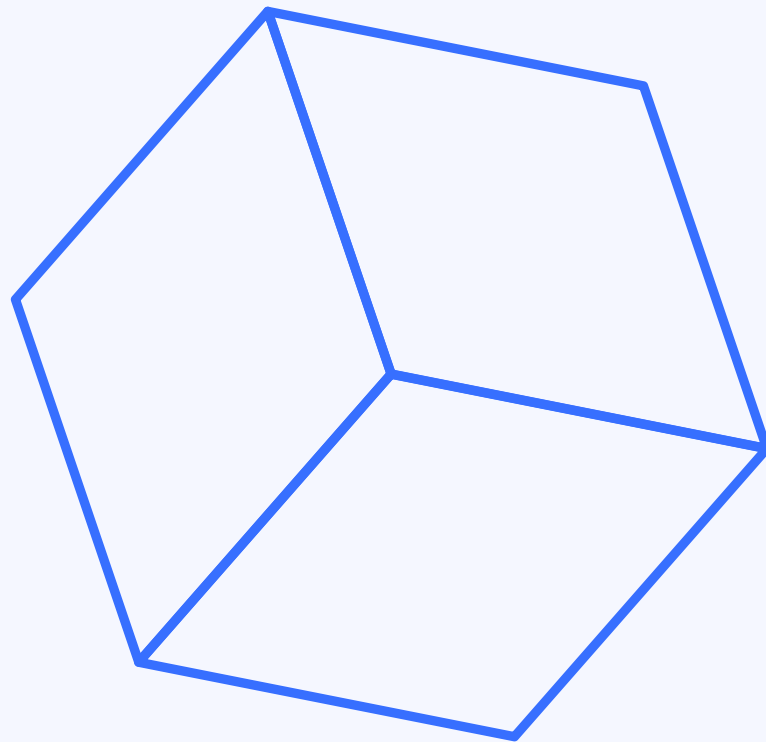


# 基于YOLO11-pose 的运动计数系统

汇报人：tao355667

2024-12-27



# 目录

CONTENTS

- 01 研究背景与意义
- 02 YOLO11介绍
- 03 系统设计与实现
- 04 软件部署
- 05 总结与未来展望





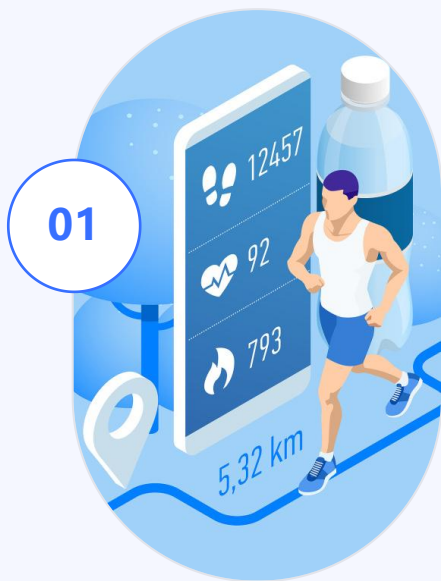
01

# 研究背景与意义

章节副标题



# 健身计数系统的发展趋势



## 健身计数系统普及的原因

随着健康生活方式的普及，人们越来越关注日常锻炼的科学性和效果评估，健身计数系统因此得到了广泛的研究和应用。



## 健身计数系统的技术演进

从最初的简单计数器到现在的计算机视觉和深度学习技术，健身计数系统的技术演进体现了对更准确、更智能健身跟踪的需求。

# 计算机视觉与深度学习



## 01、计算机视觉技术的应用

计算机视觉技术通过分析视频或图像数据，能够识别和跟踪人体动作，为健身计数系统提供了基础。

## 02、深度学习技术的应用

深度学习技术，尤其是卷积神经网络（CNN）和循环神经网络（RNN），在动作姿态估计方面取得了显著进展，提高了系统的准确性和效率。



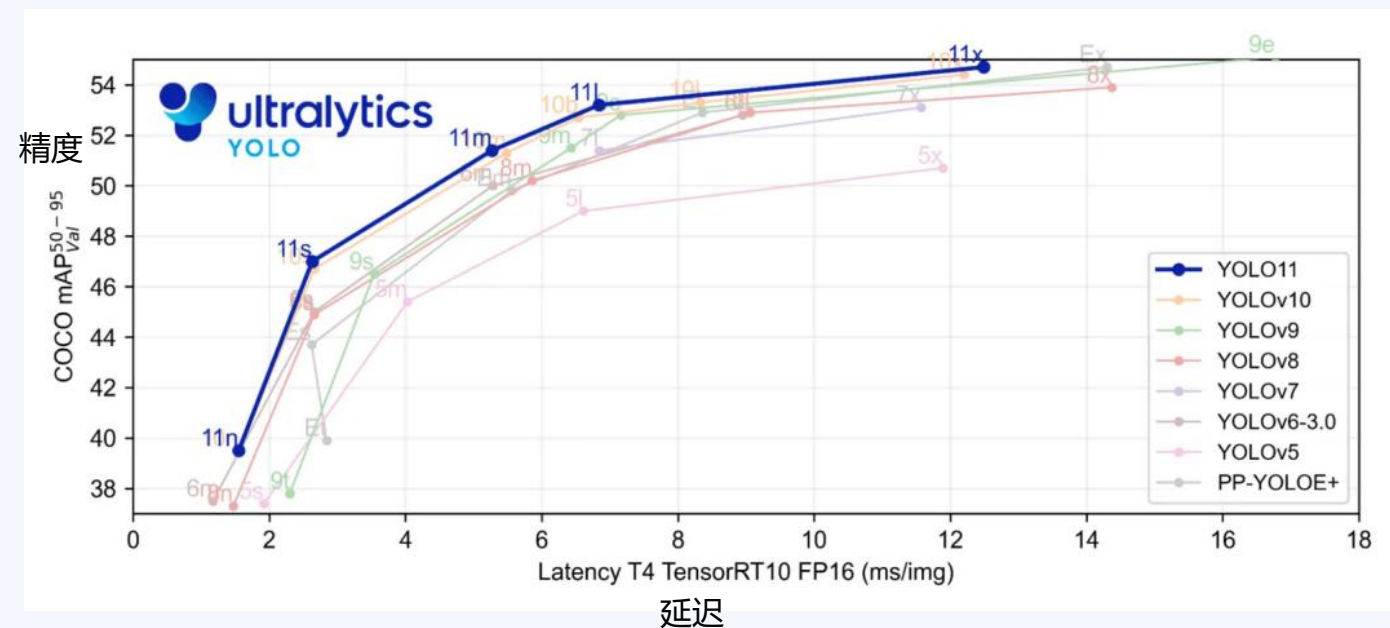
