

让ESP32-CAM小车“智能”起来

谢作如 浙江省温州科技高级中学
高毓甜 华南师范大学计算机学院

摘要: ESP32-CAM是一块性价比较高的开源硬件,适合用来做带拍照功能的Wi-Fi小车,如FPV小车。本文以一款开源的迷你小坦克为例,介绍了一种借助普通电脑算力,将小车升级为智能小车的技术方案。其核心思路是用普通电脑来实时处理Wi-Fi小车的摄像头画面信息,然后发送指令控制小车。这种“云边端协同”的做法为中小学“低成本”实施人工智能教育、开展无人驾驶方向的课程提供了一种可行的路径。

关键词: ESP32-CAM; 人工智能教育; 无人驾驶; JTank

中图分类号: G434 **文献标识码:** A **论文编号:** 1674-2117 (2023) 23-0075-03

前段时间,吴俊杰老师向笔者推荐了一个名叫“JTank”开源FPV小车的项目。该小车的核心板使用开源硬件,结构件采用3D打印,方案细节全部开源且可以自己组装。小车支持Wi-Fi控制,项目同时提供了App inventor开发的APP、微信小程序、网页等小工具,使用“第一视觉”(First Person View,简称FPV,是一种在遥控航空模型或者车辆模型上加装无线摄像头回传设备,在地面看屏幕操控模型的玩法)的方式操控,可玩性很强。整车价格不足百元,加上体积很小,很适合在课堂上作为一个教学机器人。

因此,笔者也购买了一辆小车,操控的感觉类似大疆的S1小车,很不错。在把玩一会儿后笔者

冒出新想法:能不能用普通电脑来读取这个小车的摄像头信息,然后编写Python代码控制小车?这样一来,就类似于在小车上外加了一个“大脑”,把这个普通的遥控小车升级为智能小车了。

● 对FPV小车的技术分析

这个FPV小车选择ESP32-CAM作为核心控制板。ESP32-CAM是基于ESP32芯片设计的一款开源硬件,它融合了Wi-Fi和蓝牙通信功能,对于需要拍照、录像、图像处理等应用场景非常适用。小车开机后会开启一个Wi-Fi热点,然后模拟成一个Web服务器,提供WebAPI实现来自手机端的各种控制。

通过项目作者,笔者得到了核心WebAPI接口地址,如下页表

所示。当然大家也可以自己查看ESP32-CAM源代码,或者修改代码增加更多的功能。

从表中可以看出,通信采用的是最常见的HTTP协议。用Python的Requests库,模拟各种App操控(发送HTTP请求)是很容易实现的。

● 为FPV小车增加“智能”的思路

显然,笔者目前没有办法让ESP32芯片直接处理摄像头的画面,只能采用让电脑来读取画面,然后进行实时分析。实际上,这种外加“大脑”实现智能的方式很容易理解,因为FPV操控方式本来就是用户来看画面,然后操作,至于这个用户是真实的人还是AI,都符合小车的设计思路。只要把这个普通电脑

Wi-Fi小车的核心WebAPI接口

功能	Web 地址
前进	http://192.168.4.1/state?cmd=F
后退	http://192.168.4.1/state?cmd=B
左转	http://192.168.4.1/state?cmd=L
右转	http://192.168.4.1/state?cmd=R
停止	http://192.168.4.1/state?cmd=S
摄像头	http://192.168.4.1:81/stream
灯光打开	http://192.168.4.1/state?cmd=Z
灯光关闭	http://192.168.4.1/state?cmd=w
低速挡位	http://192.168.4.1/state?cmd=T
中速挡位	http://192.168.4.1/state?cmd=Y
高速挡位	http://192.168.4.1/state?cmd=U

```
Python
from XEdu.hub import Workflow as wf
img = './pic.jpg'
ocr = wf(task='ocr')
texts= ocr.inference(data=img)
print(texts)
```

