

# 基于 OpenCV 的智能识别跟踪云台设计总结报告

2023 年 9 月 21 日

## 1 研究背景与意义

计算机视觉主要是通过计算机及摄像机等相关设备对生物视觉进行模拟，利用计算机相关特点来进行视觉信息的处理。而运动目标的检测与跟踪是计算机视觉领域一个非常热门的课题，在模式识别、图像处理、医学成像等多个领域有着非常广泛地应用，更是很多视觉应用系统的重要组成部分，如智能视频监控、人机交互、智能驾驶等。随着计算机技术的迅速发展，智能视频监控技术与人们的日常生活有着密切相关的联系，同时给人们带来了极大的便利。

与传统的视频监控不同，智能视频监控系统主要利用图像处理、模式识别和计算机视觉等技术，同时借助智能视频分析模块与目标跟踪模块，快速准确的对运动目标进行检测、跟踪、识别等一系列的高层次语义上的理解，实现全自动、全天候、实时的智能监控。随着技术的发展以及国家和人民对安防要求的日益提高，智能视频监控技术在社会生活的各个领域发挥着重要的作用。

在军事领域，智能视频监控系统在国家安防等部门为国家和人民的安全提供了一种预防手段；在机器人领域，智能视频监控系统使机器人能够自主识别并追踪特定目标，增强机器人的自主性，在机器人导航中能起到至关重要的作用；在安防领域，智能视频监控系统在银行、学校、居民社区等公共场所的实时监控、远程预警等方面有着出色地应用；在交通领域，智能视频监控系统在车辆实时检测与跟踪、车牌检测、违章检测以及车流量、人流量和道路占用率等方面有着非常广泛地应用。

运动目标的检测与跟踪是智能视频监控系统中两个关键性的问题，其结果的好坏对整个智能视频监控系统有着至关重要的影响。智能视频监控系统主要是通过检测算法将运动目标与背景分离，同时尽可能完整的提取出目标信息，然后对检测出的运动区域进行目标分类，将后续步骤中感兴趣的部分（例如车辆、行人等）分类出来，然后利用跟踪算法对检测出的感兴趣目标进行跟踪。

智能视频监控技术在过去的几十年内发生了翻天覆地的变化，同时取得了丰硕的成果，但是由于运动目标自身以及背景环境的复杂性，严重影响了检测及跟

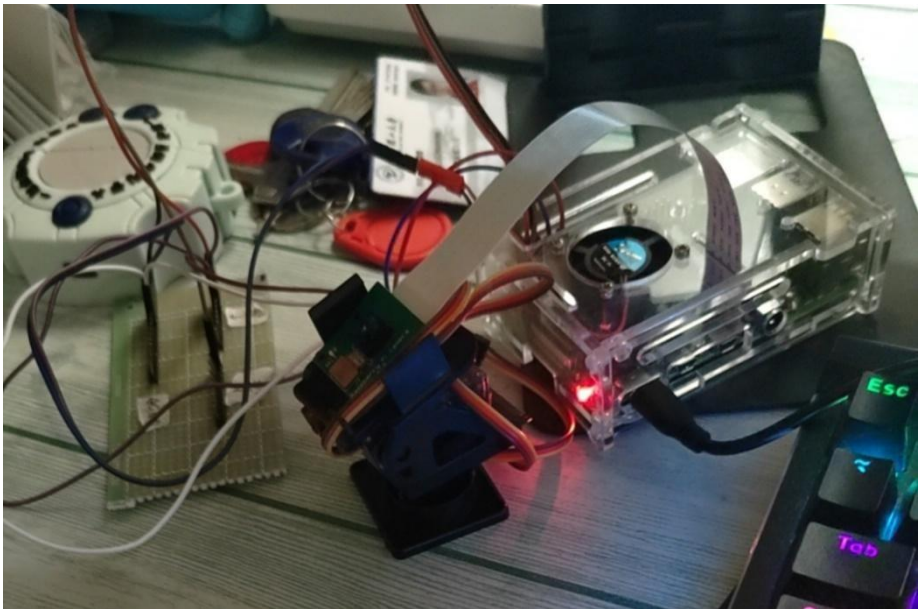
踪结果的准确性，因此，复杂场景下的运动目标检测与跟踪仍然存在着巨大的挑战。

综上所述，智能视频监控系统中的核心环节就是运动目标的检测，其结果对目标跟踪的精准度有着至关重要的影响。同时，复杂场景下的目标本身的变化也是该技术面临的一项挑战。因此，对于运动目标的检测与跟踪的研究有着非常重要的实用价值和学术意义。