02

# YOLO11介绍

章节副标题



# YOLO11的架构与性能改进



## YOLO11架构的创新点

YOLO11在前代YOLO版本的基础上，引入了C3k2和C2PSA模块，实现了端到端的目标检测能力，提高了性能和灵活性。

## YOLO11性能的提升

YOLO11在保持同等精度的同时，参数量降低了20%，在速度和准确性方面表现出色，支持多种计算机视觉任务。

# YOLO11支持的计算机视觉任务

## 物体检测与实例分割

YOLO11能够识别和定位图像中的物体，并在它们周围绘制边界框，同时进行分类。此外，它还能将图像分割成不同的区域，并为每个区域分配标签。

## 图像分类与姿态估计

YOLO11使用EfficientNet架构的变体进行图像分类，并通过姿态估计检测图像或视频帧中的关键点，用于跟踪动作或姿态。



**物体检测 (Detection)**：识别和定位图像中的物体，并在它们周围绘制边界框，同时对这些物体进行分类。

**实例分割 (Segmentation)**：将图像分割成不同的区域，并根据内容为每个区域分配标签，这在医学成像等应用中非常有用。

**图像分类 (Classification)**：基于图像内容将图像分类到不同的类别中，YOLO11使用EfficientNet架构的变体来进行分类任务。

