

8.6、人脸特征检测

8.6、人脸特征检测

8.6.1、实验目标

8.6.2、实验前准备

8.6.3、实验过程

8.6.4、实验效果

8.6.5、实验总结

8.6.1、实验目标

本节课主要学习人脸检测功能，将摄像头采集的画面分析，比对模型，在框出人脸的同时，将人脸的特征点也按点表示，并打印相关信息。

本次实验的参考代码路径为：K210_Broad\05-AI\face_detect_68lm.py

8.6.2、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](#)

8.6.3、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. 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/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. 初始化人脸特征点KPU模型，模型文件路径

为：/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. 提取检测到的人脸的信息。

```

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. 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，首先需要先检测人脸，再将人脸检测到的位置信息提取出来，再传入KPU计算并提取面部特征点，并将脸部器官用符号圈出来。

```

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))

```



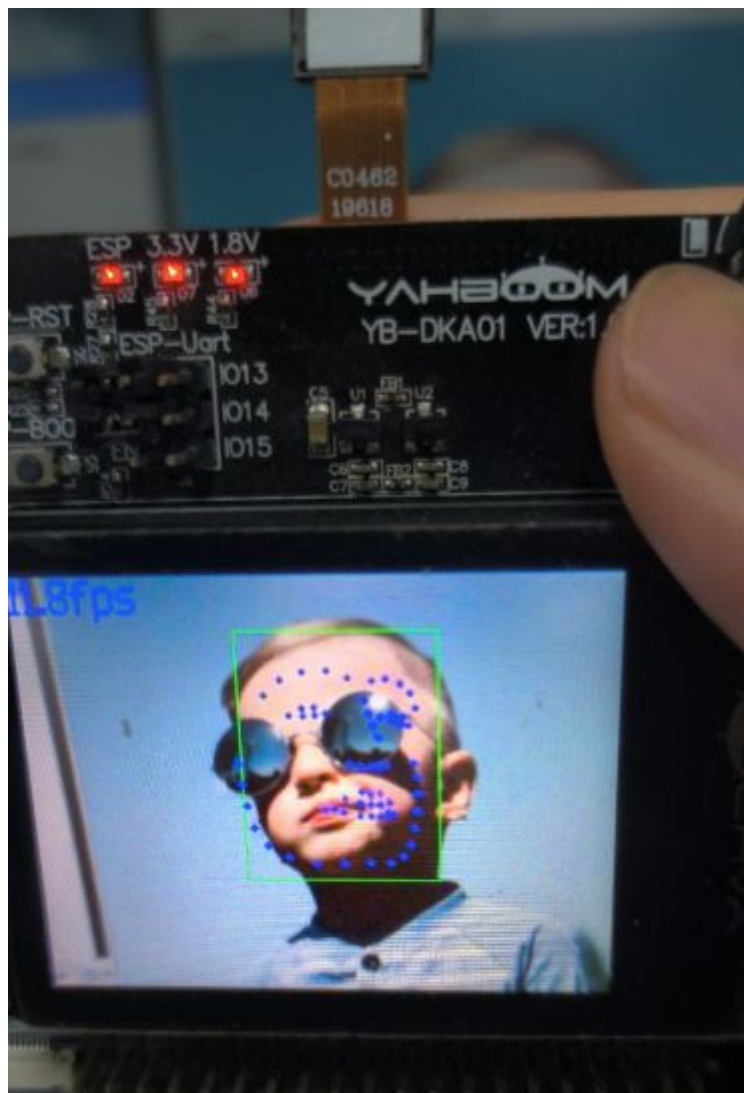
```
a = img.draw_circle(x, y, 2, color=(0, 0, 255), fill=True)
del (face_cut_128)
del (face_cut)

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

8.6.4、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，用摄像头拍摄人脸，当检测到人脸后，屏幕会出现绿框把人脸框出来，在框出人脸的同时，将脸部器官的轮廓用符号圈出来，在IDE底板打印相关数据。



8.6.5、实验总结

人脸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。

目前检测人脸的阈值为threshold=0.7，如果需要检测人脸更加准确，可以适当调整阈值。

脸部器官检测的前提是先需要检测出人脸的位置，然后再将人脸的位置图像信息传入KPU再进行一次运算，才能得到脸部器官特征的位置，最后才能用符号绘制出脸部器官的轮廓。