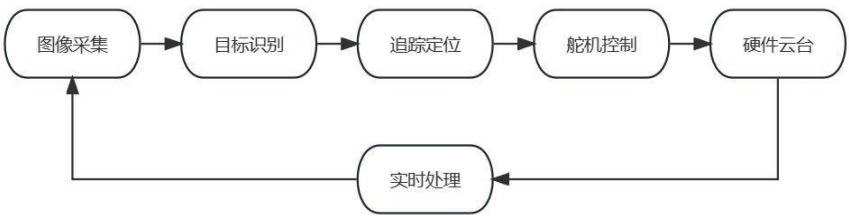
## 2 研究内容、总体方案及原理

本研究的主要内容是设计和开发一个智能视频监控系统，该系统能够自动识别和跟踪指定的目标物体。为了实现这一目标，研究涉及到以下关键内容：

实物图：



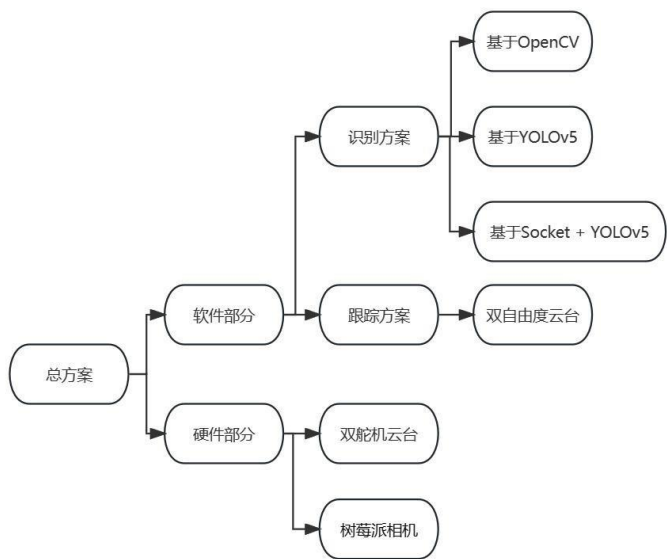
主要硬件包括：树莓派（主控）、Rpi Camera V2（相机）、两个 SG 90S（双自由度云台），通过以下流程进行智能识别跟踪：



1. 硬件云台：通过使用两个 SG 90S 舵机组成双自由度的云台，分别控制相机在水平和垂直方向上的运动。这个云台是实现目标跟踪的物理基础。

- 2. 图像采集：使用树莓派与 RPi Camera V2 相机进行图像的实时采集，这些图像将作为目标识别的输入数据。
- 3. 目标识别：应用计算机视觉技术，设计目标识别算法，将用于在图像中检测和识别目标，这些结果将作为追踪定位的输入数据。
- 4. 追踪定位：应用小孔成像以及目标与中心偏的夹角关系，设计跟踪定位算法，并将这些结果作为舵机控制的输入的数据。
- 5. 舵机控制：根据目标物体在图像中的位置信息，来调整云台的方向，确保相机能够持续跟踪目标。
- 6. 实时处理：将硬件云台、图像采集、目标识别、追踪定位和舵机控制整合成一个实时系统，能够实时捕获图像、识别目标并自动调整相机的方向。

