Report: Player Re-Identification in a Single Camera Feed

1. Overview

The goal of this project was to develop a real-time solution for **player re-identification in a single video feed**, simulating a real-world sports analytics challenge. The system must ensure that each player is assigned a unique ID, and that players retain the **same identity** even after temporarily going out of frame.

---

2. Objective

- Input: A 15-second football video (`15sec_input_720p.mp4`)
- Task: Detect and assign unique IDs to players
- Key Requirement: Maintain **consistent player IDs** even after occlusions or exits

---

3. Approach and Methodology

3.1 Detection

- Used a fine-tuned **YOLOv11 model** trained to detect both players and the ball.
- Only player detections were retained by filtering the correct class ID (class 1).

3.2 Appearance-Based Re-Identification

- Extracted a **color histogram (RGB)** from each detected player's bounding box.
- Normalized the patch size to (64x128) before computing the histogram.
- Used **cosine similarity** to compare histograms.
- If similarity to an existing ID exceeded a threshold, the same ID was reassigned.
- Otherwise, a new ID was generated.

---

4. System Components

- `detector.py`: Loads the YOLO model and performs player detection + histogram extraction
- `reid.py`: Manages the identity assignment using histogram similarity

- `main.py`: Ties detection and re-ID into a real-time video loop

---

5. Challenges Encountered

- **Class Filtering**: Initial model mistakenly tracked the ball due to wrong class ID.
- **Occlusion Sensitivity**: Histograms sometimes failed to differentiate players with similar jerseys.
- **Real-time Matching**: Balancing histogram sensitivity and false ID assignments required tuning.
- **Out-of-Frame Players**: If a player re-entered with changed lighting or pose, matching sometimes failed.

---

6. Techniques Tried and Outcomes

Technique	Outcome
YOLOv11 detection	Reliable and fast player detection
RGB histogram	Lightweight but less discriminative on similar outfits
Cosine similarity	Simple, effective, but sensitive to appearance variation

---

7. Improvements with More Time

- Replace RGB histograms with **deep Re-ID embeddings** (e.g. OSNet, ResNet50)
- Save **track histories** (e.g. positions + appearance) to improve re-matching
- Add **temporal smoothing** and Kalman filters for better ID stability
- Train on **team jersey embeddings** to reduce ID flips

---

8. Conclusion

The project demonstrates a working real-time player re-identification system using classical vision techniques. It shows that even lightweight appearance matching methods like histograms can be effective when paired with strong detection. While not perfect, it serves as a solid base for more advanced tracking solutions.

---

9. References

- Ultralytics YOLOv11: https://github.com/ultralytics/ultralytics
- OpenCV Histogram Docs: https://docs.opencv.org/4.x/d1/db7/tutorialpyhistogram_begins.html

- Scipy Cosine Distance: https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cosine.html