Approach and Methodology

I used a **YOLOv8 object detection model** to detect players in each frame of the input video. To maintain consistent player IDs across frames, I integrated the **SORT (Simple Online and Realtime Tracking) algorithm**, which uses Kalman filtering and Hungarian assignment to track bounding boxes effectively.

Techniques Tried and Their Outcomes

- **YOLOv8 + SORT (final implementation):**
  Successfully assigned consistent IDs to players across frames. SORT efficiently handled re-identification when players temporarily left and re-entered the frame.
- **Single detection per frame without tracking (initial attempt):**
  This approach failed to maintain consistent IDs, as detections were independent per frame.

Challenges Encountered

- **Module import errors:**
  Faced ModuleNotFoundError for the sort and filterpy modules, which were resolved by cloning the SORT repository from GitHub and installing filterpy via pip.
- **Video output compatibility:**
  Initially, the output video did not play due to codec issues. This was fixed by using the 'mp4v' codec for .mp4 output and ensuring correct VideoWriter initialization.
- **Model detection confidence tuning:**
  Adjusted the YOLO model’s confidence threshold to balance detection accuracy and false positives for player detection.

If Incomplete – Future Work

If I had more time or resources, I would:

- **Integrate DeepSORT:**
  Replace SORT with DeepSORT for appearance-based re-identification, improving robustness when players overlap or cross paths.
- **Cross-camera player mapping:**
  Extend the current single feed tracking to Option 1 by comparing embeddings from detections in two different camera feeds to establish consistent IDs across views.
- **Optimize performance:**
  Use GPU acceleration more effectively and implement batch processing for longer videos to reduce processing time.