**姿态估计 (Pose estimation)**：检测图像或视频帧中的特定关键点，用于跟踪动作或姿态，这些关键点被称为关键点。

**定向物体检测 (Oriented Object Detection, OBB)**：通过增加一个角度来更准确地定位图像中的旋转物体，这比传统的物体检测更为精确。

**物体跟踪(Track)**：在连续的视频帧中监控和跟踪物体的移动。



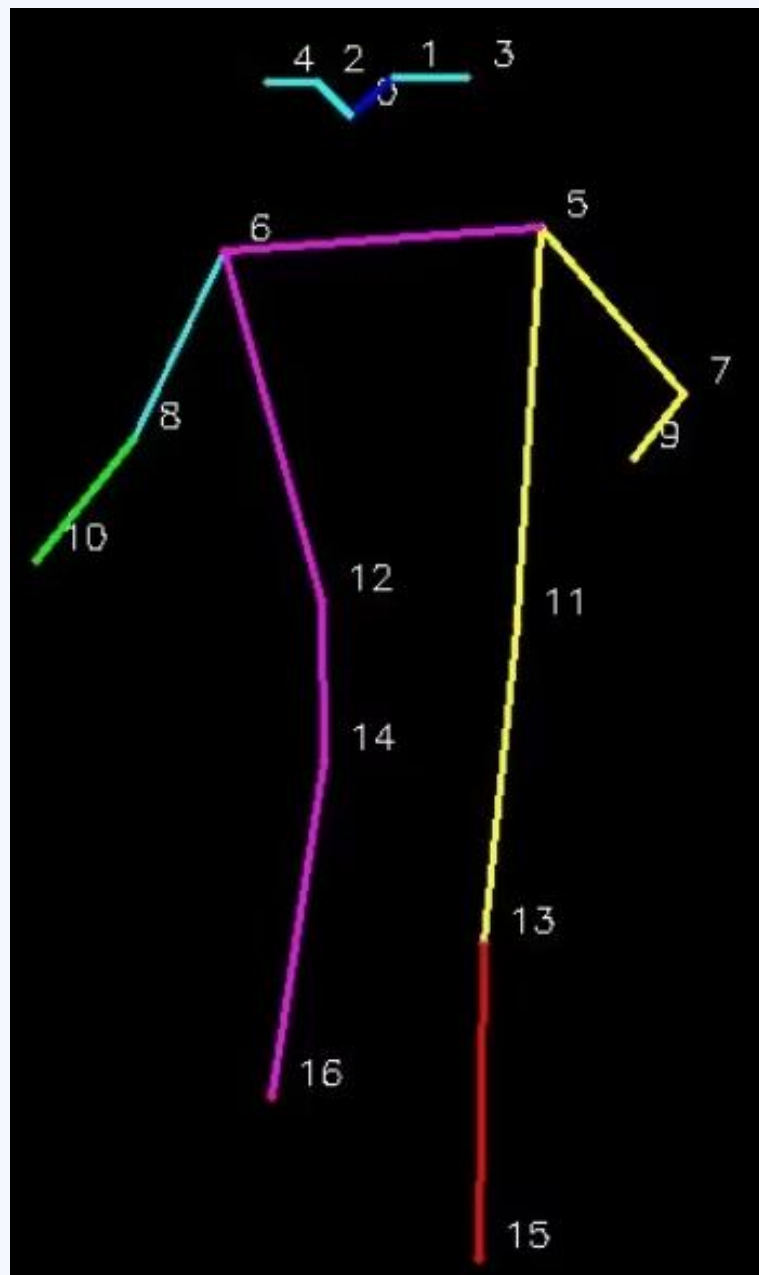
# YOLO11-pose模型的关键点

## 关键点检测

在YOLO11-pose中，有17个关键点，每个关键点代表人体的不同部位，以下是每个索引到相应身体关节的映射：

0: 鼻子, 1: 左眼, 2: 右眼, 3: 左耳, 4: 右耳, 5: 左肩, 6: 右肩, 7: 左肘, 8: 右肘, 9: 左腕, 10: 右腕, 11: 左髋, 12: 右髋, 13: 左膝, 14: 右膝, 15: 左脚踝, 16: 右脚踝。

人右



人左



03

# 系统设计与实现

章节副标题



# 软件架构与开发工具

## 开发工具的选择

开发语言为Python，利用PySide6框架构建GUI，确保软件可以在Windows和Linux系统上运行。使用VS Code作为主要的代码编辑器，利用PySide6-Designer和PySide6-UIC等工具进行界面设计和代码转换。

