



语音识别智能尋物小车

组长: 黄启明

组员: 张毅涛 张栩宁 邹莲麦子

指导老师:曹迪

ZJUTER

软外03班

00 项目意义以及场景搭建

01 系统简介与亮点特色

02 系统硬件设计与实现

03 系统软件设计

04 总结与感悟









00

项目意义以及场景搭建



项目意义以及场景搭建



将计算机视觉智能和计算机的语言智能结合,对物理世界进行改变。

可以作为一个模块搭载在其他需要根据进行物体识别和自动寻找的智能机器人上。

项目意义以及场景搭建



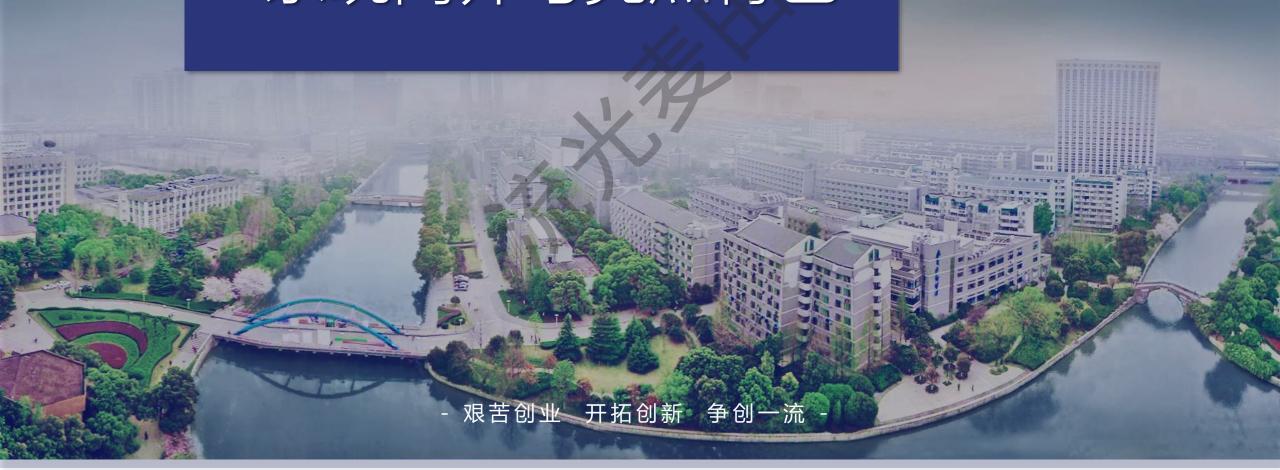
将主要可以识别的几个 物体和多个干扰项放在 一起,测试视觉识别的 精度。将小车放在远处, 让其自动寻路。