总方案：



识别方案：

方案一：基于 OpenCV

方案一采用了基于 OpenCV 的目标识别。它主要依赖于颜色分离和形状检测，具体包括以下步骤及原理：

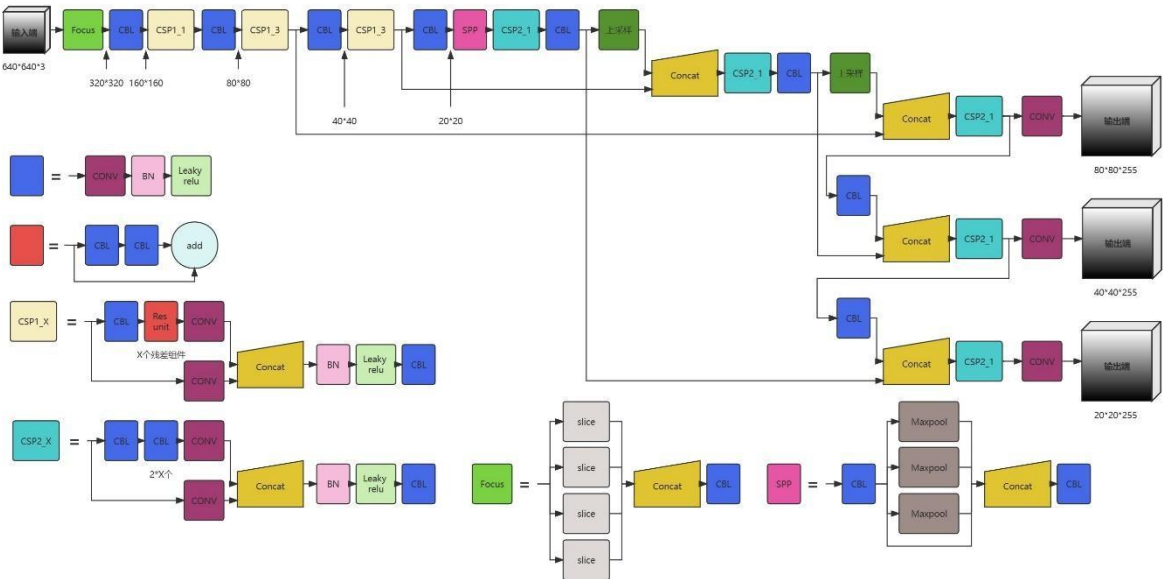
- 1. 颜色分离：通过转化为某种颜色空间后，识别某种颜色来增强识别的准确性，如将图像转换为 HSV 颜色空间，利用 HSV 颜色空间的 H 层来区分颜色，以排除一定程度的环境干扰。

2. 边缘检测：在边缘部分，像素值出现“跳跃”或者较大的变化，如果在此边缘部分求一阶导数，就会看到极值的出现，而在一阶导数为极值的地方，二阶导数为 0，基于这个原理，就可以进行边缘检测。

3. 霍夫圆检测：将二维图像空间中一个圆转换为该圆半径、圆心横纵坐标所确定的三维参数空间中一个点的过程，因此，圆周上任意三点所确定的圆，经 Hough 变换后在三维参数空间应对应一点。该过程类似于选举投票过程，遍历结束后，得票数最高点所确定的圆即为该圆周上，绝大多数点所确定的圆（以下称为当选圆），即绝大多数点均在该当选圆的圆周上，以此确定该圆。

方案二：基于 YOLOv5

方案二采用了基于 YOLOv5 的目标识别方法，其中 YOLOv5 的原理如下：



YOLOv5 (You Only Look Once v5) 是一种端到端的目标检测算法，它通过一张输入图像，直接输出图像中所有目标的类别和位置信息。

YOLOv5 的网络结构采用了 CSPDarknet53 作为主干网络，通过多层卷积和池化操作，将输入图像逐步降采样到一定尺度，然后在这个尺度上进行目标检测。具体来说，它将输入图像分成多个网格，每个网格预测一个边界框，每个边界框可以预测多个类别的概率。

