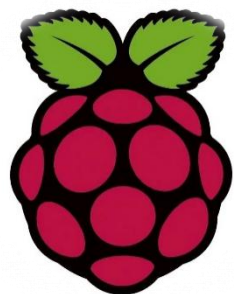




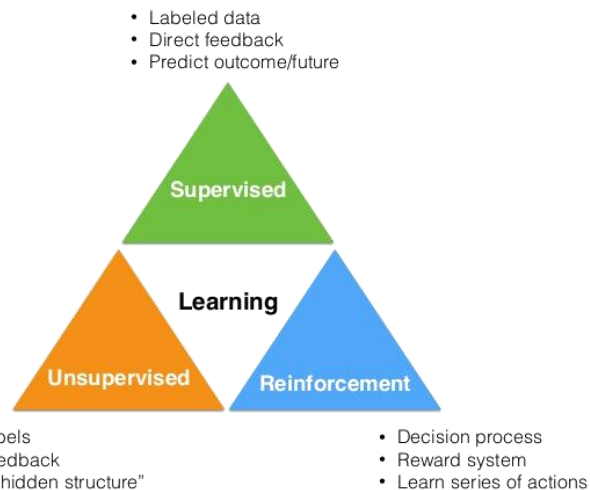
人工智能普及教育



人工智能普及教育的目标



- 让所有人简单快速了解和体验人工智能功能生产和应用的过程，消除新技术神秘感，对新技术有能力做基础的预判。
- 大部分人能了解人工智能、物联网、传感器等组合运用的过程，并能和现实中部分常见的应用场景相对照
- 一部分人基于已有框架运用人工智能和编程能解决实际问题，服务于学习、工作和生活
- 少部分人能持续学习，不断创新



