# Player Tracking in Sports Video: Assignment Report

July 2025

## Introduction

This report details the approach to tracking players in a 15-second sports video using the Ultralytics YOLOv11 model and BOTSORT tracker. The goal was to detect players (class ID 2), assign unique IDs, and annotate the output video with green bounding boxes and red IDs above players' heads, maintaining ID consistency during collisions and frame exits/re-entries.

## Approach and Methodology

### Problem Definition

The task required multi-object tracking (MOT) in a sports video, where players move dynamically, collide, and may leave/re-enter the frame. Key challenges include occlusions, similar player appearances, and camera motion. The solution needed to be accurate, robust, and reproducible.

### Methodology

1. **Detection**:
    - Used a pre-trained YOLOv11 model (`player_model.pt`) to detect players (class 2) and referees (class 3).
    - Applied a confidence threshold of 0.6 to filter out low-quality detections, reducing false positives.
2. **Tracking**:
    - Employed the BOTSORT tracker from Ultralytics, which combines motion-based tracking (IoU, Kalman filtering) with Global Motion Compensation (GMC) for camera motion handling.
    - Tuned parameters: `match_thresh=0.6` for occlusion handling, `track_buffer=120` for track retention, and `gmc_method="sparseOptFlow"` for stability.
3. **Annotation**:
    - Drew green bounding boxes (`cv2.rectangle`) around detected players.

- Rendered red player IDs (`cv2.putText`) above heads, adjusting position to avoid clipping.

4. **Debugging**:
   - Added console output for the first 10 frames, showing detected classes, confidences, and number of tracks to verify performance.
5. **Error Handling**:
   - Skipped frames with no detections to prevent tracker errors.
   - Addressed initial errors (e.g., ReID feature mismatch) by disabling ReID temporarily.

# Techniques Tried and Outcomes

1. **BYTETracker**:
   - **Description**: Initial script used BYTETracker, a motion-based tracker.
   - **Outcome**: Produced output video but had ID switching during collisions due to reliance on motion cues (IoU, Kalman filtering).
   - **Issue**: IndexError when handling single detections, resolved by skipping empty frames.
2. **BOTSORT with ReID**:
   - **Description**: Switched to BOTSORT with `with_reid=True` and `model="auto"` to leverage appearance-based ReID for better ID consistency.
   - **Outcome**: Encountered `AttributeError: 'numpy.ndarray' object has no attribute 'cpu'` due to feature format mismatch (NumPy vs. PyTorch tensors).
   - **Resolution**: Disabled ReID (`with_reid=False`) to ensure script functionality, relying on BOTSORT's motion and GMC features.
3. **Parameter Tuning**:
   - **Description**: Adjusted `match_thresh` (0.7 to 0.6) and `track_buffer` (60 to 120) to improve occlusion handling and track retention.
   - **Outcome**: Expected to reduce ID switching, though ReID absence limits appearance-based tracking.
4. **Post-Processing**:
   - **Description**: Initially implemented centroid-based ID consistency check (50-pixel threshold) to validate tracks.
   - **Outcome**: No annotations in output video, likely due to overly strict filtering. Removed to ensure all tracks are drawn.
5. **Confidence Filtering**:

- **Description**: Set `conf=0.6` in detection and filtered detections to ensure high-quality inputs.
- **Outcome**: Reduced false positives, improving tracker input quality.

# Challenges Encountered

1. **ReID Feature Error**:
   - **Issue**: BOTSORT with `with_reid=True` and `model="auto"` raised an error due to NumPy arrays being passed instead of PyTorch tensors.
   - **Solution**: Disabled ReID temporarily; recommended updating Ultralytics and trying `model="yolo11n-cls.pt"`.
2. **No Annotations in Output**:
   - **Issue**: Output video matched input, lacking bounding boxes and IDs, likely due to strict post-processing filtering out tracks.
   - **Solution**: Removed centroid-based filtering to draw all tracks.
3. **ID Switching**:
   - **Issue**: BYTETracker caused ID switches during collisions due to limited appearance handling.
   - **Solution**: Switched to BOTSORT and tuned parameters; ReID would further improve this but was disabled due to errors.
4. **Performance**:
   - **Issue**: Inference times ($\sim$1060.9ms per frame) were high, impacting processing speed.
   - **Solution**: Suggested using a smaller model or lower resolution for future optimization.

# Current Status and Outcomes

- **Successes**:
  - The script runs without errors, producing `output_tracked_video.mp4`.
  - Detects players (class 2) and referees (class 3) with high confidences (0.83896–0.92468).
  - Uses BOTSORT for improved tracking over BYTETracker, with tuned parameters for occlusion handling.
- **Limitations**:
  - Output video now includes annotations (green bounding boxes, red IDs), but ID switching may occur without ReID.

  - ReID was disabled due to feature mismatch errors, limiting appearance-based tracking.

## Next Steps with More Time/Resources

1. **Enable ReID**:
   - Update Ultralytics to v8.3.x and use `model="yolo11n-cls.pt"` for ReID, improving ID consistency during collisions.
   - Download `yolo11n-cls.pt` from Ultralytics and test with `with_reid=True`.
2. **Custom ReID Training**:
   - Train a ReID model on a sports dataset (e.g., SportsMOT) to capture player-specific features like jersey numbers.
3. **Advanced Post-Processing**:
   - Reintroduce centroid-based filtering with a higher threshold (e.g., 100 pixels) or add IoU-based track validation.
   - Implement temporal smoothing for IDs to reduce switching.
4. **Performance Optimization**:
   - Use a smaller YOLO model (e.g., `yolo11n.pt`) or reduce video resolution to lower inference times.
   - Explore GPU acceleration for real-time processing.
5. **Dataset Enhancement**:
   - Retrain `player_model.pt` on datasets like TeamTrack or SportsMOT, focusing on occlusion scenarios.
6. **Alternative Trackers**:
   - Test advanced MOT algorithms (e.g., dual-mode Bayesian inference from TeamTrack) for complex sports scenarios.

## Conclusion

The script successfully detects and tracks players in a sports video, producing an annotated output with BOTSORT. While ID switching during collisions remains a challenge without ReID, the current solution is robust and reproducible. With further time, enabling ReID and optimizing performance would enhance accuracy and usability.

## References


- Ultralytics YOLO Docs - Track Mode: https://docs.ultralytics.com/modes/track/

- botsort.yaml: https://github.com/ultralytics/ultralytics/blob/main/ultralytics/cfg/trackers/botsort.yaml
- TeamTrack Dataset: https://arxiv.org/abs/2409.12790
- Ultralytics YOLOv8 Trackers Comparison: https://medium.com/pixelmindx/ultralytics-yolov8-object-trackers-botsort-vs-bytetrack-comparison-d32d5c82ebf3