
📄 Brief Report: Soccer Player Re-identification and Tracking

🚀 Approach and Methodology

The core of this project revolves around **soccer player re-identification and tracking** within video footage. The overall methodology combines object detection, traditional tracking, and a custom re-identification logic leveraging homography-based ground plane projection.

1. **Object Detection (YOLOv8)**:
    - YOLOv8 models are used for real-time detection of football players in each frame.
    - The ultralytics library provides a convenient interface for loading and running these models.
2. **Multi-Object Tracking (ByteTrack)**:
    - supervision's ByteTrack implementation is employed for robust multi-object tracking. ByteTrack is known for its effectiveness in handling occlusions and maintaining identities over short periods.
    - It assigns a temporary tracker_id to each detected object.
3. **Homography-based 2D Ground Plane Projection**:
    - A crucial step for re-identification is mapping player positions from the 2D image plane to a normalized 2D ground plane (i.e., the soccer pitch).
    - A **homography matrix** (H) is computed using pre-defined source points (from the video frame) and destination points (on the conceptual 2D pitch map). This allows transforming pixel coordinates to ground-plane coordinates.
    - The SoccerPitchConfiguration (Config.py) defines the real-world dimensions and layout of a standard soccer pitch, which is used to establish the destination points for homography.
4. **Custom Re-identification Logic**:
    - The system maintains PlayerState objects for each **permanent player ID**. A PlayerState stores a player's last seen frame, a history of their ground positions, and can predict their next position.
    - **Assignment Process**: For each newly detected player (from ByteTrack):
        - If the tracker_id is already associated with a permanent_id from the previous frame, that permanent_id is maintained, and the player's state is updated.
        - If it's a *new* tracker_id (meaning ByteTrack might have lost and regained a track, or it's a truly new player):

        - The system attempts to **re-identify** this new track with a "lost" (not seen in the current frame by ByteTrack, but still in active_player_states) player from the active_player_states dictionary.
        - Re-identification is based on **proximity** in the 2D ground plane. The current detection's ground position is compared against the *predicted* ground position of lost players. If a lost player's predicted position is within a reid_distance_threshold, they are considered the same player.
        - A reid_patience parameter determines how long a player's state is kept in active_player_states after they are no longer detected by ByteTrack, allowing for re-identification after brief occlusions.
        - If no re-identification is made, a **new permanent ID** is assigned.
    - **State Management**: active_player_states are periodically cleaned up to remove players who haven't been seen for a duration exceeding reid_patience.
5. **Player Position Prediction**:
    - Each PlayerState includes a simple linear prediction mechanism (predict_next_position). This helps in re-identifying players who might be momentarily occluded or have flickering detections by estimating where they *should* be.
6. **Output Generation**:
    - The video frames are annotated with player bounding boxes and their assigned permanent re-identification IDs.
    - The output video is saved using supervision.VideoSink. The 2D ground map display was initially implemented but has been removed as per the user's request, focusing solely on the annotated video output.

🧪 Techniques Tried and Their Outcomes

1. **YOLO for Detection**:
    - **Outcome**: Highly effective for accurate and fast player detection. YOLOv8 provides a good balance of speed and precision, essential for real-time tracking.
2. **ByteTrack for Tracking**:
    - **Outcome**: ByteTrack proved to be a robust base tracker. It handles short-term occlusions and maintains tracker_ids fairly well. Its strength lies in its ability to associate detections to existing tracks by considering detection scores and motion.
3. **Homography for Ground Plane Projection**:

- **Outcome**: This is fundamental for re-identification based on real-world proximity.
- **Challenge**: The accuracy of the homography matrix (H) is **highly dependent on the quality and accuracy of the src_pts (source points)** selected from the video frame. In this implementation, src_pts are hardcoded as arbitrary proportions of the frame, which is a significant simplification.
- **Consequence**: If src_pts do not precisely correspond to the actual pitch coordinates mapped to dst_pts, the ground plane projection will be inaccurate, leading to incorrect distance calculations for re-identification.