机器学习实操的7个步骤



Step 1
收集数据



Step 2
数据准备



Step 3
选择模型



Step 4
训练



Step 5
评估



Step 6
参数调整



Step 7
预估



目 录

PART 01 人工智能教学案例



PART 02 物联网教学案例



PART 03 综合实践案例



PART 04 无限拓展案例

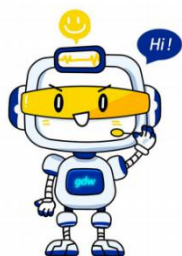




Part 1

人工智能教学案例

机器学习数据采集、标注、训练、验证、应用过程理论和实践
语音识别、文字识别等案例实践促使融会贯通

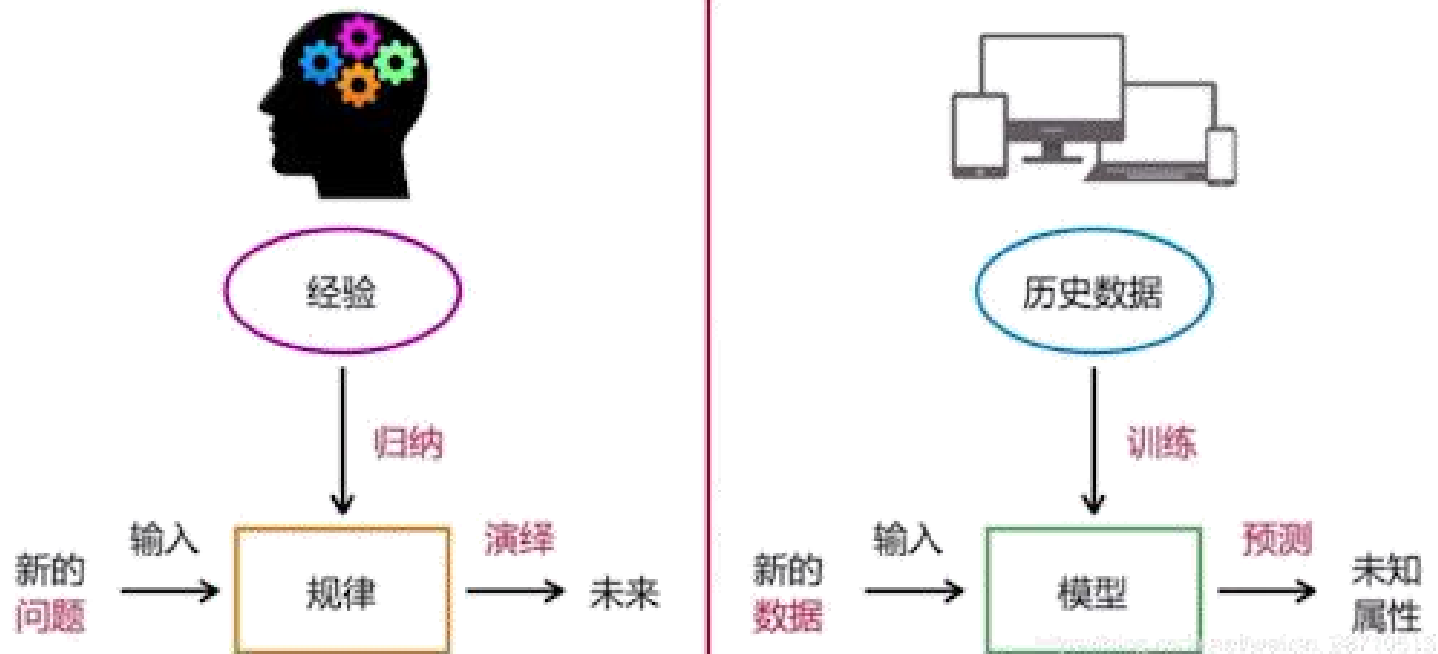


学习的可能性



- 机器学习的方式和人脑思考的方式很相似。

人类思考 vs 机器学习



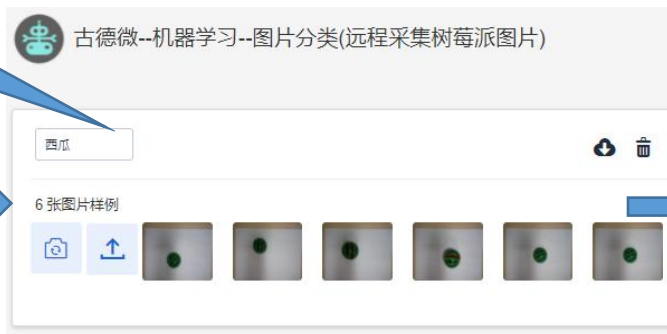
机器学习实践过程



拍照采样



标注



训练

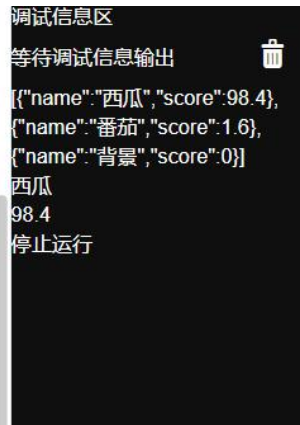


下载模型应用

验证



图形化或python调用模型



```
img = Image.fromarray(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
# 创建一个可以在给定图像上绘图的对象
draw = ImageDraw.Draw(img)
# 字体的格式
fontStyle = ImageFont.truetype('/usr/share/fonts/truetype/wqy/wqy-zenhei.ttf', textSize)
# 绘制文本
draw.text((left, top), text, textColor, font=fontStyle)
# 转换回OpenCV格式
return cv2.cvtColor(np.asarray(img), cv2.COLOR_RGB2BGR)

labels_path = '/home/pi/model/labels.txt'
model_path = '/home/pi/model/model.tflite'

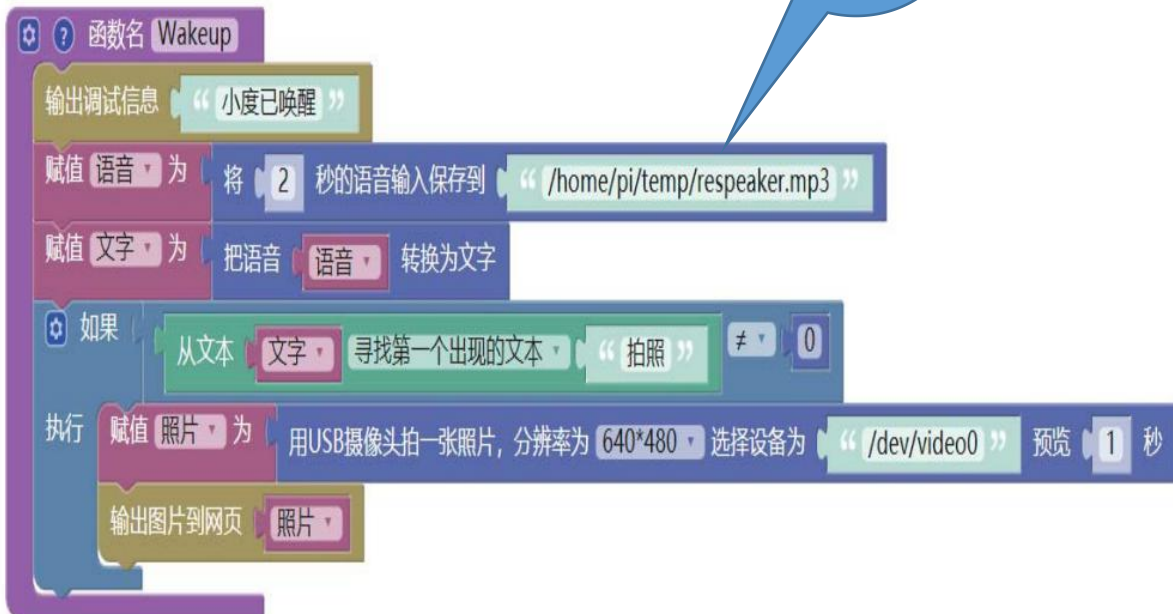
labels = load_labels(labels_path)
```