01

系统简介与亮点特色



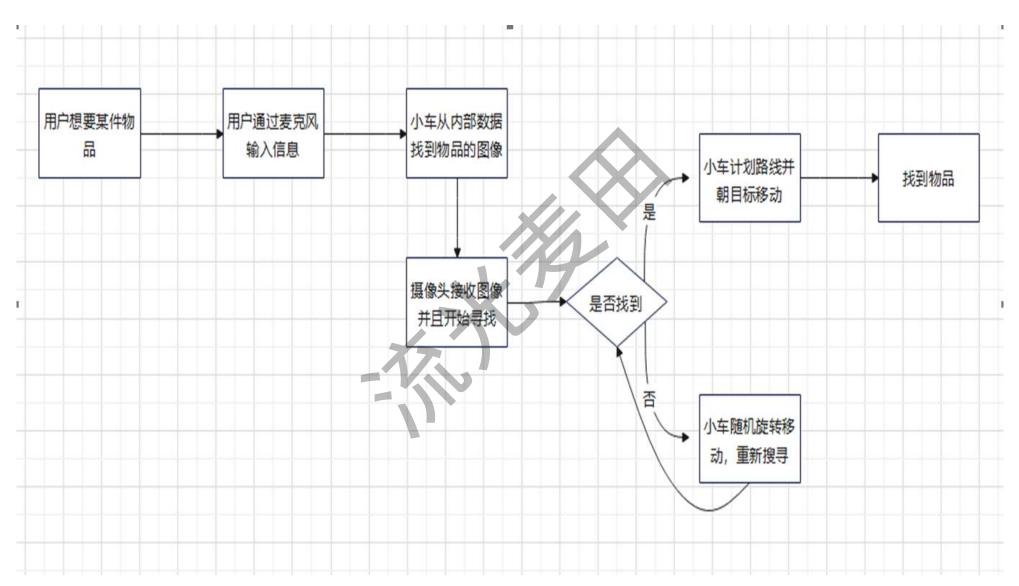


语音识别自动寻物系统,用于在一定空间内,在使用者语言指令的控制下,小车能够从语言中提取关键词,从而利用计算机视觉,去寻找目标寻找物。

该系统的意义在于,利用语音控制和计算机视觉便利了人们寻物的动作过程,将现在比较流行的语言智能和计算机视觉智能结合在一起,并作用于物理世界的实践。

业务逻辑





厚德健行

业务逻辑



小车摄像头实时拍 摄环境图片传给PC

使用者在环境中说出寻物指令

SHE SHE

图像信息

数字信息

将图像信息读 入YOLO模型 中进行分析

小车主板根据PC端 的数字信息实时改 变小车的运动路径

小车端

PC端

亮点特色



语音输入控制:用户可以通过语音输入指令,提供给小车需要寻找的目标物体名称。

目标图像生成:小车通过得到的目标物体图像,在嵌入的物体图像库中找到相对应的物体种类。

目标检测:小车搭载摄像头,通过图像识别技术,可以识别出特定的物体。

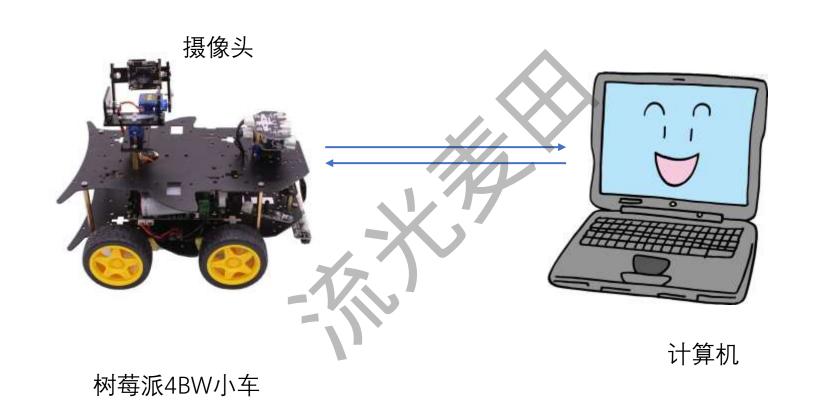
自主导航:小车可以基于目标检测结果,自主导航并朝向目标物体前进。



02

系统硬件设计与实现







03

系统软件设计



系统核心功能实现



- (1)能够识别出用户的语音输入内容并回答,并且提取一个关键词并保存。
- (2)系统可以通过物体的图像提取出该物体的名称。
- (3)小车与计算机有实时通信,根据计算机的实时通信改变路线。
- (4)小车的寻找是对当前环境巡视一圈,类似于人在没有线索时找东西的过程中扫视周围环境。

(1) 识别出用户的语音输入内容并回答



Openai ChatGPT-3

```
if "相关" in spoken_command:
    content_prompt = f"请根据关键词\"{keyword}\" 生成一段有美的内容"
    content = generate gpt3_response(content_prompt)
    speak(content)

else:
    modified_response = f"我在找{keyword}快让我看看 "
    speak(modified_response)
    with open("D:/Desktop/animal_follower/PC/target.txt", "w", encoding="utf-8") as f:
        cleaned_keyword = keyword.lower().replace('"', '')
        f.write(cleaned_keyword)
```

(2) 图像辨认



```
def scan(dir):
   net = model zoo.get model('yolo3 darknet53 coco', pretrained=True, root='D:\AI project\gluoncv\model')
   try:
       x, img = data.transforms.presets.yolo.load test(dir, short=512)
       print('Shape of pre-processed image:', x.shape)
       class IDs, scores, bounding boxs = net(x)
       # print('class:', class IDs[0][0])
       # print('scores:', scores[0][0])
       # print('bounding_box:', bounding_boxs[0][0])
       with open("D:/Desktop/animal_follower/PC/target.txt","r",encoding="utf-8") as f:
           target=f.read()
       print("我在找"+target)
       targetIDs=selectID(class IDs)
       for key in targetIDs:
           if net.classes[targetIDs[key]]==target:
               print("我看到了",target)
               return class_IDs[0][key],scores[0][key],bounding_boxs[0][key]
       # plt.show()
       print("没有找到目标")
       return class IDs[0][50], scores[0][50], bounding boxs[0][50]
   except:
       print("刚刚那张图片损坏了")
   return 0
```

