

Yolov5 分类器后端使用手册

1. 修改分类类别

我们修改 `nanoka.py` 文件里，这一个部分就可以，如图 1 所示。

```
# -----  
# Model parameters  
imgsz = (640, 640)  
stat = ['apple', 'orange']  
# Cache  
yolo_cache = np.zeros([640, 640, 3])  
# YOLOv5 model headless
```

图 1 识别类型统计

2. 修改摄像头

我们还是修改 `nanoka.py` 的另一处，在你自己的电脑上插入一个摄像头，然后修改成对应的 `int` 类型数字即可，如图 2 所示。

```
# Load camera  
cap = cv2.VideoCapture(0)  
# Flask
```

图 2 指定摄像头 0

3. 数据如何测试

首先我们打开工程目录，然后双击启动我们的工程，如图 3 所示。

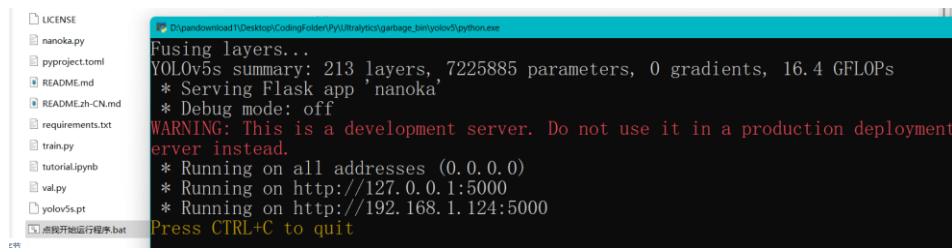


图 3 启动工程

使用 `CTRL+C` 可以退出程序，你的服务开启在了 `0.0.0.0` 上，对局域网有效，局域网 IP 为 `192.168.1.124`，服务端 5000。自己用浏览器测试一下，在浏览器内输入 <http://127.0.0.1:5000/val> 启动测试（测试 person 人类类别、cat 猫类别），如图 4 所示。

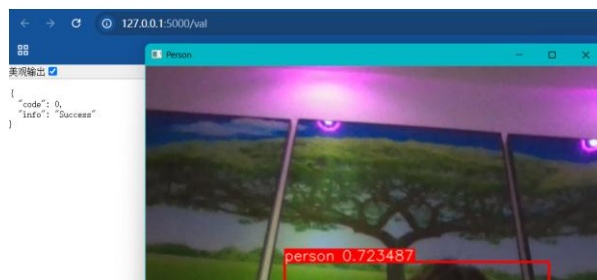


图 4 测试案例

能出来这个框那你基本上没问题，因为默认使用的摄像头 0 是你的前置相机。这时候你可以访问 <http://127.0.0.1:5000/image> 来获取静止图像测试，这个图像只会识别你设置的类别，所以不要乱问我为什么无效，而且类别要遵守在 `data\coco128.yaml` 中存在的 80 个类别，不要乱讲。

4. 数据如何使用

我就开放了一个 API，那就是 <http://127.0.0.1:5000/request> 这个，如图 5 所示。

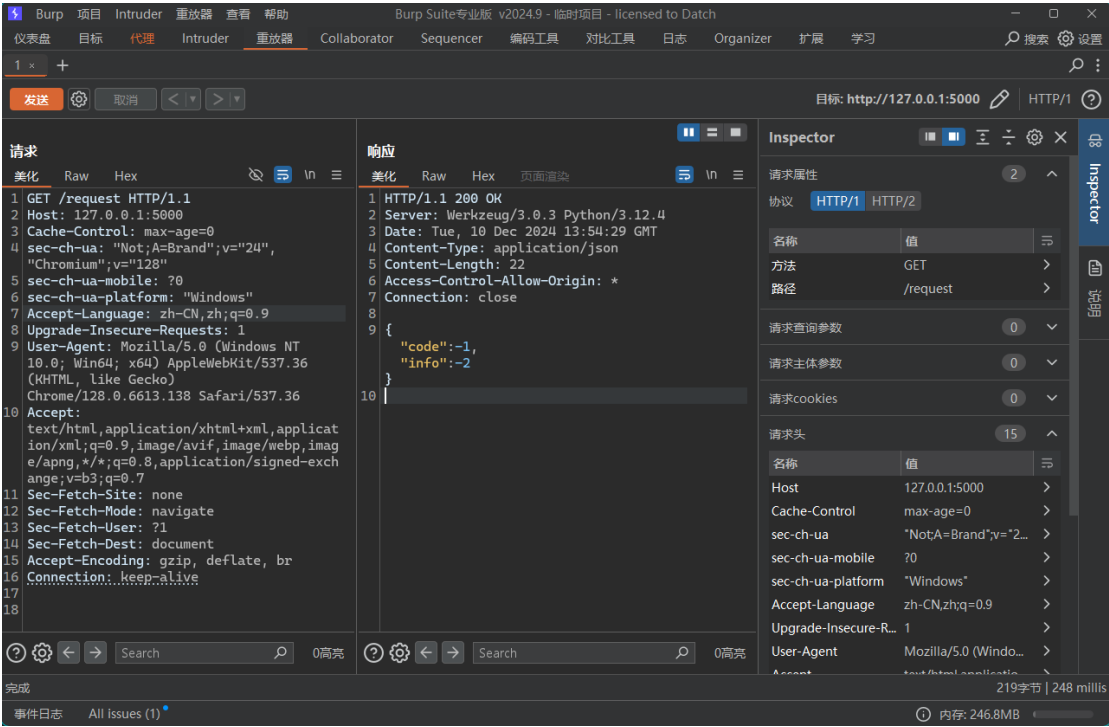


图 5 查看我们的请求和返回值

不难看出我们返回的是一个 **Json 组**，**keys** 分别为 `["code", "info"]`。

`code` 表示返回码（取值范围{0,-1}）。当 `code` 属性为 0 的时候表示数据正常；为-1 的时候表示数据异常。

`info` 表示返回类别（取值范围为{-1, -2} ∪ Q）。取值为-1 表示有多个物体存在，无法区分；取值为-2 时表示没有识别到对应物体；取值为 Q 内的数值时，则表示他们的对应分类。

本质上就是发送一个 **HTTP 的 GET 请求**即可，使用 **ESP32** 操作十分简单，但是要访问局域网地址，也就是 192.168.1.124 那个对局域网开放的地址，一定不要忘了加端口号。