人工智能应用



小度小度关键词语音唤醒，请创建一个Wakeup新函数



语音拍照

人脸检测

```
import cv2 #导入opencv库

# 检测图像中的人脸位置信息
def detect_face(img): # 检测人脸函数
    # 将图片等比例缩小1/2, 以便加快检测速度
    small_img = cv2.resize(img, (0, 0), fx=0.5, fy=0.5)
    # 将图像转换为灰度图像, 因为opencv人脸检测器需要灰度图像
    gray = cv2.cvtColor(small_img, cv2.COLOR_BGR2GRAY)
    # 加载OpenCV人脸检测分类器Haar
    face_cascade = cv2.CascadeClassifier('/usr/local/share/opencv4/haarcascades/haarcascade_frontalface_alt.xml')
    # 检测图片中人脸图像, 返回值是一张脸部区域信息的列表 (左上角x,y,宽,高)
    face_rects = face_cascade.detectMultiScale(gray)
    return face_rects * 2

# 根据给定的 (x, y) 坐标和宽度高度在图像上绘制矩形
def draw_rectangle(img, rect): # 绘制矩形框函数
    (x, y, w, h) = rect
    cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)

# 创建摄像头对象
cam = cv2.VideoCapture(0)

# 设置摄像头的分辨率
cam.set(cv2.CAP_PROP_FRAME_WIDTH, 640);
cam.set(cv2.CAP_PROP_FRAME_HEIGHT, 480);

# 判断摄像头是否打开
while cam.isOpened():
    ret_val, img = cam.read() # 从摄像头读取, 返回两个值, 第一个是布尔值, 表示读取图像是否正确, 第二个值代表图像三维像素矩阵
    if ret_val:
        face_rects = detect_face(img) # 从图像中检测人脸的位置信息
        for rect in face_rects:
            draw_rectangle(img, rect) # 在人脸的位置处画一个矩形
            cv2.imshow('photos', img) # 创建一个窗口显示图片, 窗口的标题为photos
            if cv2.waitKey(1) == 27: # 如果检测到按下ESC键就退出
                break
    cv2.destroyAllWindows() # 释放资源
```

Step1 创建摄像头对象, 并设置分辨率

Step2 从摄像头读取图像并检测人脸位置信息

Step3 在人脸位置处绘制矩形框, 并通过窗口显示出

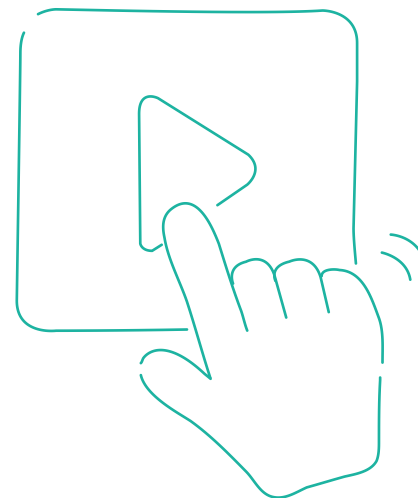




Part 2

物联网教学案例

远程控制、数据采集、万物互联



远程控制灯



```
import paho.mqtt.client as mqtt
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
LED=5
AN=25
GPIO.setup(AN, GPIO.IN)
GPIO.setup(LED, GPIO.OUT)
client=mqtt.Client()
client.connect("www.gdwrobot.top",1883,60)
time.sleep(1)
while True:
    isanxia = GPIO.input(AN)
    if isanxia == 1:
        client.publish('zengliang2/LED', "开灯")
    elif isanxia == 0:
        client.publish('zengliang2/LED', "关灯")
    time.sleep(0.1)
```