图2 核心代码

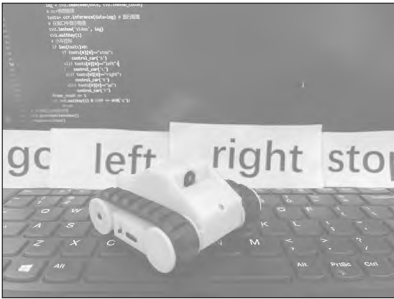


图3 “识行小车”图

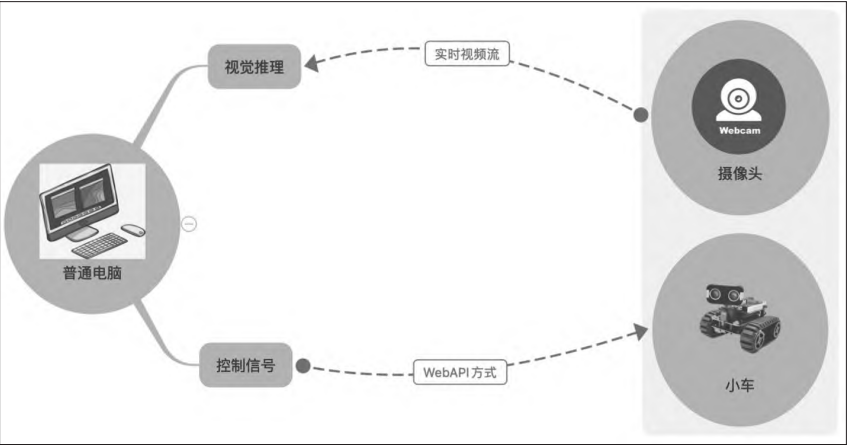


图1 让小车实现“智能”的示意图

想象成一个人就行了,如图1所示。

电脑上用什么工具来实时处理摄像头画面呢?笔者知道XEdu新增了一个名叫XEduhub的模块,内置了常见的深度学习模型,如图像分类、行人识别、人体和脸部关键点检测、OCR等,同时也支持自己训练的深度学习和机器学习模型。因此,借助XEduhub能够实现对常见物体的检测。

● XEdu代码实现和效果测试

笔者设计了几个简单的场景,

如小车检测到“手”或者特定的手势(交警的手势)就启动或者停止,再如小车识别到文字指令就执行相应动作,如left、right、go、stop等。然后,尝试用XEdu来实现这些功能。

XEduhub的核心语句为2句。导入库后用“wf(task=***)”语句来指定任务类型,用inference方法对数据(如图2)进行推理即可。以OCR任务为例,4句代码就能实现读入一张图,然后输出图片中的文

字信息。

笔者做了一个能认出文字指令的小车(如图3),用小纸条(写上指令)就能控制小车的前后左右和停止的运行状态,全部代码约50多行,核心代码和注释如下页图4所示。需要强调的是,对画面的读取要使用“requests.get()”,并且要加上“stream=True”的参数,这是因为这款FPV小车使用了视频流的方式来传输画面。不同的画面传输方式,也要对应不同的读取方式。

以上代码在XEdu的文档中即可获取,或者访问浦育平台,在项目栏目输入“ESP32-CAM”能够找到。但在浦育只能查看代码,不能真实运行,因为浦育平台的代码是运行在容器中,不能访问小车的IP地址。所以,如果代码要运行在本机中,还是建议下载XEdu的一键安装包,省去搭建环境的麻烦。

● 对“智能”小车教育价值的进一步思考

当完成了FPV小车的各种智

```

Python
# 导入库
import time, requests, cv2
import numpy as np
from XEdu.hub import Workflow as wf

# 定义控制小车运动函数
def control_car(cmd):
    url = "http://192.168.4.1/state?cmd=" # 拼接URL
    response = requests.get(url + cmd)
    time.sleep(0.3)
    requests.get(url='http://192.168.4.1/state?cmd=S') # 停止

# 定义相机流的URL, 需要根据实际设备或服务端的地址进行修改
url = 'http://192.168.4.1:81/stream'
response = requests.get(url, stream=True)
cv2.namedWindow('Video', cv2.WINDOW_NORMAL)
# 实例化模型为OCR文字识别
ocr = wf(task='ocr')
max_size=16384 # chunk大小实际情况进行调整
for chunk in response.iter_content(chunk_size=max_size):
    data = np.frombuffer(chunk, dtype=np.uint8)
    img = cv2.imdecode(data, cv2.IMREAD_COLOR)
    texts= ocr.inference(data=img) # 进行OCR推理
    # 小车控制
    if len(texts)>0:
        if texts[0][0]=="stop":
            control_car('S') # 停止
        elif texts[0][0]=="left":
            control_car('L') # 左转

```

图4 核心代码

能应用实验后,笔者发现了一个新的实施人工智能教育的方案,并开始思考这一方案的教育价值。因为长期以来,大家受传统机器人教育

的影响,总会认为人工智能的模型推理要在开源硬件上完成,人为提高了实施人工智能教育的门槛。

首先,可以肯定这是一种低成

本的实施人工智能教育的方式。相对来说,能实现AI模型推理功能的小车主板都比较昂贵,而借助这种方式能够将任何一种可以遥控的小车升级为“智能”小车,都可以用来做无人驾驶课程。当然,需要注意的是不要让这个小车做竞速之类的工作,毕竟网络传输还是会产生一定的延时。

其次,这种方式与当前流行的“云边端协同”的计算方式是一致的。各种支持网络读取的摄像头,都可以赋予一定的“智能”,学生可以结合校园中各种摄像头设备,实现各种真实场景应用,如研究校园暴力行为、研究农场的虫害情况、研究校园的生态信息等。另外,用Wi-Fi摄像头(ESP32-CAM)来做车牌识别,其效果可能不会逊色于当前商业购买的智能门禁。

参考文献:

[1]XEdu文档.XEduHub功能详解[EB/OL].(2023-12-1)[2023-11-23].https://xedu.readthedocs.io/zh/master/xedu_hub/introduction.html #.

[2]XEdu文档.XEduHub项目案例集[EB/OL].(2023-11-23)[2023-12-1].https://xedu.readthedocs.io/zh/master/xedu_hub/projects.html #.e