分类器的框架，到基于深度学习的端到端的检测框架，物体检测一步步变得愈加成熟。具体训练过程这里省略掉，下面只介绍 K210 如何对训练好的模型进行部署并输出识别结果。

#### K210 实时物体检测方案步骤：

- 1、 训练模型
- 2、 模型转换成 kmodel
- 3、 K210 加载模型
- 4、 K210 获取视频图像
- 5、 图像转换成模型需要的尺寸大小
- 6、 运行模型获取 KPU 处理结果
- 7、 获取输出层的结果
- 8、 在图片中圈出识别结果显示

## 四、实验步骤

### 1、代码流程

#### 系统内部初始化部分：

- 系统时钟初始化
- 串口初始化
- 硬件引脚初始化
- I0 电压设置
- 系统中断初始化
- Flash 初始化

#### 外部硬件初始化

- Lcd 初始化
- 0v2640 初始化

#### 物体检测初始化

- 模型加载
- 物体检测层配置初始化

#### 人脸检测业务逻辑层

- 等待摄像头采集完成
- 传入摄像头采集的图像到 KPU 运行模型
- 等待 KPU 处理完成
- 获取 KPU 最终处理的结果
- 把 KPU 处理的结果带入区域层计算最终识别位置和结果
- 根据识别到的结果逐一标记

## 2、核心代码如下：

```
int main(void)
{
    sysclock_init();    /* 系统时钟初始化*/
    uarths_init();      /* 串口初始化*/
    hardware_init();    /* 硬件引脚初始化*/
    io_set_power();     /* 设置 IO 口电压*/
    plic_init();        /* 系统中断初始化 */
    lable_init();       /* 层初始化*/

    /* flash init */
    printf("flash init\n");
    w25qxx_init(3, 0);
    w25qxx_enable_quad_mode();
#ifdef LOAD_KMODEL_FROM_FLASH
    model_data = (uint8_t*)malloc(KMODEL_SIZE + 255);
    uint8_t *model_data_align = (uint8_t*)((uintptr_t)model_data+255)&(~255));
    w25qxx_read_data(0xA00000, model_data_align, KMODEL_SIZE, W25QXX_QUAD_FAST);
#else
    uint8_t *model_data_align = model_data;
#endif

    // 初始化 LCD
    lcd_init();
    lcd_draw_picture_half(0, 0, 320, 240, logo);
    lcd_draw_string(100, 40, "Hello Yahboom!", RED);
    lcd_draw_string(100, 60, "object detection demo!", BLUE);
    sleep(1);
    /* 初始化摄像头*/
    ov2640_init();

    /* 初始化物体检测模型 */
    if (kpu_load_kmodel(&obj_detect_task, model_data_align) != 0)
    {
        printf("\nmodel init error\n");
        while (1);
    }
    obj_detect_r1.anchor_number = ANCHOR_NUM;
    obj_detect_r1.anchor = anchor;
    obj_detect_r1.threshold = 0.5;
    obj_detect_r1.nms_value = 0.2;
```

```

region_layer_init(&obj_detect_rl, 10, 8, 125, 320, 240);
/* enable global interrupt */
sysctl_enable_irq();
/* system start */
printf("System start\n");

while (1)
{
    g_dvp_finish_flag = 0;

    while (!g_dvp_finish_flag)
        ;

    /* run obj detect */
    memset(g_ai_od_buf, 127, 320*256*3);
    for (uint32_t cc = 0; cc < 3; cc++)
    {
        memcpy(g_ai_od_buf + 320 * (cc * 256 + (256 - 240) / 2), g_ai_buf_in + c
c * 320 * 240, 320 * 240);
    }

    /*运行模型*/
    g_ai_done_flag = 0;
    kpu_run_kmodel(&obj_detect_task, g_ai_od_buf, DMAC_CHANNEL5, ai_done, NULL);
    while(!g_ai_done_flag);

    /*获取 KPU 处理结果*/
    float *output;
    size_t output_size;
    kpu_get_output(&obj_detect_task, 0, (uint8_t **)&output, &output_size);

    /*获取输出层的结果*/
    obj_detect_rl.input = output;
    region_layer_run(&obj_detect_rl, &obj_detect_info);

    /* 显示视频图像*/
    lcd_draw_picture(0, 0, 320, 240, (uint32_t *)display_buf_addr);

    /* 画识别结果 */
    region_layer_draw_boxes(&obj_detect_rl, drawboxes);
}

```

```
return 0;
}
```

### 3、代码编译方法：

把本课程资料中的 object\_detection 文件夹复制到 SDK 中的 src 目录下，然后进入 build 目录，删除 build 目录下所有文件，最后运行以下命令编译。

```
cmake .. -DPROJ=object_detection -G "MinGW Makefiles"
```

```
make
```

编译完成后，在 build 文件夹下会生成 object\_detection.bin 文件，

### 4、代码烧录方法

打开 kflash 将 object\_detection.bin 文件烧录到 K210 开发板上。

```
MINGW64:/i/D/K210/kendryte-sdk/standalone-sdk/kendryte-standalone-sdk...
Administrator@Admin-PC MINGW64 /i/D/K210/kendryte-sdk/standalone-sdk/kendryte-standalone-sdk/build (develop)
$ make
[ 16%] Built target nncase
[ 83%] Built target kendryte
[100%] Built target object_detection
Administrator@Admin-PC MINGW64 /i/D/K210/kendryte-sdk/standalone-sdk/kendryte-standalone-sdk/build (develop)
$
```

扩展部分：

如果想把代码和模型文件分开烧录，需要修改代码

```
#define LOAD_KMODEL_FROM_FLASH 0
```

为

```
#define LOAD_KMODEL_FROM_FLASH 1
```

然后重新生成 bin 文件，这个时候我们需要把模型文件和 bin 文件打包成一个 kfpkg 文件在烧录。具体方法请参考人脸检测章节进行修改。

#### 四、实验现象

LCD 显示器先显示图片 logo 和文字，一秒后打开摄像头采集的画面，并且实时检测 20 种物体并标记位置和显示识别的结果。

#### 五、实验总结

1. 物体检测与人脸检测大部分是相同的。
2. 物体检测可以检测多种物体。