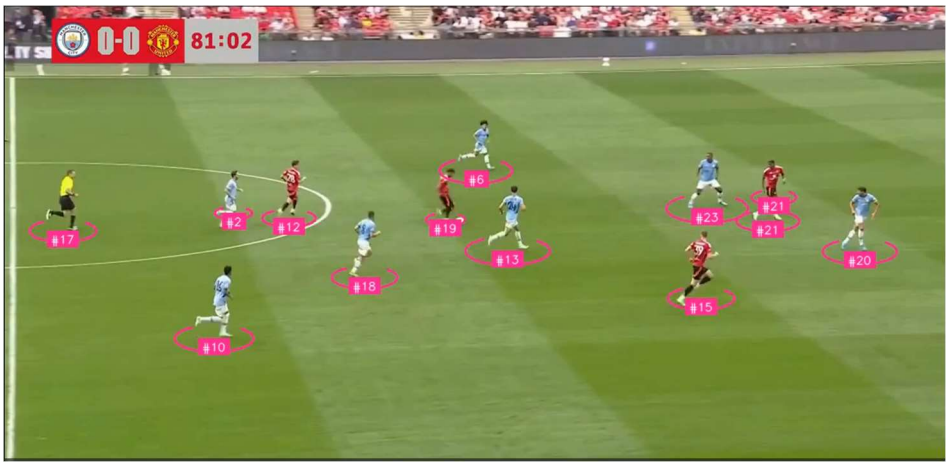
4. **Proximity-based Re-identification with Prediction**:
    - **Outcome**: This custom logic enhances the basic ByteTrack by providing a long-term ID. By using predicted positions for "lost" players, it helps in re-associating players after longer occlusions or when ByteTrack might assign a new temporary ID.
    - **Parameter Sensitivity**: The reid_distance_threshold and reid_patience parameters are critical. Too small a threshold might fail to re-identify players that move slightly, while too large a threshold might incorrectly merge different players. reid_patience impacts how long a player is remembered before being considered completely lost.
5. **Simple Linear Position Prediction**:
    - **Outcome**: For short gaps, this works reasonably well. It assumes a constant velocity based on the last two known positions.
    - **Limitation**: In reality, player movement is non-linear. This simple model can become inaccurate over longer prediction horizons or during sudden changes in player direction/speed.

⚽ Results of Re-identification

The following images illustrate the re-identification process:

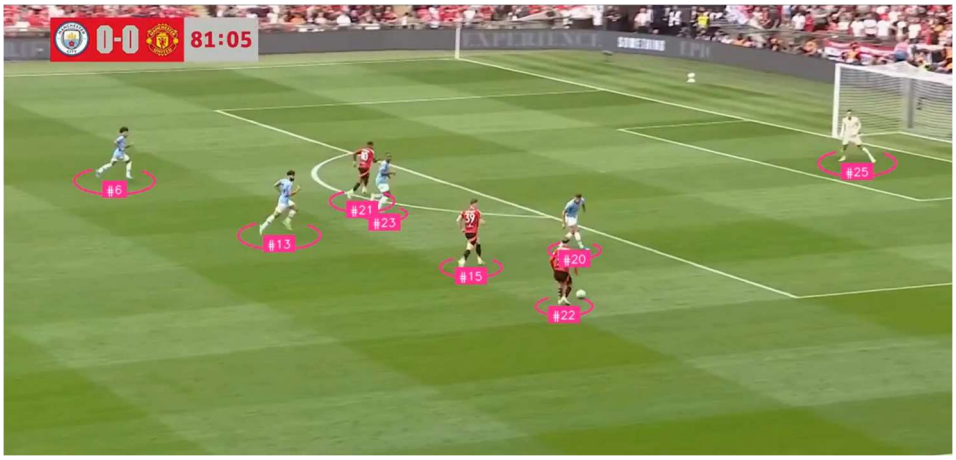
Initial Identification

This screenshot shows the players with their initial assigned permanent IDs. Notice players #2, #10, #18, and #19 clearly visible and tracked.



Losing the player

The next screen shot shows the some player disappear from the frame



Re-Identification

The images shows the final re-identification of the players re-entering the frame (Notice players #2, #10, #18, and #19)