

四足巡检机器人（具有图像识别能力）

1.简介

该项目是本人参加学校苗子工程项目做的作品，该作品是基于开源项目[【毕业设计】履带四足复合机器人 - 立创开源硬件平台](#)所开发的，与原作品相比本作品使用了原作品中所提到的mg996r舵机来实现以较低成本的四足结构，另外本作品没有使用openmv来实现图像识别，而是使用更划算的ESP32cam来实现图像识别，该图像识别的方式是由32cam传输视频流到服务端，在服务端上进行图像识别后，将识别标注后的图像发送到客户端。不过该方式有缺点，需要强大的服务端来进行图像识别处理，因此在理想的云端服务器下，使用公网ip，是可以突破仅限于局域网范围内的。本作品的四足行走效果并不理想，如果预算充足还是建议使用原作品使用的舵机。

2.开发流程

2.1.摄像头开发

2.1.1准备工作

首先在网上选购ESP32cam（可以额外加购不同类型的摄像头，如平角、广角）

编译平台（ArduinoIDE/Thonny）

不同编译平台对应着不同编译语言，使用ArduinoIDE则要使用C++语言开发，使用Thonny则要使用micropython语言开发

2.1.2开发环境准备

如果使用ArduinoIDE开发的话，首先你需要安装开发板

安装ESP32开发板：

1.打开文件(File)->首选项(Preferences)

在附加开发板管理器网址（Additional Boards Manager URLs）添加一下URL

```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
```

2.安装ESP32开发板

打开工具(Tools)->开发板(Board)->开发板管理器(Boards Manager)

在开发板管理器中搜索esp32

找到esp32 by Espressif Systems，点击安装

安装完成后，你就可以在工具->开发板中找到ESP32的各种开发板型号

Note

上面的安装方法需要科学网络才可以顺利安装，如果没有科学网络请去Arduino社区寻找开发板包下载

3.环境配置

点击工具->开发板->esp32, 选择AI Thinker ESP32-CAM
选择好后点击文件->示例->ESP32->Camera->CameraWebServer

在文件开头把

```
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
```

的注释去掉

修改你的ssid与password, 点击运行, 打开浏览器输入串口调试器显示的ip就可以看到摄像头效果了,
如果你有什么想法可以根据示例代码进行修改

如果使用thonny开发

点击运行->配置解释器, 配置好esp32cam相关的包

然后, 连接32cam, 写入程序

2.1.3训练模型

参考另一篇PDF文档[使用yolov8训练模型](#)

推荐素材资源网站: [Safety Helmet and Reflective Jacket](#)

这个网站可以下载很多图片训练资源, 节省拍摄图片准备步骤

模型训练网站: [MaixHub](#)

这个网站可以训练很多类型的模型, 可以轻松入手

2.1.4搭建图像识别服务端

原理是esp32传输视频流到服务端, 服务端讲识别标注后的图像发送到客户端

打开pycharm,就在ultralytics-main目录下创建一个新的py文件
使用程序:

```
import socket
import cv2
import io
from PIL import Image
import numpy as np
from ultralytics import YOLO
import os

# 设置YOLO模式
os.environ["KMP_DUPLICATE_LIB_OK"] = "TRUE"
model = YOLO("你训练好的best.pt模型路径") # load a custom model

# 设置接收来自ESP32-CAM视频流的套接字
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM, 0)
s.bind(("0.0.0.0", 9091))

while True:
    data, IP = s.recvfrom(100000) # 从ESP32-CAM接收视频帧
    bytes_stream = io.BytesIO(data)
    image = Image.open(bytes_stream)
    img = np.asarray(image)
```

```

img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR) # 将RGB转换为OpenCV的BGR

# 使用YOLO模型对捕获的帧进行对象检测
results = model(img) # 对帧执行检测

# 在框架上显示结果
result_img = results[0].plot() # YOLO在图像上可视化结果的方法
cv2.imshow("ESP32-CAM YOLO Detection", result_img)

# 按q键退出
if cv2.waitKey(1) == ord("q"):
    break

cv2.destroyAllWindows()
s.close()

```

使用方法：

打开电脑的cmd，输入ipconfig查看自己的ip，在32cam代码输入ip与端口号(代码里使用的是9091端口)

将代码写入后，运行程序电脑应该就能接收到识别后的图像了

2.2.QT页面编写

软件安装与环境搭配请参考CSDN,如[【保姆级图文教程】QT下载、安装、入门、配置VS Qt环境-CSDN博客](#)

首先创建好工程后，点击ui文件，在这里你可以图形化设计ui界面

简单说一下怎么使用，构建套件建议使用MSVC,不建议使用MinGW,因为有些ui功能使用不支持MinGW，在创建项目的时候就要选好，否则后期在左边栏点击项目进去配置比较麻烦。

工程创建好后项目框架为一个.h、一个main.cpp和.cpp，还有.ui这几个文件，在.h文件引入需要的库函数，.h文件里的private slots声明函数，

private里定义变量

在.cpp文件里写实现的方法(不是 main.cpp)。

你可以直接使用本文件夹目录里的qt工程，直接看效果

2.3小车组装与编写程序

2.3.1组装

打螺丝跟拼积木一样就可以组装了

建议拼好腿就可以开始烧录程序测试了，把履带接上了再测试舵机可能会承受不住折腾

2.3.2程序实现

分析功能：舵机控制、电机控制、接收传感器信息（姿态调整、避障等功能），wifi通信、摄像头控制