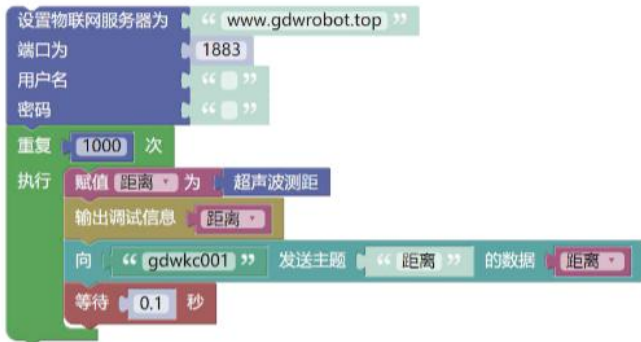
控制树莓派上面小灯:



数据采集



③回到古德微编程网页，点击“更多功能”回到编程界面，编写发送端的程序，运行程序，进行测试。



④回到数据采集网页，呈现图标。



③使用 python 编写发送随机数程序，运行程序，进行测试。

```
import paho.mqtt.client as mqtt
import time
import random
client=mqtt.Client()
client.connect("www.gdwrobot.top",1883,60)
time.sleep(1)
while True:
    tstDat= random.choice(range(100))
    client.publish('zengliangjc/sjs', tstDat)
    time.sleep(0.1)
```

④回到数据采集网页，呈现图标。



万物互联



①将红外遥控扩展板安装到树莓派上。



通过红外、蓝牙等，可将各种外围设备接入物联网远程控制

②设置红外发射模式和 GPIO 端口号，并调试输出当前红外模式和 GPIO 端口号。

设置为 红外发射模式 ， GPIO为 17

输出调试信息 获取当前红外模式和GPIO号

调试信息区
等待调试信息输出
["send", 17]