## 软件架构概述

本系统采用模块化设计，主要由UI界面类（Ui\_MainWindow）、YOLO11接口类（yolo\_workouts）和主程序类（MainWindow）三个类组成，实现了系统的功能并方便后续的维护和升级。



# 运动计数功能的逻辑



```
if self.angle[ind] < self.down_angle:
    if self.stage[ind] == "up":
        self.count[ind] += 1
    self.stage[ind] = "down"
elif self.angle[ind] > self.up_angle:
    self.stage[ind] = "up"
```

`self.angle[ind]` : 获取用户当前的角度

更新状态:

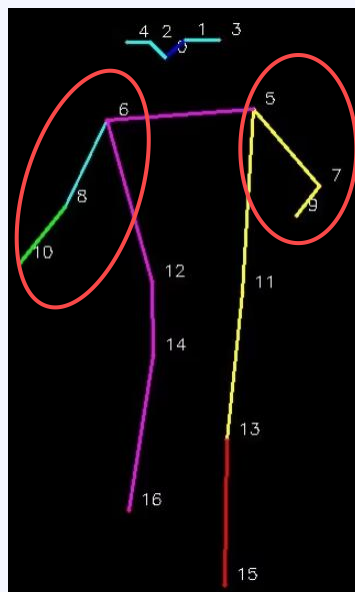
如果当前的角度小于下限阈值, 则将用户的当前阶段 `self.stage[ind]` 设置为 "down", 表示用户现在是处于 "下" 位置。接着, 如果当前角度大于上限角度 `self.up_angle`, 这说明用户已返回到 "上" 位置, 因此将当前阶段更新为 "up"。

更新计数:

在确认用户处于 "下" 阶段时, 代码接着检查 `self.stage[ind]` 的值。如果用户之前处于 "上" 阶段 (即 `self.stage[ind] == "up"`), 那么这说明用户完成了一次重复的锻炼, 因此计数器 `self.count[ind]` 增加1。

# 俯卧撑计数

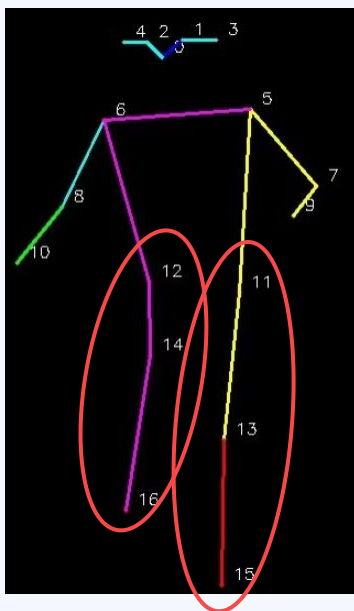
俯卧撑计数选择了关键点[6,8,10]（右肩、右肘、右手）和关键点[5,7,9]（左肩、左肘、左手），通过计算关键点的角度变化，进行运动计数。用户可根据自己与摄像头的位​​置关系，选择合适的朝向，以保证检测效果，若右手臂距离屏幕较近，就选择关键点[6,8,10]（右肩、右肘、右手），若左手臂距离屏幕较近，就选择关键点[5,7,9]（左肩、左肘、左手）。计数的上角度阈值为 $131^\circ$ ，下角度阈值为 $89^\circ$ 。







# 深蹲计数

深蹲计数识别了两组关键点：一组包括[12,14,16]，对应于身体的右侧（右腰、右膝、右脚）；另一组包括[11,13,15]，对应于身体的左侧（左腰、左膝、左脚）。通过分析这些关键点之间的角度变化，系统能够准确计数深蹲动作。为了优化检测效果，用户需要根据自己相对于摄像头的位置选择合适的朝向。如果用户的右腿更靠近摄像头，系统将优先考虑右侧的关键点；相反，如果左腿更靠近，系统将优先考虑左侧的关键点。计数的上角度阈值为 $131^\circ$ ，下角度阈值为 $89^\circ$ 。





选择模型:

选择运动项目:

运动项目的朝向:  
身体距离摄像头较近的一侧, 如左臂距离摄像头近, 就是左侧

模型从小到大: n s m l x  
pt可直接推理  
ncnn适合树莓派 (yolo说的), 但实际上效果不好  
onnx需要在全局python环境中安装onnx库

