Player Re-identification Assignment Report

Project Overview

This project is aimed at building a robust Player Re-Identification system using a single video feed. The goal is to track and assign consistent IDs to players even when they leave and re-enter the frame. The system uses:

- YOLOv8 for player detection
- Norfair for tracking across frames
- OpenCV for visualization and video processing

All code and resources are shared via GitHub: PlayerTracking-YOLOv8

Approach and Methodology

1. Player Detection

- Used YOLOv8n pretrained model for lightweight, real-time performance.
- Detected players in each frame and extracted bounding boxes.

2. Tracking with Norfair

- Transformed each bounding box into a center point (x, y).
- Fed these points into Norfair's tracker with a custom distance function (Euclidean).
- Assigned unique IDs to each detected player.

3. Visual Annotations

- Drew bounding boxes and center dots for each player.
- Overlayed consistent colors for each unique ID.
- Displayed an in-frame match timer (MM:SS).

4. Output Generation

- Created annotated video using OpenCV.
- Saved tracking log (tracking_log.csv) with frame, id, cx, cy.

Techniques I Tried & Outcomes

Technique Tried	Outcome
Using YOLOv8n for detection	Fast inference, good enough for small-scale test videos.
Norfair tracking with center points	IDs remained consistent even after partial occlusion.
FPS Tuning and smoothing	Helped sync detection frame rate and video playback.
Random color mapping for each ID	Visually helped identify players clearly.

Challenges Encountered

1. Re-identification After Re-entry:
    - Difficult to maintain the same ID if the player re-enters after long occlusion.
    - Norfair may create a new ID due to the large motion gap.
2. Low Confidence Detections:
    - YOLO occasionally missed players in motion blur.
    - Could be improved with better model or frame pre-processing.
3. Tracking Accuracy:
    - False positives in background objects during certain frames.
    - Limited by model size (used YOLOv8n due to Colab constraints).
4. Google Colab Storage/Memory:
    - Frame-by-frame processing needed optimization to avoid timeouts.

If I Had More Time / Resources...

- Use a Stronger YOLO Variant (YOLOv8m/l) for better accuracy.
- Train a Re-ID embedding model to track players based on appearance.
- Add smoothing filters for better bounding box stability.
- Use DeepSort instead of Norfair for feature-based tracking.
- Implement Multi-Camera Setup for cross-angle identification.

Final Output

- Input Video: 15sec_input_720p.mp4
- Code Notebook: Players_TrackingCode.ipynb
- Output Video: player_tracking_output.mp4
- Log File: tracking_log.csv

Repo: https://github.com/SasiRekhaSadanala/PlayerTracking-YOLOv8