Part 3

综合实践案例

物联网语音点歌台、手势识别拍立得、食堂餐盘计价仪

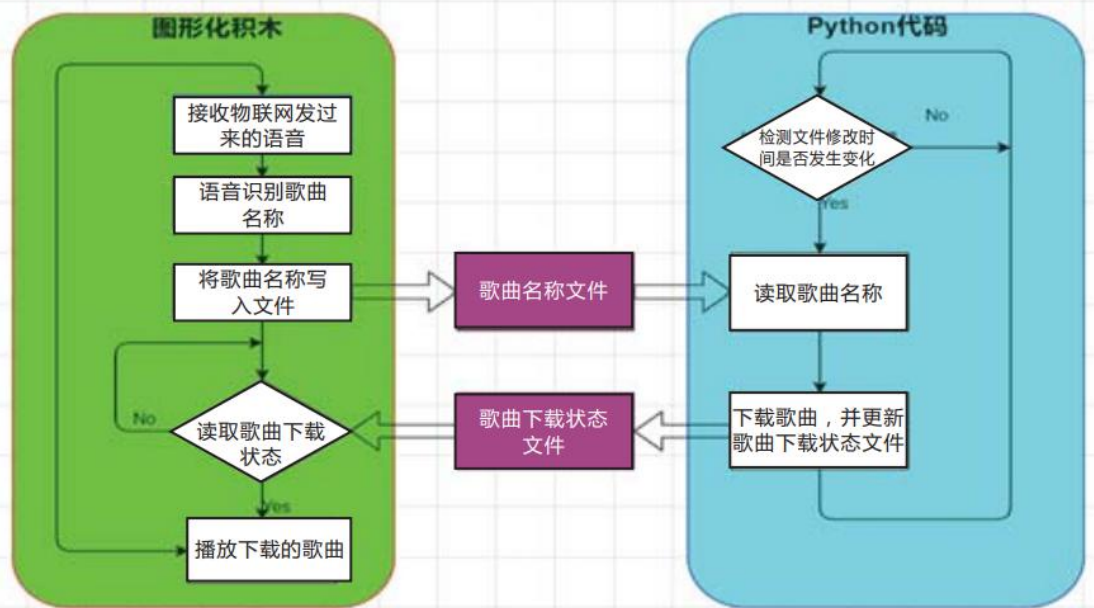


物联网点歌台



01 该案例的流程图如下。

图形化积木接收点歌语音，获得歌名，将歌名写入文件，等待python代码下载歌曲完成后播放



04 Python代码 测是否有需要下载的音乐名字，如有则下载音乐 完整源码参考/home/pi/pycode/qqmusic.py 主函数如下图：

```
#Step 1:初始化对象，其中QQmusic是自定义的类，用来下载歌曲
QQ = QQmusic()
songpath = '/home/pi/temp/song.txt' #保存歌曲名称的文件路径

#Step 2:获取初始状态时，保存歌曲名称文件的修改时间。
with open(songpath, 'w') as f:
    mtime_before = os.path.getmtime(songpath)

#Step 3: musicok.txt记录了音乐是否下载完成的状态，此处初始化为init状态。
with open('/home/pi/temp/musicok.txt', 'w') as f_musicok:
    f_musicok.seek(0, 0)
    f_musicok.write('init')

#Step 4: 删除当前文件所在目录下所有的mp3文件
os.system('sudo rm -rf *.mp3')

#Step 5: 开始循环检测保存歌曲名称的文件是否有被修改，如果被修改，说明需要下载歌曲
while True:
    #获取歌曲名称文件的修改时间
    mtime = os.path.getmtime(songpath)
    #如果文件的修改时间发生了变化，说明有新的歌曲名称写入
    if mtime != mtime_before:
        mtime_before = mtime
        print(1111)
        with open(songpath) as f:
            songname = f.readline() #读出歌曲名称
            QQ.downloadMusic(songname) #根据歌曲名称下载歌曲
        time.sleep(0.01)
```