角度: 51.19

计数: 3

阶段: 下

计数清零

选择视频文件

摄像头开始

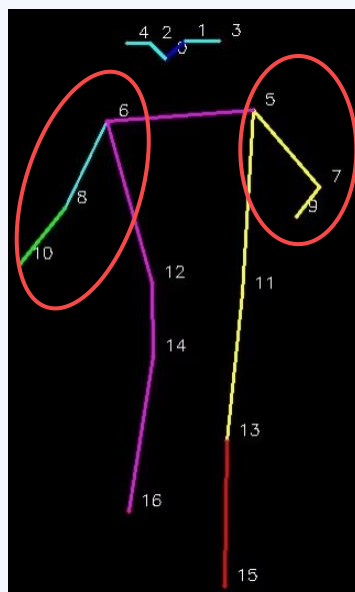
视频开始 ☐ 保存识别结果视频

停止

状态: 文件 D:/code/YOLO-motion-counting/motion-counting/video/60kg前蹲-右.mp4 播放中

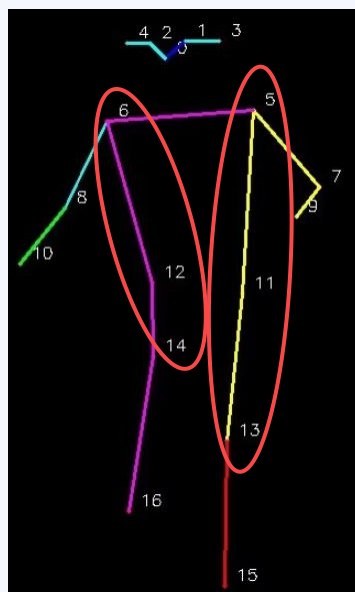
# 引体向上计数

引体向上计数选择了关键点[6,8,10]（右肩、右肘、右手）和关键点[5,7,9]（左肩、左肘、左手），通过计算关键点的角度变化，进行运动计数。用户可根据自己与摄像头的位关系，选择合适的朝向，以保证检测效果，若右手臂距离屏幕较近，就选择关键点[6,8,10]（右肩、右肘、右手），若左手臂距离屏幕较近，就选择关键点[5,7,9]（左肩、左肘、左手）。计数的上角度阈值为 $131^\circ$ ，下角度阈值为 $89^\circ$ 。



# 仰卧起坐计数

仰卧起坐计数选择了关键点[6,12,14]（右肩、右腰、右膝）和关键点[5,11,13]（左肩、左腰、左膝），通过计算关键点的角度变化，进行运动计数。用户可根据自己与摄像头的位关系，选择合适的朝向，以保证检测效果，若右手臂距离屏幕较近，就选择关键点[6,12,14]（右肩、右腰、右膝），若左手臂距离屏幕较近，就选择关键点[5,11,13]（左肩、左腰、左膝）。计数的上角度阈值为 $115^{\circ}$ ，下角度阈值为 $85^{\circ}$ 。



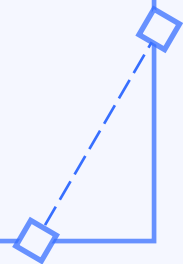




04

# 软件部署

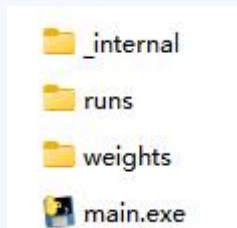
章节副标题



# 软件打包



## 使用PyInstaller打包软件



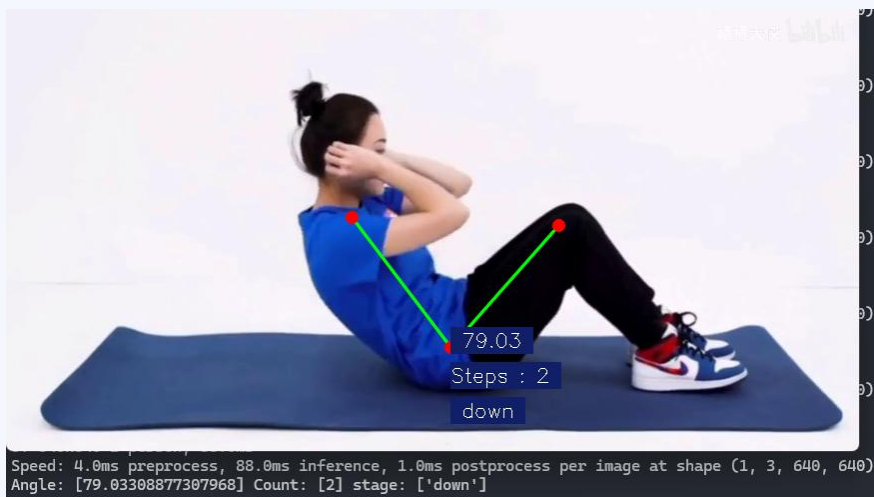
PyInstaller是一个流行的开源工具，它允许开发者将Python程序打包成单个可执行文件（在Windows上是.exe文件，在macOS上是.app文件，在Linux上是没有前缀的可执行文件）。