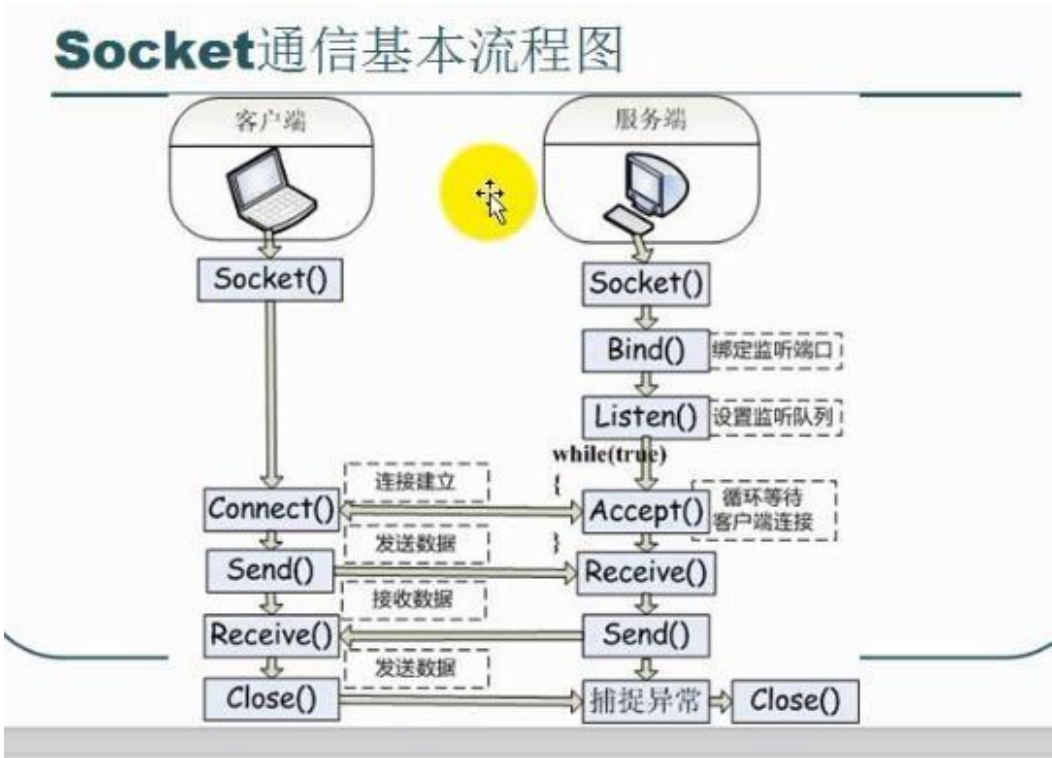
在每个网格中，YOLOv5 预测一个边界框，该边界框由 4 个坐标值和一个置信度值组成。4 个坐标值分别表示边界框的中心点坐标、宽度和高度，置信度值表示该边界框内是否存在目标。同时，每个网格还预测多个类别的概率，这些类

别包括了数据集中的所有目标类别。最终，YOLOv5 采用非极大值抑制（NMS）算法，剔除重叠的边界框，输出目标类别和位置的最终结果。

YOLOv5 的训练过程采用交叉熵损失函数，并结合了类别平衡和焦点损失等技术，使得网络更加稳定和准确。此外，YOLOv5 还采用了数据增强、多尺度训练等技术，增加了数据量和多样性，提高了网络的泛化能力和鲁棒性。

**方案三：基于 Socket + YOLOv5**

方案三结合了 YOLOv5 和 Socket 技术，以实现高速目标识别。其中 Socket 的原理及步骤如下：



服务器端：

1. 创建一个用于监听连接的 Socket 对象。
2. 用指定的端口号和服务器的 ip 建立一个 EndPoint 对象。
3. 用 Socket 对像的 Bind() 方法绑定 EndPoint。
4. 用 Socket 对像的 Listen() 方法开始监听。
5. 接收到客户端的连接，用 Socket 对象的 Accept() 方法创建一个新的用于和客户端进行通信的 Socket 对象。
6. 通信结束后关闭 socket。

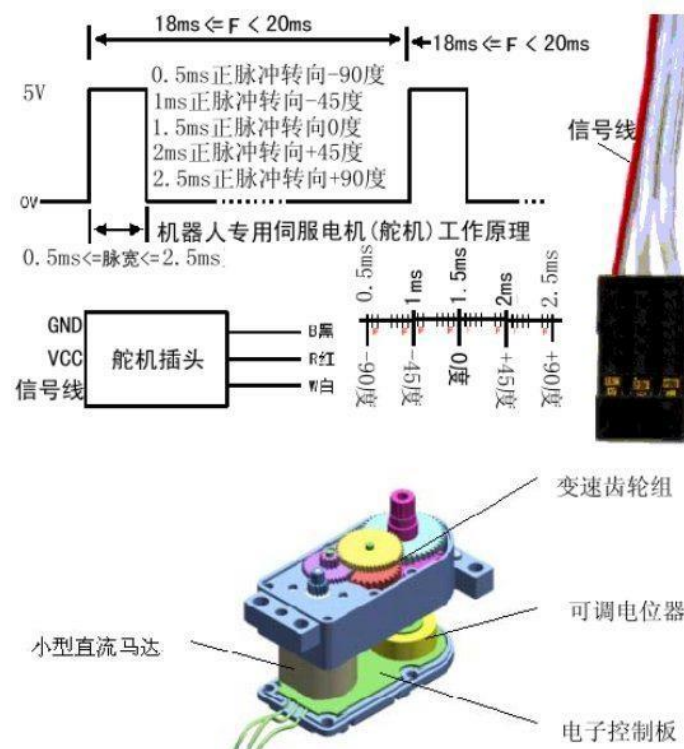


客户端：

1. 建立一个 Socket 对象。
2. 用指定的端口号和服务器的 ip 建立一个 EndPoint 对象。
3. 用 Socket 对象的 Connect() 方法以上面建立的 EndPoint 对象做为参数，向服务器发出连接请求。
4. 如果连接成功，就用 Socket 对象的 Send() 方法向服务器发送信息。
5. 用 Socket 对象的 Receive() 方法接受服务器发来的信息。
6. 通信结束后关闭 Socket。