百度搜索“python qq 音乐下载”可以找到该案例的python代码和说明。
Python最大优势是基本所有想法都能下载到源代码或库



手势识别拍立得



课程内容

Part1 打印机的使用

主要内容：打印机基础知识介绍、打印文本、打印图片、生成条形码并打印、生成二维码并打印。

机械电子知识：按键的连接与使用、打印机的连接方法；

人工智能知识：条形码分类、编码方式及应用，二维码编码方式及应用。

Part2 摄像头的使用

Part3 拍照并打印

Part4 拍立得照相机的组装

Part5 人脸识别初步

Part6 手势识别

主要内容：手势识别基础知识介绍、拍一张人脸照片并拍照、识别手势控制打印输出。

编程知识：列表的建立与查询。

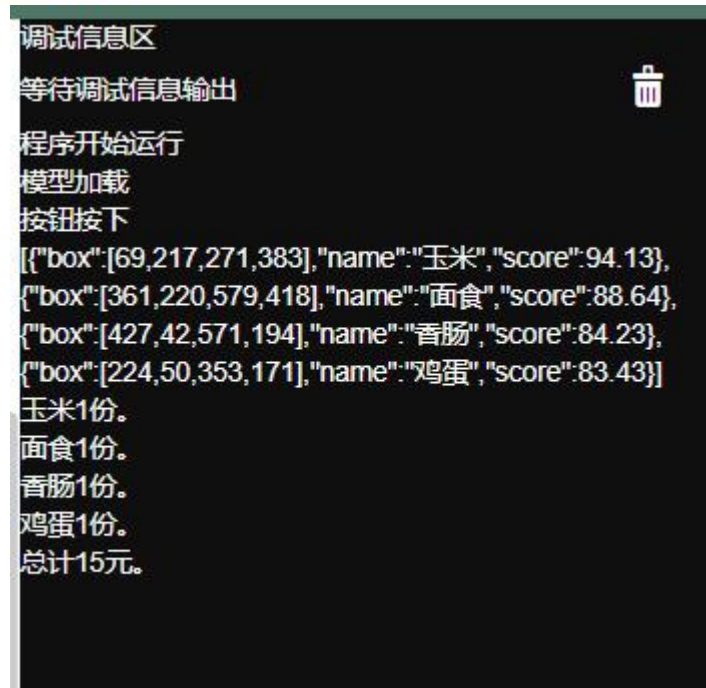
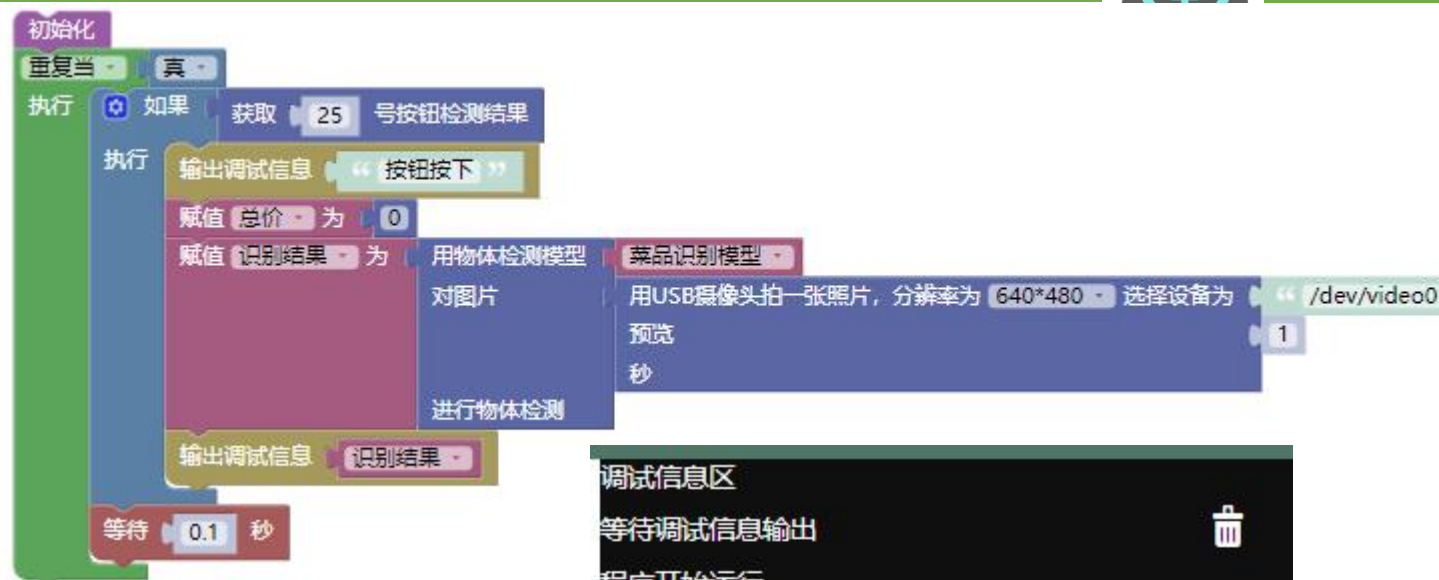
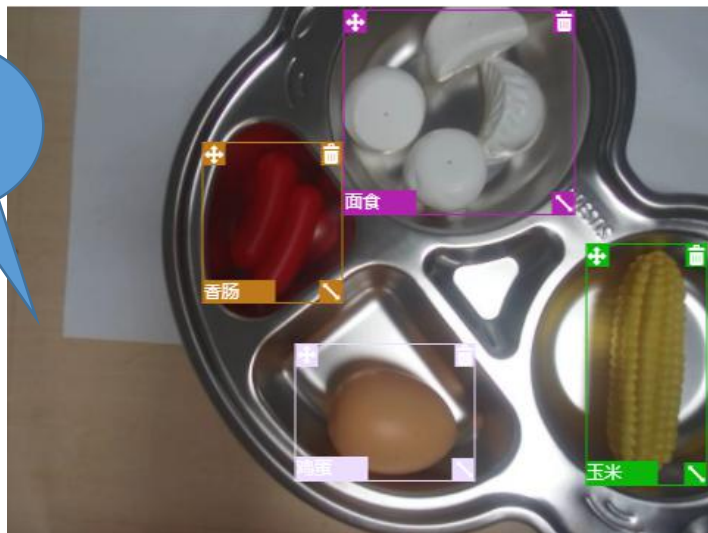
人工智能知识：手部区域识别、概率分数。