## 打包过程中的注意事项

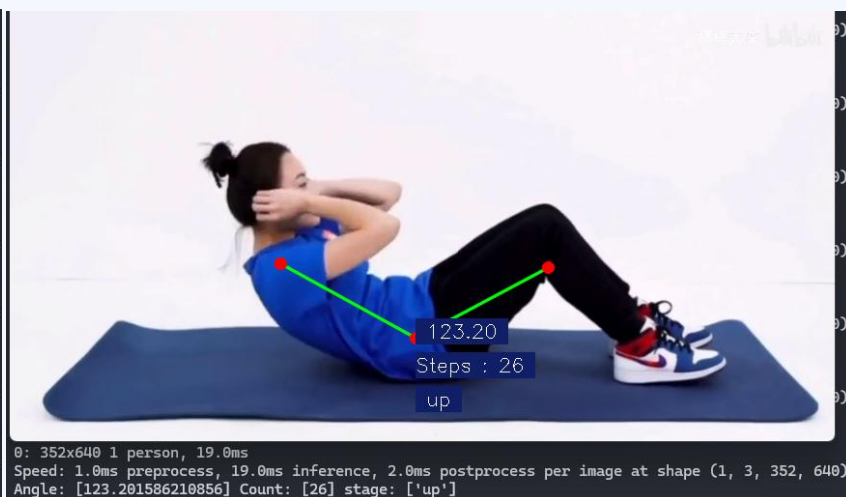
1. 在打包过程中，确保所有依赖项都已正确安装，并且程序能够在开发环境中正常运行。
2. 需要将模型权重文件和视频文件等外部资源也进行打包。

# Windows部署

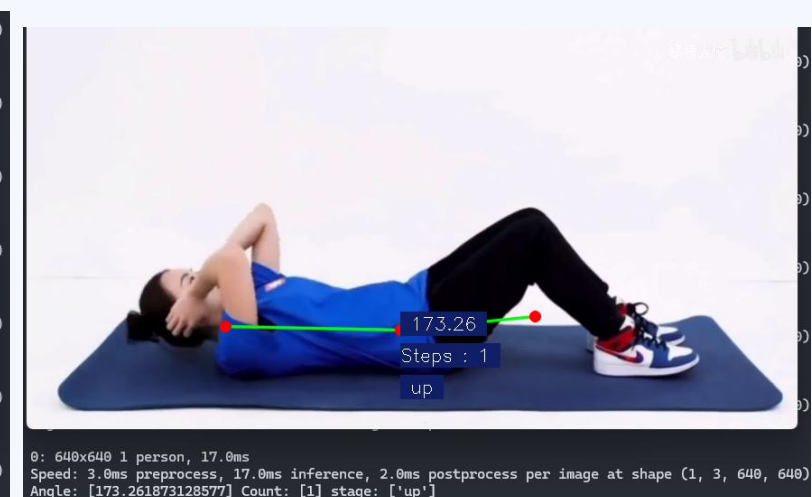
在Windows平台上，软件支持.pt、ONNX、NCNN格式的模型。对于.pt格式，需要确保NVIDIA GPU支持CUDA，并安装相应的CUDA Toolkit和cuDNN。对于ONNX格式，需要安装onnxruntime-gpu。



NCNN (YOLO11n) (约90ms)