跟踪方案：

本方案使用了双舵机组成双自由度的云台，在识别的同时同步转动，即可实现小球跟踪，其中使用的 SG 90S 舵机的原理如下：



SG 90S 舵机是基于直流电机的转动原理。它由一个直流电机、一个减速齿轮箱和一个电路控制芯片组成。当输入电压施加在直流电机上时，电机会产生旋转力矩。通过减速齿轮箱的作用，将电机的转速降低，并将输出扭矩增大。在 SG 90S 舵机中，减速齿轮箱通常采用塑料材料制成，齿轮数量也比较少，因此舵机的输出扭矩相对较小，但也可以达到  $1.8\text{kg/cm}$  左右。

在舵机的电路控制芯片中，会根据输入的 PWM 信号生成对应的脉宽，控制舵机的转动角度和速度。通常情况下，PWM 周期为 20ms，脉宽范围为 0.5ms 到 2.5ms。其中，0.5ms 对应舵机的 0 度，2.5ms 对应舵机的 180 度，1.5ms 对应舵机的中间位置。

需要注意的是，SG 90S 舵机是一种小型舵机，因此其输出扭矩和精度可能不如大型舵机，而舵机在运行过程中也会产生一定的温度漂移和寿命问题。因此，在使用时需要谨慎设计控制电路，以保证系统的性能和稳定性。

### 3 研究方法

识别方法：

方案一：基于 OpenCV

1. 颜色分离（HSV）：首先，将图像从 RGB 色彩空间转换为 HSV 色彩空间。然后，通过提取 H 通道，识别目标物体的颜色，从而排除一定程度的环境干扰。
2. 边缘检测（Canny 算子）：使用 Canny 边缘检测算子，对 HSV 图像进行边缘检测，以找到目标物体的轮廓。
3. 霍夫圆检测：应用霍夫圆检测算法，以检测并识别图像中可能存在的球状目标，因为球体在不同方向看都呈现圆形。

方案二：基于 YOLOv5

1. YOLOv5 模型部署：在树莓派上部署 YOLOv5 模型，用于目标识别。
2. 目标识别：在树莓派上通过运行 YOLOv5 模型，对图像中的目标进行识别，获取目标的类别、位置和置信度等信息。

方案三：基于 Socket + YOLOv5

1. YOLOv5 模型部署：在主机上部署 YOLOv5 模型，用于目标识别。
2. Socket 通信：建立 Socket 连接，监听 Socket 连接，将树莓派上捕获的图像传输到主机。
3. 目标识别：在主机上通过运行 YOLOv5 模型，对图像中的目标进行识别，获取目标的类别、位置和置信度等信息。
4. 结果回传：将识别结果通过 Socket 连接回传给树莓派，以实现高速目标识别的闭环控制。

The diagram illustrates the pinhole camera model and its geometric representation using similar triangles.

**Left Diagram: 3D Camera Model**

- 物理成像平面 (Physical Imaging Plane):** The plane where the image is formed.
- 像素平面 (Pixel Plane):** A specific plane within the physical imaging plane.
- 像素坐标系  $O-u-v$  (Pixel Coordinate System  $O-u-v$ ):** A coordinate system centered at the origin  $O$  on the pixel plane, with axes  $u$  and  $v$ .
- 相机 (Camera):** The central part of the model, containing the **光心  $O$  (Optical Center  $O$ )**.
- 焦距  $f$  (Focal Length  $f$ ):** The distance from the optical center  $O$  to the pixel plane.
- 相机坐标系  $O-x-y-z$  (Camera Coordinate System  $O-x-y-z$ ):** A coordinate system centered at the optical center  $O$ , with axes  $x$ ,  $y$ , and  $z$ .
- 点  $P$  (Point  $P$ ):** A point in the 3D scene.
- 点  $P'$  (Point  $P'$ ):** The projection of point  $P$  onto the pixel plane.

