RE-IDENTIFICATION IN A SINGLE FEED

1.Introduction

The goal of this project was to implement a player re-identification system using a single video feed. The system detects players using a pre-trained YOLOv11 model and ensures consistent tracking even when players leave and re-enter the frame.

2.Approach and Methodology

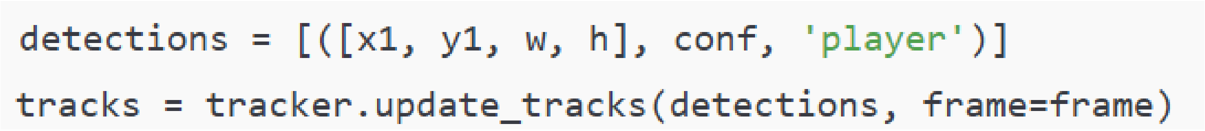
Used **Ultralytics YOLOv11** for detecting players which is already provided in the document.
Integrated **DEEP SORT** to track players with consistent IDs.
Reduced the bounding size or box size slightly to improve clarity.
Real time simulation with help of cv2 from Collab.

DEEP SORT:

Deep sort stands for Simple Online and Realtime Tracking is a powerful tracking by detection algorithm. It extends the original SORT algorithm by incorporating **appearance-based features**, enabling **re-identification** of objects (players, in our case) across frames even after masking.

Main Role: Maintain the same ID for players when they re-enter the frame later in the video.

Feed detections from YOLO to Deep SORT is done as following:



```
detections = [([x1, y1, w, h], conf, 'player')]
tracks = tracker.update_tracks(detections, frame=frame)
```

3. Techniques and Outcomes

a) Yolo Model for accurate detection of the players in the video provided.
b) Deep sort tracking for maintaining consistent IDs.
c) With help of bounding boxes no adjacent player is given the same ID and helps to differentiate with individual players.

4. Challenges Encountered

a) Large bounding boxes but no same player has the same Player ID.

b) **cv2.imshow()** not supported in Collab.
Used cv2_imshow() which is supported by Collab.

c) **Real-time simulation was limited** due to Collab's slower frame processing and lack of GPU but not greater impact on the video.

5. Future Improvements

a) Speed up the code so that it works faster and smoother on longer videos.