

智能网球检测系统

黄勖、徐安妮

2025 年 6 月 8 日

目录

- ① 项目概述
 - 主要功能
- ② 技术架构
- ③ 技术实现细节
 - 传统检测方法
 - YOLO 深度学习检测
 - 混合检测策略
- ④ 关键问题及解决方案
- ⑤ 使用指南
- ⑥ 性能指标
- ⑦ 项目文档说明
- ⑧ 未来改进方向

项目概述

- **目标:** 开发一个高精度的网球检测系统，主要用于机器人捡球场景。
- **能力:** 在各种光照条件和复杂背景下准确识别网球，处理多个网球。

主要功能

- 单张图片网球检测
- 支持多种检测方法（传统图像处理、深度学习）
- 适应不同光照和场地条件

技术架构

系统采用混合检测策略，结合传统计算机视觉和深度学习方法的优点。

- **传统检测模块** (`tennis_detection.py`): 基于颜色和形状。
- **YOLO 检测模块** (`yolo_singleton.py`): 基于 YOLOv5, 单例模式, 本地缓存。
- **混合检测模块** (`hybrid_detection.py`): 结合两者优势, 自适应选择。
- **主处理模块** (`process.py`): 统一接口, 错误处理。

传统检测方法

- ① 预处理: 图像增强、噪声去除。
- ② 颜色分割: HSV 颜色空间提取特征颜色。
- ③ 形态学操作: 去除噪点、连接断裂区域。
- ④ 轮廓分析: 基于面积、圆度等特征过滤。
- ⑤ 连接网球分离: 距离变换和分水岭算法。

YOLO 深度学习检测

基于 YOLOv5 预训练模型进行优化:

- **模型选择:** YOLOv5s。
- **单例模式:** 避免重复加载, 提高效率。
- **本地缓存:** 模型本地化, 避免重复下载。
- **设备优化:** 优先 CPU, 提高兼容性。
- **错误处理:** 完善的异常捕获。

混合检测策略

组合两种方法的优点:

- 优先 YOLO: 网络条件允许时首选。
- 自动切换: YOLO 失败时切换到传统方法。
- 结果合并与去重: 提高准确率。
- 置信度评估: 选择最终结果。

关键问题及解决方案

- **YOLO 模型重复下载:**
 - 解决方案: 单例模式、本地缓存、全局状态记录。
- **SSL 证书验证错误:**
 - 解决方案: 配置 SSL 上下文、增加超时、错误处理。
- **NumPy 类型 JSON 串行化:**
 - 解决方案: 转换为标准 Python 类型、显式转换。
- **网球连接问题:**
 - 解决方案: 分水岭算法、形状后处理、YOLO 结果增强。

使用指南

环境准备:

```
pip install -r requirements.txt
```

单张图片检测示例:

```
from process import process_img

results = process_img("path/to/image.jpg")
print(results)
```

性能指标

- **平均检测时间:** 70-100 毫秒/张 (使用缓存模型)
- **准确率:** >90
- **召回率:** >85
- **误检率:** <5

项目文档说明

- `process.py`: 主处理入口 (推荐使用)
- `yolo_singleton.py`: 优化后的 YOLO 单例实现
- `src/tennis_detection.py`: 传统方法检测
- `src/hybrid_detection.py`: 混合检测算法
- `doc/README.md`: 详细技术文档
- `slide/presentation.tex`: 本演示文稿

未来改进方向

- ① **模型优化**: 使用网球专用数据集微调 YOLO 模型。
- ② **加速检测**: 探索 TensorRT/ONNX 等加速方案。
- ③ **多目标跟踪**: 添加目标跟踪功能。
- ④ **距离估计**: 结合相机参数估计网球距离。
- ⑤ **边缘计算优化**: 针对嵌入式平台优化。

谢谢观看！

提问与交流