**Right Diagram: Similar Triangles**

- 相似三角形 (Similar Triangles):** This diagram shows the geometric relationship between the 3D point  $P$  and its projection  $P'$  on the image plane.
- 点  $P$  (Point  $P$ ):** The 3D point in space.
- 点  $P'$  (Point  $P'$ ):** The projection of point  $P$  onto the image plane.
- 光心  $O$  (Optical Center  $O$ ):** The center of the camera.
- 焦距  $f$  (Focal Length  $f$ ):** The distance from the optical center  $O$  to the image plane.
- 距离  $Z$  (Distance  $Z$ ):** The distance from the optical center  $O$  to the point  $P$ .
- 点  $A$  (Point  $A$ ):** The projection of point  $P$  onto the image plane.
- 点  $B$  (Point  $B$ ):** The projection of point  $P$  onto the image plane.
- 点  $X$  (Point  $X$ ):** The projection of point  $P$  onto the image plane.
- 点  $X'$  (Point  $X'$ ):** The projection of point  $P$  onto the image plane.

依次控制 x 轴和 y 轴的 SG 90S 舵机旋转来调整云台的方向，确保相机中心时刻对准跟踪目标，降低跟踪丢失的可能，实现持续跟踪。

识别结果:

优点:

1. 识别速度非常快，几乎实时。
2. 对于简单背景下的目标识别表现良好。

1. 受环境干扰较为严重，容易受到光线变化和背景干扰的影响，导致误识别。

- ### 方案二：基于 YOLOv5

测试数据:



预训练集	识别时长	准确度
yolov5x.pt	12.46s	94.97%
yolov5l.pt	7.24s	95.68%
yolov5m.pt	4.09s	97.90%
yolov5s.pt	1.87s	97.40%
yolov5n.pt	0.99s	96.21%

数据分析：

容易发现，当模型量级降低时，识别时长显著降低，且对于小球识别而言，精确度差别不大，故而选择 yolov5n.pt 模型更优，但一次识别需要 0.99s 仍旧太慢，需要考虑更好的优化方案。

优点：

- 1. 目标识别非常准确，能够检测多个类别的目标。
- 2. 对于不同场景和复杂背景下的目标具有较好的鲁棒性。

缺点：

- 1. 识别速度相对较慢，通常每秒只能处理一帧图像，无法实现实时跟踪要求。
- 2. 在树莓派等低功耗设备上运行时，可能存在性能限制。

方案三：基于 Socket + YOLOv5

根据方案二的结果可知 yolov5n.pt 模型在识别速度上具有显著优势，且精确度差别不大，故本方案决定在 yolov5n.pt 模型上进行测试。

测试数据：

预训练集	总时长		准确度
	传输时长	识别时长	
plan2	0.99s		96.21%
640	0.0951s		95.80%
	0.0127s	0.0824s	
480	0.07794s		95.40%
	0.0083s	0.06964s	
320	0.07025s		94.80%
	0.00548s	0.06477s	

数据分析：

当使用方案三后可以发现对于同样 640 \* 640 的图片，方案三提升了 10 倍左右，但也只是到达了每秒 10 帧，我们可以考虑降低图像的大小，发现 320 \* 320 的模型识别速度更快，但在实际应用中发现，320 \* 320 的模型因图像过小导致图像不清晰，学习出来的模型很容易误识别，而 480 \* 480 的模型与 320 \* 320 的模型相比，识别总时长差别不大，但不容易误识别，故而选择使用 480 \* 480 的模型。

