

Liat.ai - Player Re-Identification in Single Feed

Objective:

To develop a real-time system that assigns and maintains consistent IDs to players across frames of a sports video, even after disappearing and reappearing.

Approach & Methodology

1. *Detection* :

- Used a fine-tuned YOLOv11 model ('best.pt') to detect players (class ID = 2).
- Frame-by-frame detection using 'ultralytics' library.

2. *Tracking Logic* :

- Custom-built IoU + centroid-based tracker.
- Each detection is compared with existing tracked players:
- If IoU or center distance is high, reuse ID else assign new ID.

3. *Optimizations* :

- Smoothed bounding boxes by rounding
- Limited max ID to avoid jumping
- Implemented 'max_missing' logic to retain memory of exited players

Techniques Tried

Technique	Outcome
IoU-based Tracking	Worked well for close frames
Center Distance Fallback	Helped re-identify players after occlusion
Bbox Rounding	Reduced flickering and ID instability
Capped ID Range	Solved large ID jump issues

Challenges Faced

- 🚧 NumPy arrays not hashable → led to ID mismatch
- 🚧 Re-identifying missed players was tricky without appearance features
- 🚧 Tracking errors increased if players entered abruptly or in groups

Technologies Used

- Python 3.10+ — Core programming language used
- YOLOv11 (Ultralytics) — Fine-tuned object detection model for players and ball
- OpenCV — Frame-by-frame video processing, drawing bounding boxes and saving output
- NumPy — For numerical operations and bounding box comparisons
- Virtual Environment (venv) — Environment isolation to manage dependencies

Project Summary

This project tackles the problem of player re-identification in sports video footage using computer vision techniques. The goal is to ensure each player is assigned a unique ID that remains consistent throughout the video, even if the player temporarily exits the frame or reappears later. A YOLOv11 object detection model was used to detect players in each frame, and a custom tracker was implemented to match detections across frames using IoU and centroid distance. The tracker maintains a dictionary of active and recently missing players, and uses a combination of spatial proximity and detection confidence to assign IDs accurately. The final system is capable of producing a clear video with stable tracking and ID labels across the full sequence.

Future Work

With more time, we could:

- Add appearance features using colour histograms or embedding-based DeepSORT
- Handle cross-camera player matching
- Track more players (>30) using clustering or smarter memory
- Build a dashboard with analytics like total player count, ID stability, player heatmaps

Status:

Completed for single-camera Re-ID use case

All known bugs resolved, ID tracking stable and capped.

Outcome

The project successfully delivers a functional re-identification system capable of stable player tracking in a single camera feed. It addresses common challenges such as ID flickering, large ID jumps, and re-identification after temporary disappearance. The final version ensures all players receive consistent, bounded IDs and maintains visual clarity in the output video.

GitHub link:

https://github.com/Samartha21BRS1698/Player_re-identification_liat-ai

Author

Samartha

samu36939@gmail.com

[LinkedIn](#)

[GitHub](#)