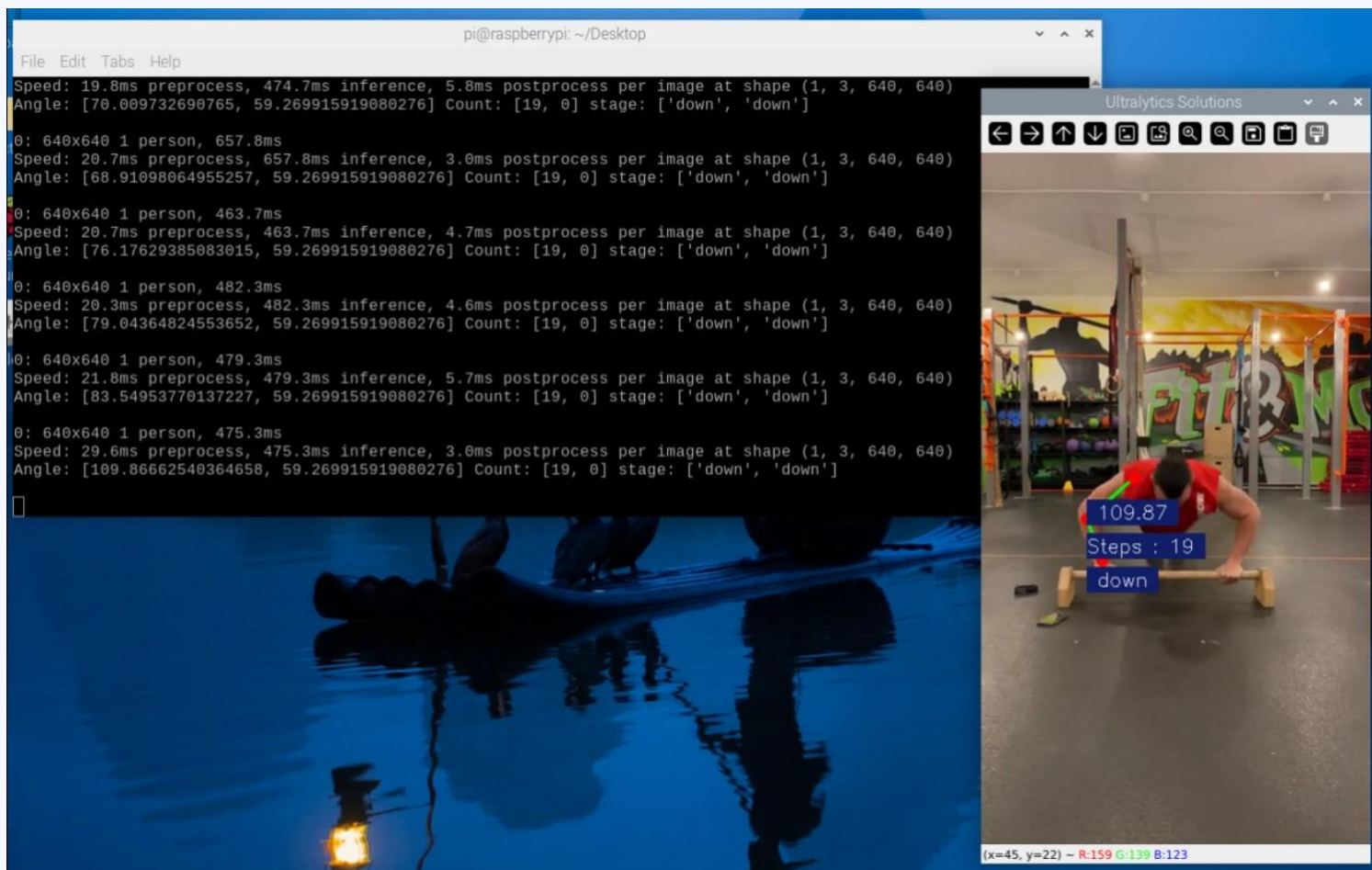
.pt (YOLO11n) (约20ms)



ONNX (YOLO11n) (约20ms)