优点：

1. 识别速度较快，通常在每秒 10 到 15 帧之间，具有实时性。
2. 识别精准，能够在不同场景下准确识别目标。

缺点：

1. 需要额外的硬件和通信开销，包括主机和树莓派之间的网络通信。
2. 对于一些实时性要求极高的应用，可能仍然需要进一步优化以提高识别速度。

综合而言，每个方案都有其优点和局限性。方案一在速度方面表现出色，但受环境干扰较大，适用于简单的应用。方案二在识别准确性方面表现出色，但速

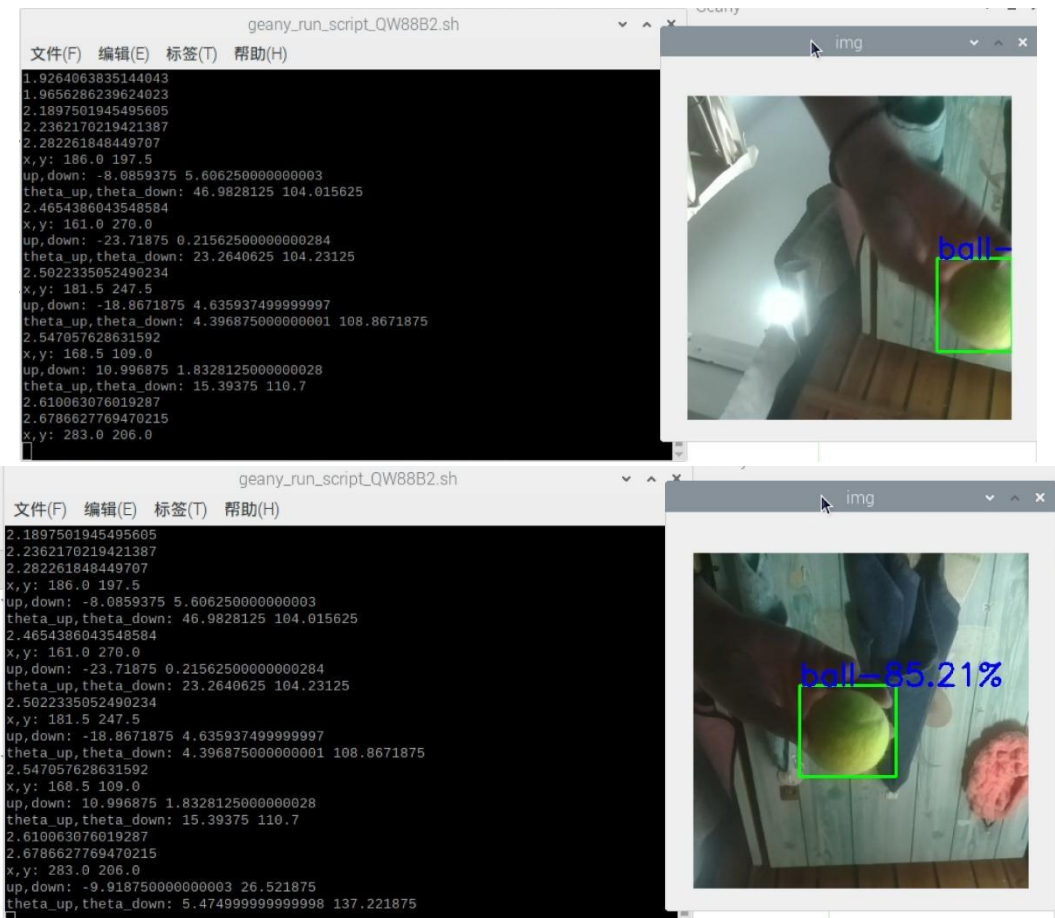
度较慢，适合对准确性要求较高但不需要实时响应的应用。方案三在速度和准确性方面取得了较好的平衡，适用于需要实时目标识别和跟踪的应用场景。在经过综合考虑后，决定选择方案三。

**跟踪结果：**

结果与计算基本吻合，只要目标不在识别间隔内移出镜头可视范围（即移动速度过快），均可以实现实时追踪。但因为其精度过低，在物体运动时图像中心与物体中心存在细微偏差，但仍在误差范围内，不会影响跟踪目标。