食堂餐盘计价仪



标注
训练



```
38 draw.text((0, 0), 'fps='+str(fps), tuple(textColor), font=fontStyle)
39 return cv2.cvtColor(np.asarray(img), cv2.COLOR_RGB2BGR)
40 except:
41     traceback.print_exc()
42     return None
43
44
45 labels_path = '/home/pi/model/object_detection/labels.txt'
46 model_path = '/home/pi/model/object_detection/model.tflite'
47 g_threshold = 0.7
48
49 labels = load_labels(labels_path)
50 print("#load model")
51 interpreter = Interpreter(model_path = model_path)
52
53 interpreter.allocate_tensors()
54 input_details = interpreter.get_input_details()
55 output_details = interpreter.get_output_details()
56 _, height, width, _ = input_details[0]['shape']
57
58
59 try:
60     cap = cv2.VideoCapture(0)
61     while cap.isOpened():
```



图形化或
python
识别计价

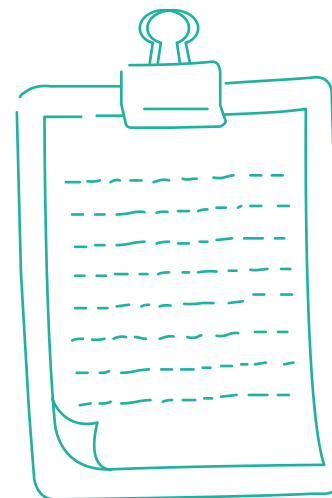




Part 4

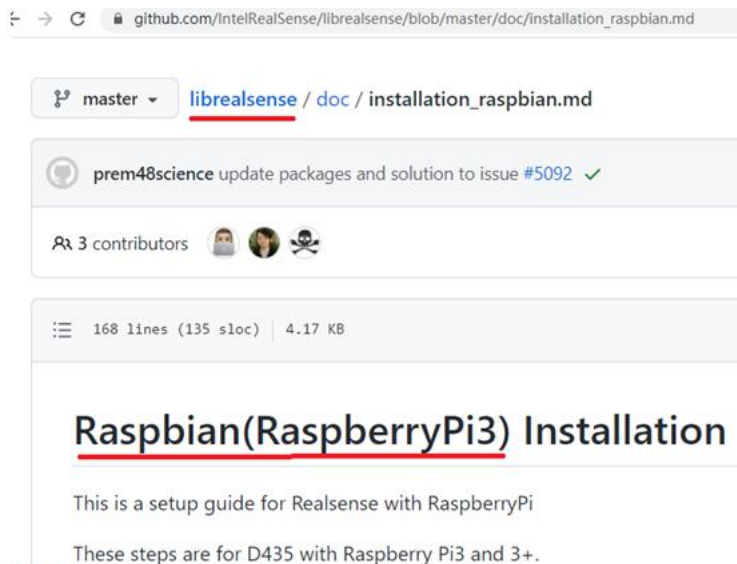
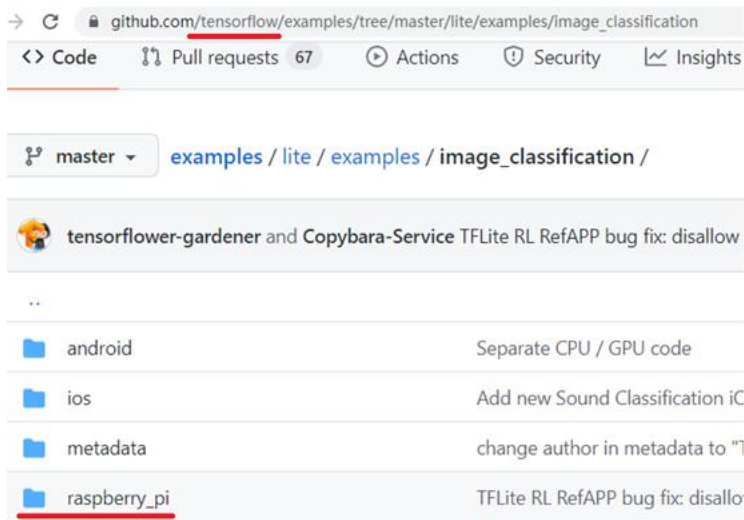
无限拓展案例