# 树莓派部署

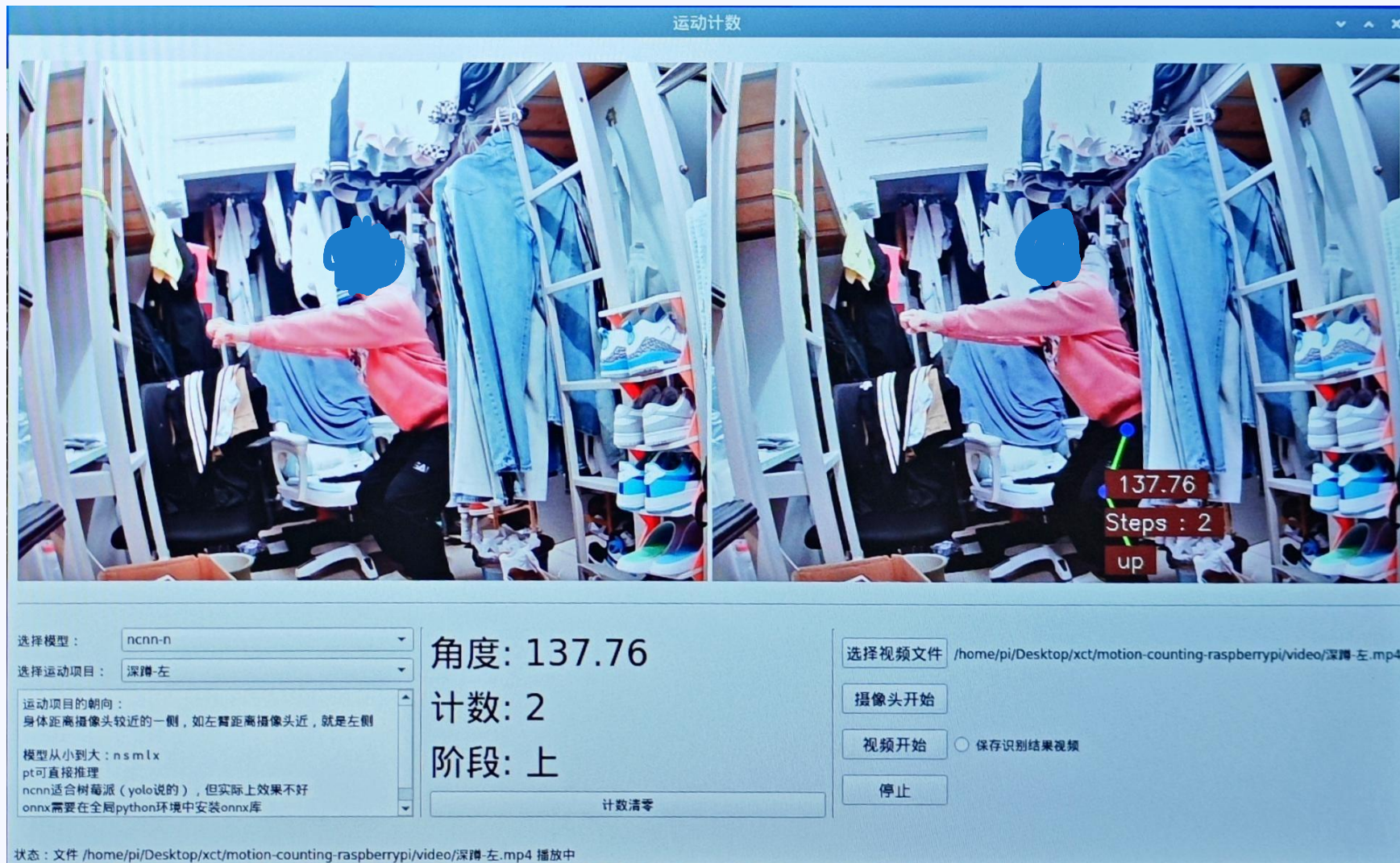
因为硬件性能限制，树莓派上通常运行YOLO11n模型，NCNN格式针对移动和嵌入式平台进行了优化，适合在树莓派上运行AI任务。



树莓派上推理NCNN (YOLO11n) (约500ms)



# 树莓派部署



软件在树莓派上运行



05

# 总结与未来展望

章节副标题



# 总结与展望



## 总结

成功开发了一款基于YOLO11-pose的运动计数系统，该系统能够准确识别和计数多种健身动作，如俯卧撑、深蹲等，为用户提供科学的锻炼数据和反馈。



## 展望

为了提高用户便利性，系统将优化以支持移动设备，如智能手机和平板电脑，实现随时随地的健身跟踪和数据同步。



谢谢

