Player Re-Identification in Soccer Footage

Assignment Context

Company: Liat.ai (Stealth Mode)
Role: AI Intern — Computer Vision & Sports Analytics

Objective:
To implement a solution for player re-identification in soccer videos — ensuring each player retains a consistent ID even when they leave or re-enter the frame.

Problem Statement

Modern sports analytics requires accurately tracking individual players across video feeds. This is challenging due to:

- Occlusions (players blocking each other)
- Players leaving/re-entering frame
- Similar jerseys/numbers
- Variable camera angles & motion blur

Analytical Goal

- Detect players in each video frame using a pre-trained YOLOv11 model.
- Track players frame-to-frame to maintain consistent IDs.
- Output a video showing bounding boxes + IDs.
- Document limitations & propose next steps for robust re-identification.

Tools & Libraries Used

Tool	Purpose
Python 3.10	Main programming language
Conda	Virtual environment management
Ultralytics YOLOv11	Object detection — player bounding boxes
OpenCV	Video I/O, frame extraction, drawing boxes
SORT (Simple Online Realtime Tracking)	Basic tracker implementation (toy version)
Jupyter Notebook	Experiments & intermediate tests

Tool	**Purpose**
Git/GitHub	Version control & final submission

Data & Model Details

- **Input Video:** 15sec_input_720p.mp4 (15-second soccer clip)
- **YOLO Weights:** best.pt — custom fine-tuned YOLOv11 model trained for player & ball detection.

Implementation Steps

Environment Setup

- Created player_reid Conda environment.
- Installed dependencies: ultralytics, opencv-python, filterpy.

Frame Extraction & EDA

- Extracted sample frames using OpenCV to test YOLO detection.
- Verified detection bounding boxes with model ('sample_frame.jpg', show=True).

Object Detection

- Used YOLOv11 to detect players in each video frame.
- Achieved good detection even with small players or partial occlusions.

Tracking Pipeline

- Integrated a basic version of SORT (Simple Online Realtime Tracking).
- Linked YOLO detections to unique IDs per frame.
- Generated output video tracked_output.mp4.

Jupyter Notebook

- Documented each step, frame extraction, detection outputs.
- Tested the detection model independently before running the tracker.

Results

- Successfully generated a pipeline that detects and tracks players.
- Output video shows bounding boxes with ID labels.
- Pipeline runs in real-time on short clips.
- Demonstrates clear understanding of detection + tracking flow.

Challenges Encountered

Challenge	Description
IDs flip frequently	Basic SORT used does not implement proper Kalman filter or IOU matching logic, so new IDs are assigned when players overlap or move fast.
Occlusions	Players blocking each other caused missed detections.
Low resolution	Some blurry frames reduced detection confidence.

Next Steps & Improvements

If given more time or production goals:

- Integrate a robust tracker like **DeepSORT** or **ByteTrack** for motion + appearance embedding.
- Add **jersey number recognition** to link IDs more reliably.
- Use **vision-language models (VLMs)** for multi-modal matching.
- Handle multiple camera feeds for cross-camera player mapping.
- Optimize latency for real-time inference in live games.

Submission Deliverables

File	Purpose
track.py	Main detection + tracking script
sort.py	Basic tracker implementation
player_reid_notebook.ipynb	Experiments, EDA & test runs
15sec_input_720p.mp4	Input video
tracked_output.mp4	Final output video
README.md	Setup instructions
REPORT.md	This report

Reflections

This project deepened my hands-on skills in:

- Combining detection + tracking pipelines.

- Working with real-time video streams.
- Debugging environment issues (Conda, pip, Git).
- Understanding practical limitations of simple trackers vs. production-grade solutions.

It was a valuable exercise in end-to-end vision pipelines for sports analytics.

Thanks!

Prepared & submitted by: **Poornima KC**

Ready to submit!

I'm excited to discuss this approach further and explore how I can contribute to building real-time sports analytics systems at Liat.ai.