Player Re-Identification in a Single Feed

Liat.ai – AI Intern Assignment Report

Kirtiraj Jamnotiya
kjamnotiya@gmail.com — https://linkedin.com/in/kjamnotiya

June 9, 2025

Overview

This report presents a complete solution to the problem of **Player Re-Identification in a Single Camera Feed** using object detection and tracking techniques. The goal is to ensure that each player retains a consistent identity (`player_id`) throughout the video, even when they leave and re-enter the frame.

Video: 15-second clip (720p) provided in the dataset.

Approach

Our pipeline consists of two main stages:

Step 1: Player Detection using YOLOv11: A fine-tuned YOLOv11 model detects players in each frame.

Step 2: Tracking and Re-identification using DeepSORT: DeepSORT assigns and maintains consistent IDs across frames using motion and appearance cues.

The overall system simulates a real-time player tracking scenario, ensuring smooth identity preservation even under partial occlusion or temporary disappearance.

Methodology

1. Detection

We use the provided YOLOv11 model, pre-trained to detect players and the ball. The model outputs bounding boxes and class labels per frame.

- Detection Confidence Threshold: `0.4`
- Classes used: `player (class 0)`, `ball (class 1)`

2. Tracking with DeepSORT

We integrated the `deep_sort_realtime` library for robust multi-object tracking. DeepSORT leverages both motion (Kalman Filter) and appearance (cosine similarity of embeddings) for assigning identities.

Key features:

- Appearance feature extractor: CNN-based embeddings
- Distance metric: cosine distance
- Tracks are updated across frames using Hungarian Algorithm

3. ID Consistency Logic

Each new player entering the frame is assigned a unique ID. DeepSORT maintains identity even when a player leaves the frame and reappears. For evaluation, we visually validate that players receive the same ID throughout their trajectory.

Implementation Details

- Framework: Python 3.10
- Libraries: `ultralytics`, `opencv-python`, `deep_sort_realtime`, `numpy`, `matplotlib`
- Input: `data/15sec_input_720p.mp4`
- Output: `output/result.mp4`
- Environment setup: `create_env.sh` (conda)

Results

The output video demonstrates consistent player IDs throughout the clip.

- **Green** bounding boxes indicate detected players.
- **Blue** labels display the player ID (`ID:<number>`).
- Re-appearance is correctly handled by DeepSORT in most cases.

Challenges Faced

- **Occlusion:** Short-term occlusions sometimes confused the tracker.
- **Similar Appearance:** Players with identical jerseys increased ID switching chances.
- **Low Light Conditions:** Some frames had insufficient contrast for reliable tracking.

What Could Be Improved

- Incorporate re-ID embeddings using a fine-tuned person ReID model.
- Train a stronger custom YOLOv11 model using additional team-specific sports data.
- Fuse temporal information across frames using optical flow or 3D ConvNets.
- Deploy post-processing to smooth out ID switches (e.g., using Kalman-IOU logic).

Conclusion

This project effectively solves real-time player re-identification using YOLOv11 + DeepSORT. The modular design enables easy extension to cross-camera setups or longer videos. Further work could explore advanced appearance models and cross-view matching for complete robustness.

Acknowledgements

- **YOLOv11:** Ultralytics fine-tuned model for detection
- **DeepSORT:** Reliable and efficient multi-object tracker
- **Liat.ai:** For the opportunity and dataset