树莓派是全功能微型电脑，有全功能软硬件开源社区
各大厂商都对树莓派兼容支持



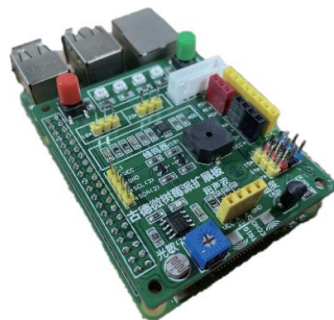


开放性强、稳定性高、独创性佳 各大厂家软硬件都主动支持树莓派



各大厂家主动支持树莓派（包括Google和intel）

树莓派



主控板有树莓派3B、4B等，扩展板集成了多个传感器的功能，开放性强、稳定性高。基于树莓派几乎可以实现所有创意，各大厂家的软件、硬件都主动支撑树莓派，古德微和老师们只需专注于研究教学需求和课程需求。

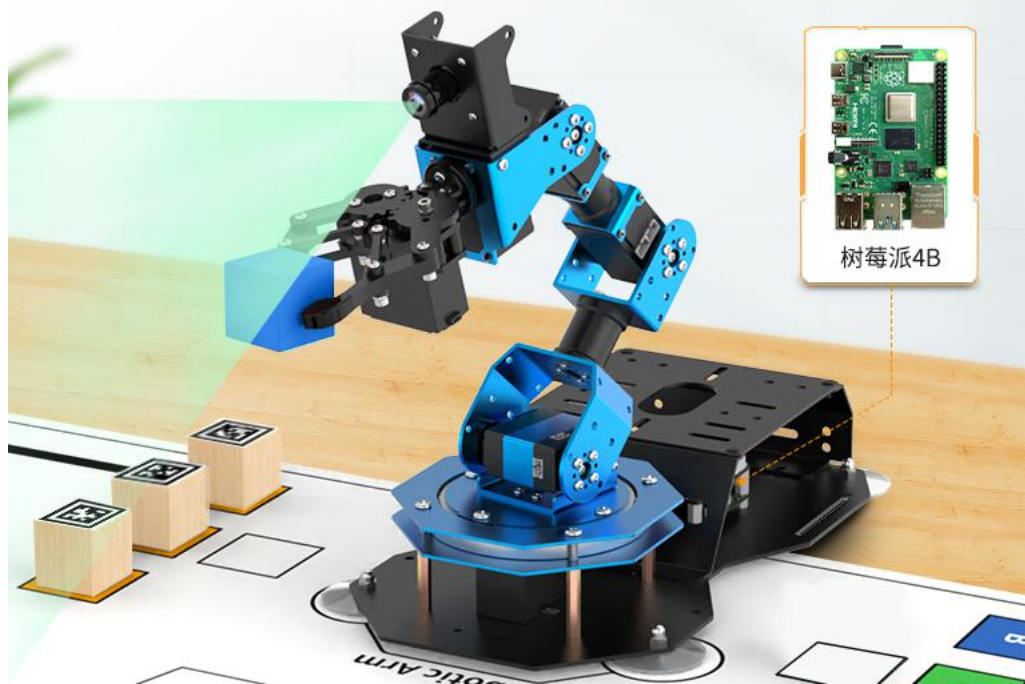
学生具备自我学习的能力后，可以持续学习，运用人工智能和编程能解决实际问题，服务于学习、工作和生活



无限拓展

ArmPi FPV智能视觉机械臂以树莓派4B为主控，OpenCV为图像处理库，搭载高清120°广角摄像头，拥有第一视觉。它采用ROS机器人操作系统，内置MoveIt运动学。通过Python编程，它可以实现人脸识别、颜色追踪、码垛、智慧仓储等多种AI视觉识别功能！

它不仅能满足用户对机器视觉、机械臂运动控制、力矩控制等算法的学习和验证，还为手眼协作、视觉抓取等二次开发提供快速、便捷的集成方案。我们提供学习资料，开放源代码，让您快速玩转AI智能机械臂！



 **树莓派4B**

TonyPi智能视觉人形机器人

AI人工智能 / Python编程 / OpenCV / 机器视觉

- 智能搬运
- 语音交互
- 自动踢球
- 人脸识别
- 智能巡线
- 颜色追踪
- 标签识别

手机APP控制

树莓派4B





感谢观看

