Player Re-identification in Sports Footage

1. Introduction

This system is intended to process a football game's video feed in order to identify teams based on jersey color, detect players, assign unique IDs, and display their real-time movement on a live tactical map and on video.

2. Methodology

The system operates through a multi-stage pipeline, as outlined below:

2.1 Model Initialization

Each frame's players were identified using a YOLO-based object detection model. A pre-trained weight file (models/best.pt) that is optimized for identifying the various classes from a football feed is used to load the model.

2.2 Video Ingestion and Output Handling

OpenCV is used to process the input video. To set up an output writer that saves the annotated results as a new video file, frame width, height, and frame rate are extracted.

2.3 Detection and Identity Tracking

For each frame:

- The detector identifies bounding boxes around players.
- Non-player classes are filtered out.
- Player crops are extracted and passed to a team classifier.
- A Euclidean distance-based tracker (GlobalTracker) determines whether the player corresponds to an existing ID (based on proximity and team) or should be assigned a new ID.

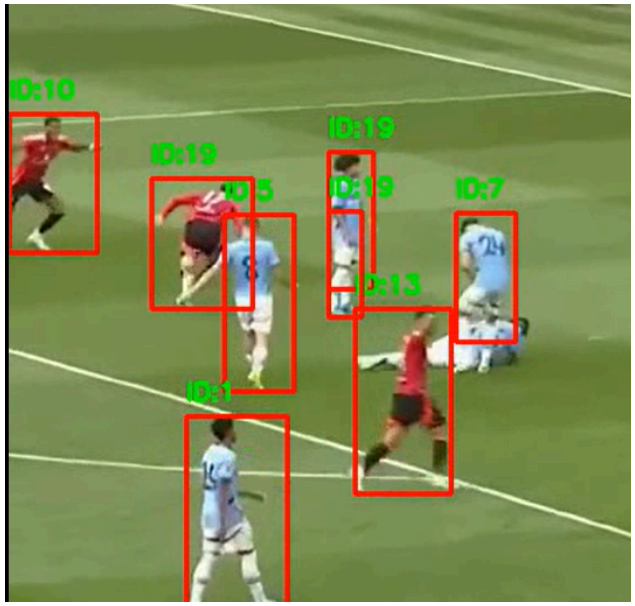

Fig.1: The Player ID is stored with the Team Jersey Color to prevent false detections

2.4 Tactical Map Rendering

Using Matplotlib, a live tactical map displays player positions as colored square markers, each labeled with a team code and unique player ID. This map provides a bird's-eye strategic view of player distribution across the field.

2.5 Output Generation

Two outputs are generated:

- ❖ A video with bounding boxes and player/team labels.
- ❖ A real-time tactical map is updated for every frame

3. Approaches to the problem statement

Approach 1: DeepSORT Algorithm

The YOLO object detector and the DeepSORT algorithm were used to create the tracking component during the first stage of implementation. DeepSORT associates detections across frames using appearance features and motion prediction. The deep-sort-realtime==1.3.2 model was applied.

The object detector frequently generated bounding boxes that were too big and occasionally overlapped with nearby players. The tracker found it challenging to distinguish between distinct players due to this spatial overlap. Furthermore, player regions were not accurately covered because the bounding boxes did not adapt to the objects' scale. DeepSORT was deemed unsuitable for the particular requirements of this task due to its incapacity to resolve overlapping detections in scenarios with dense clustering.

Fig.2: DeepSort Bounding Boxes were overlapping

Approach 2: Gallery-Based Identification

An alternative approach was considered to address the limitations of overlapping bounding boxes by using visual similarity between player crops. In its initial form, this method relied on frame-to-frame matching. Specifically, cropped images of each detected player from the current frame were compared only against the crops from the immediately preceding frame. The assumption was that the visual appearance of a player would not significantly change between two consecutive frames, allowing for straightforward matching based on visual features.