(3) 实时通信



利用socket通信将小车与性能较好的计算机相连,使用计算机来进行模型的使用并将分析结果用简单的数字信息传送给小车,使得系统对环境图片具有更高效的分析能力,提高了系统整体的算力。

```
def upload():
    try:
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        #192.168.199.1和8088分别为服务端 (pc) 的IP地址和网络端口
        s.connect(('192.168.50.107', 8086))
    except socket.error as msg:
        print(msg)
        print(sys.exit(1))

video = cv2.VideoCapture("http://192.168.50.1:8080/?action=stream")
    cv2.namedWindow("img",cv2.WINDOW_NORMAL)
    ret,frame = video.read()
    time.sleep(2)
    if(frame is None):
        print("2")
    cv2.imwrite("/home/pi/Desktop/gluoncv/image.png", frame)
```

(4) 寻物策略



```
if y2 - y1 > 250:
   if x1 > 640 or x2 > 640 or x1 < 0 or x2 < 0 or y1 > 480 or y2 > 480 or y1 < 0 or y2 < 0:
       # 图像显示不全, 需要后退
       back(t / 2)
       print("back")
   else:
       if (x1 + x2) / 2 > 400: # 目标偏左
          x=(((x1+x2)/2)-340)
          t=x/680*t
          right move(t)
          print("close_right_move")
       if (x1 + x2) / 2 < 240: # 目标偏右
          x=(340-((x1+x2)/2))
          t=x/680*t
          left_move(t)
          print("far_left_move")
       if 240 <= (x1 + x2) / 2 <= 400: # 位置合适,不动,接着拍摄即可# 找到了目标, 但是严嵩太远, 需要接近目标
          print("位置合适,ok")
          run_stop_put()
else:
   if y1 > 480 or y2 > 480 or y1 < 0 or y2 < 0:
       # 图像显示不全, 需要后退
       back(t / 2)
       print("back")
   else:
       if (x1 + x2) / 2 > 400 or x1<20 or x2<20:# 目标偏左
          t=(x2-x1)/680*2*t
          right_move(t)
          print("right_move")
       if (x1 + x2) / 2 < 240 or x2>640 or x1>640: # 目标偏右
          t=(x2-x1)/680*2*t
          left_move(t)
          print("left move")
       if 240 <= (x1 + x2) / 2 <= 400:# 位置合适找到了目标, 但是距离太远, 需要接近目标
          forward(t)
          print("forward")
```



<u>04</u>

总结与展望



总结



我们小组在这一个星期的时间完成了以下工作:

- 1. 环境搭建: 将学校提供的已组装完成的树莓派小车可以通过手机APP进行短 距离控制,并且成功与电脑上的树莓派操作系统连接,可以连接上WIFI热点 进行代码、网页的操作。
- 2. 功能开发:使用Python编程语言完结合树莓派的GPIO、OpenCV以及Chatgpt等的API,开发了小车的控制程序和图像识别程序通过计算机网络技术的学习内容将这些功能相连接,使小车能够通过语音输入指令,搜寻捕捉指定物体并进行自主导航避障。
- 3. 测试和优化:在完成上述程序后,我们对整个系统进行了反复测试和优化,确保其稳定性和可靠性,最终可以基本稳定的实现语音树莓派寻物小车的基本功能。

 厚德健行



如果有充裕的时间,或者在后续学期甚至毕业设计时,我们认为可以实现以下比较有创意、有应用前景的实验:

- 1.**基于深度学习的目标检测与跟踪**:通过引入深度学习技术,训练模型来识别和跟踪特定目标,实现小车更加智能化的自主导航,可以设计更加复杂的路线,并添加相关的寻路算法以使其应用于更加广泛的场合。
- 2.**基于语音识别的自然语言对话系统**:通过引入自然语言处理技术,实现小车与人的自然语言对话,使得人们可以更加方便地控制小车。
- 3.**基于机器学习的行为预测和规划**:通过引入机器学习技术,对小车的行为进行预测和规划,使得小车能够更加灵活地适应不同的环境和任务。
- **4.基于树莓派各个机器组件的组装和实现**:通过将树莓派小车和机械臂或坦克等其他组件的结合、合作,实现功能更完整、实用性更强的智能电子机械产品,比如实现物体的寻找和抓取找回。
- 5.**基于摄像头人脸识别功能的交互**:通过引入摄像头的人脸识别技术,实现树莓派智能小车的人脸识别登录和操控,增加系统的安全性。



退出ppt播放视频

厚德健行