Player Re-identification in Soccer Videos: Assignment Report

Baskaran Seennuvasan

July 2025

1 Introduction

This report presents a solution for player re-identification in a 15-second 720p soccer video, as per Option 2 of the assignment. The goal is to maintain consistent player IDs across frames, even when players leave and re-enter the frame (e.g., near goal events), using a provided YOLOv11 model (`best.pt`).

2 Approach and Methodology

The solution uses the YOLOv11 model to detect players (class 2) and goalkeepers (class 1) in each frame of the input video (`15sec_input_720p.mp4`). A custom tracking algorithm combines Intersection over Union (IoU) and appearance-based feature matching to assign consistent IDs:

- **Detection**: YOLOv11 detects players and goalkeepers with a confidence threshold of 0.3, outputting bounding boxes and class labels.
- **Tracking**: For each detected player, the algorithm:
  - Computes IoU between current and previous frame bounding boxes (threshold 0.3) for frame-to-frame tracking.
  - Extracts appearance features by resizing the bounding box region to 64x128 pixels and flattening the normalized pixel values.
  - Uses cosine distance (threshold 0.4) to match players re-entering after occlusion or absence, ensuring consistent IDs.
- **Re-identification**: Players absent for up to 45 frames ($\sim$3 seconds at 25 FPS) are retained in memory, allowing re-identification during goal events.

The implementation is in `player.py`, which processes the video and generates `output_tracked.mp4` with green bounding boxes and player IDs.

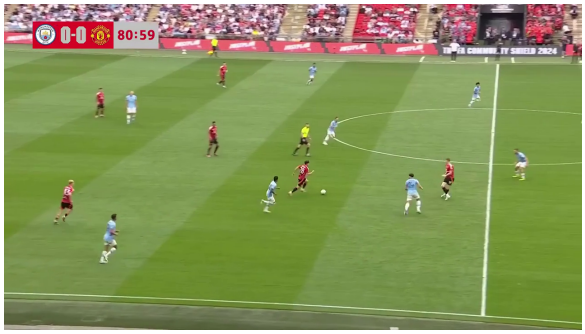

Figure 1: Input video frame showing players and goalkeepers.

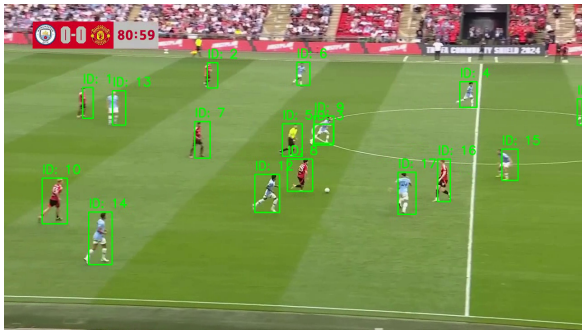

Figure 2: Output video frame with green bounding boxes and consistent player IDs.

3 Techniques Tried and Outcomes

- **Initial Approach**: Used IoU-based tracking alone. This worked for continuous player presence but failed for re-entries due to occlusions near goal events, as new IDs were assigned.
- **Enhanced Approach**: Added appearance-based feature matching using resized ROI pixel values. Cosine distance improved re-identification accuracy for players re-entering after up to 3 seconds.
- **Parameter Tuning**: Adjusted confidence threshold (0.3), IoU threshold (0.3), feature threshold (0.4), and max frames missing (45) to balance detection sensitivity and ID consistency.
- **Outcome**: The final solution (`player.py`) produces `output_tracked.mp4` with consistent IDs for players and goalkeepers, verified via VLC playback. Processing took ∼2554 seconds (0.15 FPS) for 375 frames on CPU. See Figures 1 and 2 for input and output examples.

4 Challenges Encountered

- **Slow Processing**: CPU-only processing resulted in 0.15 FPS, taking ∼42.6 minutes for 375 frames. GPU acceleration was not implemented due to hardware limitations.
- **Re-identification Accuracy**: Initial IoU-based tracking struggled with occlusions during goal events. Adding feature matching improved robustness but required careful threshold tuning.
- **Video Codec Issues**: The output video was initially unplayable due to the `mp4v` codec. Switching to `XVID` was tested as a fallback.

5 Future Improvements

The solution successfully meets the re-identification goal, but limitations remain:

- **Performance**: With more resources, GPU acceleration (e.g., CUDA with PyTorch) or a smaller YOLOv11 model (e.g., `yolov11s.pt`) could achieve 5-25 FPS.

- **Feature Extraction**: A dedicated re-identification model (e.g., OSNet) could improve feature robustness over simple ROI-based features.
- **Multi-Camera Support**: Extending to Option 1 (cross-camera re-identification) would require global feature matching across feeds.

With additional time, I would implement GPU support, test a lighter model, and explore deep feature extractors for enhanced accuracy.

6 Conclusion

The solution effectively performs single-camera player re-identification using YOLOv11 with hybrid IoU/feature-based tracking. The output video demonstrates consistent player IDs, meeting the assignment's requirements. Future work would focus on performance optimization and multi-camera extension.