However, this assumption did not hold in practical scenarios. Due to rapid player movement, occlusion, and camera perspective, the appearance of a player could change notably even between adjacent frames. Additionally, once an incorrect ID assignment occurred—such as two players being swapped—the tracker would treat the incorrect crop as ground truth for the next frame. This caused the ID error to persist and propagate forward.

Approach 3: Modified Gallery-Based Identification

The technique was expanded to employ a gallery system, which stored several of each player's previous crops, in order to get around this. Every new crop was compared to a random selection of five pictures taken from each player's folder during the matching process. Inconsistent lighting, angles, and partial occlusions across samples resulted in high variability even though this improved the matching window and diversity. As a result, identity assignment varied between frames and similarity scores lost their reliability. The tracking was less reliable than the later, simpler spatial proximity-based approach because the same player would often be given multiple IDs.

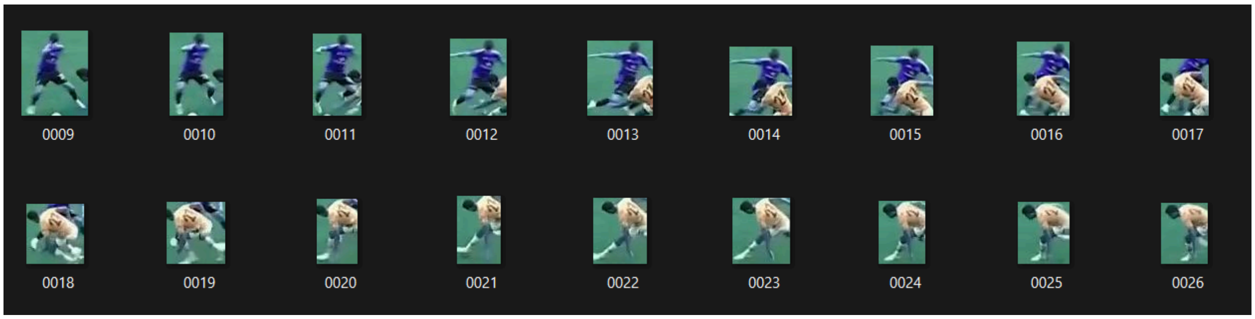

Fig. 3: ID Mismatch errors

Approach 4: Euclidean distance approach

The idea for the current approach came from observing how players typically behave in a football match. In any given frame, there are only a limited number of players on the field, and from one frame to the next, those players don't suddenly disappear or move drastically. It wouldn't make sense for completely new players to randomly appear out of nowhere in every frame. Based on this, it became clear that tracking should rely on spatial continuity — that is, assuming a player in the next frame is likely to be near where they were in the previous frame. This insight led to replacing the earlier gallery-based method, which compared crops visually and often failed due to lighting and angle differences. Instead, the new method uses a simple distance check: if a detected player in the current frame is close to where a known player of the same team was in the last frame, they're likely the same person. This assumption allowed for more stable ID assignment, without relying on visual appearance, which had proven unreliable in the earlier attempts.

To complement this frame-wise identity tracking, a tactical map was also introduced, providing a live top-down visualization of player positions on the field. This helped in both verifying the accuracy of the tracking process and analyzing player movement patterns across time.

4. Challenges Faced

1. **Limited Hardware**: No dedicated GPU; slow inference on CPU affected real-time processing and testing.

2. **Complexity of Object Tracking**: Required significant effort to understand tracking principles and identity preservation across frames.

3. **MediaPipe Limitations**: MediaPipe was tested but found unsuitable for multiplayer sports scenarios due to poor performance in crowded frames.

4. **ID Swapping Between Players**: Frequent swapping of IDs when players were close or crossed paths, causing persistent misidentification.

5. **Duplicate IDs in the Same Frame**: Multiple players were sometimes assigned the same ID in a single frame, violating identity uniqueness.

6. **Incorrect Detection of Body Parts**: The model occasionally detected torso and legs as separate individuals, leading to false positives.

7. **Visual Variability**: Gallery-based matching failed due to lighting, angle, and resolution changes, reducing identification consistency.