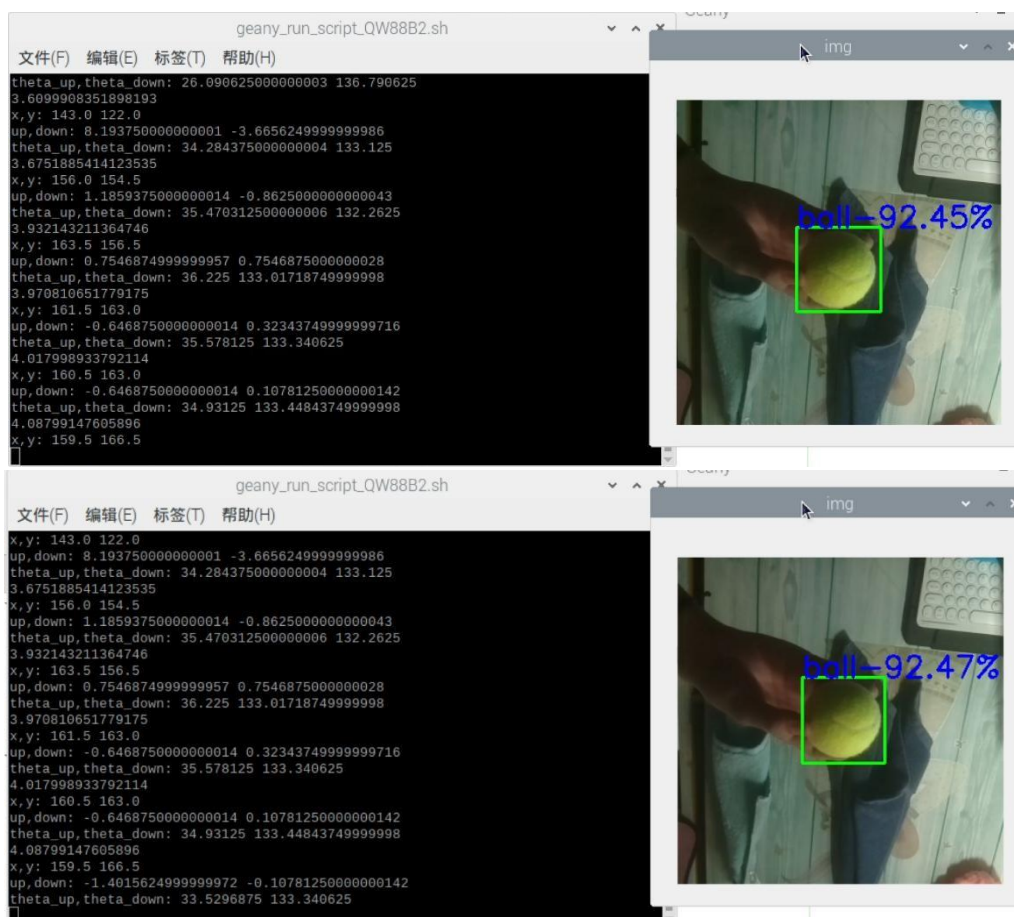
**最终结果：**

**运动：**



图中为运动时连续的两次识别，均可做到精准识别，并实时跟踪，假设后续将不进行移动，可以根据后续位置进行精准矫正，若后续将继续移动，则继续实时追踪。

**静止：**



图中为静止时连续的两次识别，均可做到精准识别，并将物体矫正至中心，以此来对物体向各个方向运动后进行识别留有空间。

## 5 参考文献

- [1] Glenn Jocher, Ayush Chaurasia, ... Mrinal Jain. (2022). ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation (v7.0). Zenodo. <https://doi.org/10.5281/zenodo.7347926>
- [2] 杨波. 视频图像序列中运动目标检测与跟踪算法的研究[D]. 兰州理工大学, 2020. DOI:10.27206/d.cnki.ggsgu.2020.000902.
- [3] 王新. 复杂场景下运动目标检测与跟踪技术的研究[D]. 江苏大学, 2018.
- [4] 马睿, 王荣, 魏峻超. 一种基于树莓派和 Socket 通信的视频小车设计方案[J]. 电脑编程技巧与维护, 2020(12):129-130. DOI:10.16184/j.cnki.comprg.2020.12.050.
- [5] 王博文, 宗碧, 雷铠伊等. 智能监控摄像头的研究——基于 Dlib、OpenCV 和双舵机云台摄像头[J]. 现代信息科技, 2020, 4(07):89-91+94. DOI:10.19850/j.cnki.2096-4706.2020.07.029.

- [6] Assemahagn F S. 艺术体操中小球的检测与跟踪研究[D]. 西安电子科技大学, 2021. DOI:10.27389/d.cnki.gxadu.2021.000046.
- [7] 田小锋. 基于二轴云台的目标检测与跟踪[D]. 长安大学, 2023. DOI:10.26976/d.cnki.gchau.2022.000902.
- [8] 何惠彬, 江亚俊, 肖晓晖. 基于 OpenCV-Python 和树莓派实现图像数据远程采集[J]. 福建农机, 2022(03):27-30.
- [9] 肖小龙. 基于区域的图像分割及视觉显著性检测算法研究[D]. 华东理工大学, 2014.
- [10] 张倩. 基于深度学习的图像显著性区域检测技术研究[D]. 山东师范大学, 2020. DOI:10.27280/d.cnki.gsdsu.2020.000466.
- [11] 秦云凡. 基于凸包的图像显著性区域检测[D]. 河北科技大学, 2021. DOI:10.27107/d.cnki.ghbku.2020.000534.
- [12] 赵恒逸. 二轴云台的目标跟踪技术研究[D]. 南京航空航天大学, 2020. DOI:10.27239/d.cnki.gnhhu.2019.000693.