📝 Player Re-Identification – Brief Report

👤 Submitted by

Dibyajyoti Dutta
Role: AI Intern Assignment – Liat AI

☑ Task Chosen

Option 2: Re-Identification in a Single Feed

The goal was to detect and assign consistent IDs to players appearing in a 15-second video clip, even if they temporarily go out of frame and reappear later.

🔧 Approach and Methodology

1. Player Detection

- Used a YOLOv11-based model (best.pt) provided in the assignment.
- Detected all objects in the video, filtered only class 0 (players).
- Detections were performed frame-by-frame using the Ultralytics library.

2. Tracking and ID Assignment

- Implemented a simple re-identification tracker using Intersection over Union (IoU).
- For each detected bounding box:
  - Compared it to all previous players.
  - If IoU > 0.3, considered it the same player and reassigned previous ID.
  - Otherwise, assigned a new unique ID.
- Stored color and ID per player for visual consistency.
- Tracked players across frames using a last_seen timer (15-frame memory).

3. Output Generation

- Annotated each frame with bounding boxes and IDs using OpenCV.
- Final output was saved as output.avi using cv2.VideoWriter().

🧪 Techniques Tried

- Basic IoU tracking: Worked well for short videos with moderate player movement.
- Color-coded tracking: Helped visually differentiate re-identified players.

- YOLO confidence thresholding: Improved stability by ignoring low-confidence detections.

⚠️ Challenges Encountered

- No built-in tracking model: Used custom IoU logic instead of DeepSORT due to time/resource constraints.
- Partial occlusion and crossing players: Sometimes resulted in ID swaps.
- Google Colab limitations: GUI functions (cv2.imshow, cv2.waitKey) were not supported; used workarounds.

🧩 Incomplete Parts / Future Improvements

If given more time or resources, I would:

- Integrate **DeepSORT** or **ByteTrack** for more robust tracking.
- Extract **appearance features** (e.g., color histograms or embeddings) to aid in re-identification.
- Fine-tune the YOLO model further for tighter player boxes.
- Support real-time inference or web deployment.

🙌 Final Note

I thoroughly enjoyed working on this assignment. The open-ended nature gave me the freedom to think creatively and focus on practical implementation. Looking forward to discussing this in the interview!

— **Dibyajyoti Dutta**