Report
On
Soccer Players Re-identification and Tracking
(Option 2: Re-identification in a single feed)

Under the assignment

For

AI/ML Intern

In Company

Liat.ai

By:

Vikas Thakur

Btech CSE 7th Sem

HPTU, Hamirpur

Introduction:

In real-time sports analytics and surveillance, it is essential to accurately track multiple players throughout a video, even when players leave and re-enter the frame. Traditional object trackers like Deep SORT often fail to maintain consistent IDs over time, leading to identity switches. To address this, we introduce a unique **Pose-based Re-identification (Re-ID)** system that uses body posture (keypoints) to ensure identity consistency throughout the video.

Objective:

- Detect and track players in a sports video.
- Re-identify players after occlusion or exit.
- Provide a minimap and motion trails for better visualization.
- Assigning same identity to players re-entering in the frame

Approach and Methodology:

1. Player Detection Using YOLOv5

The first step in our system is to identify all players visible in each video frame. For this, we used the **YOLOv5 object detection algorithm**, known for its speed and accuracy. We trained or fine-tuned a custom YOLOv5 model (best.pt) to focus specifically on player detection. This model outputs a set of bounding boxes — each representing a detected player in the frame.

- The detection occurs on a **frame-by-frame basis**, enabling consistent identification of players as they move through the scene.
- YOLOv5 was selected because of its balance between real-time performance and accuracy, which is crucial for sports or surveillance footage.

2. Object Tracking with Deep SORT

After detection, the bounding boxes are passed to the **Deep SORT (Simple Online and Realtime Tracking)** algorithm. Deep SORT is a tracking algorithm that not only follows objects over time but also assigns **unique ID numbers** to each one. This allows the system to keep track of the same player across multiple frames, even if they move quickly or change direction.

- Deep SORT uses a **Kalman filter** to predict the next position of each object, and a **feature extractor** to verify appearance similarity.
- This ensures that even with momentary occlusion or overlap, players retain the same ID — to a certain extent.

However, if a player **leaves the scene** and then returns later, Deep SORT typically assigns a **new ID**, thinking it is a new person. This is a key challenge addressed in the next stage.

3. Pose-Based Re-Identification (PoseReID)

To overcome ID reassignment issues when players return, we introduced a **Pose Re-Identification module** using **Mediapipe Pose**. Instead of relying on appearance alone, this module looks at the **body posture or skeletal structure** of players.

- For each tracked player, the system captures a **pose embedding** — a vector representing the 3D positions of body joints.
- These embeddings are stored, and when a new person appears, their pose is compared to previously saved ones using **cosine similarity**.
- If a match is found within a similarity threshold, the system **associates the new person with a previous player ID**, preserving consistency.

This method ensures that returning players are **correctly re-identified**, avoiding duplicate identities in the video.

4. Trail Overlay for Movement Visualization

To show movement history, a **trail overlay** system draws lines indicating where each player has been over time. Each line connects the past few positions of a player’s bounding box, giving a sense of direction and speed.

- This feature helps in visual analysis but can be visually overwhelming if too many trails are shown.
- Therefore, we allow customization to limit the number of stored points or turn trails off entirely.

5. Minimap Projection

Lastly, to enhance the visual representation, a **minimap** feature is added. This component projects each player's position onto a small section of the frame — similar to the minimaps used in video games.

- It offers a bird's eye view of the field, useful for coaching, analysis, or strategy planning.
- Player IDs are shown here too, giving a clear overview of team movement.

Techniques tried & their Outcomes:

1. Object Detection Techniques

Techniques Tried:

- **YOLOv5** (final choice)
- **OpenCV Haar Cascades** (initial test)
- **YOLOv8** (tested but discarded due to resource constraints)

Outcomes:

- Here Cascades were lightweight but **too inaccurate** for real-world player detection.
- YOLOv8 performed slightly better than YOLOv5 but required a **GPU for smooth inference**, which was not available in our setup.
- YOLOv5 gave an **excellent balance between speed and accuracy** on CPU, making it ideal for our final model.

2. Tracking Algorithms

Techniques Tried:

- **Kalman Filter + Hungarian Algorithm (SORT)**
- **Deep SORT** (final choice)

Outcomes:

- Standard SORT worked for basic tracking, but suffered from frequent ID switches when players overlapped or re-entered.
- Deep SORT, with its **appearance-based embedding matching**, handled crowded frames better.
- However, it still failed in cases of long occlusion or when a player exited and re-entered the frame. This limitation led to introducing Pose-Based Re-Identification.

3. Re-Identification (Re-ID)

Techniques Explored:

- **Deep feature-based Re-ID (CNN embeddings)** — not implemented due to model complexity.
- **Pose-based Re-ID using Mediapipe** — final choice.

Outcomes:

- While CNN-based Re-ID would offer high accuracy, it required **model training and a GPU**.
- Pose-based Re-ID using **Mediapipe** was simpler, light-weight, and didn't need training. It worked surprisingly well in distinguishing players by **body structure and posture**.
- Using **cosine similarity between pose embeddings**, we were able to **recover correct IDs** for players who exited and later re-entered the frame.

4. Visualization Techniques

Trail Overlay:

- Initially, we allowed **unlimited trail history**, but the screen became cluttered.
- We then **limited trail length** to the last 15–20 frames to reduce visual noise.

Minimap:

- Introduced to simulate a real-time overview, especially useful in sports contexts.
- It made the system feel **game-like** and helped analyze overall player movement.

5. Performance & Optimization

- **Resolution Handling**: Processed video at 720p to maintain balance between detail and performance.
- **Real-time Constraints**: Despite being on CPU, optimizations in detection and reduced pose analysis frequency helped keep frame processing feasible.

Challenges encountered:

1. ID Switching & Reuse

Tracking failed to retain consistent IDs when players briefly left and re-entered the frame. Deep SORT often assigned new IDs, mistaking returning players as new ones. To encounter this we used Pose-based Re-ID using Mediapipe was added to reduce ID reuse by matching body structures, improving continuity.

2. No GPU Availability

Without a GPU, YOLO and pose estimation ran slowly, affecting real-time usability. To handle this, We chose lightweight models (YOLOv5), reduced frame resolution, and optimized the processing steps for better CPU performance.

3. Pose Estimation Failures

Players partially out of frame or in motion blur often caused Mediapipe to fail, leading to missing embeddings. For this such frames were skipped without crashing the system, relying on Deep SORT temporarily.

4. Cluttered Visual Output

Trail lines and bounding boxes for multiple players made the screen messy and hard to interpret. To encounter this, Trail lengths were limited, and visual elements like minimap and trail overlays were made more minimal.

5. Output Saving Issues

Early versions failed to save output video due to missing directories. Output directory checks and proper video writer settings ensured reliable saving.

Future Work:

With extended time and better computational resources, several impactful enhancements can be made to this player re-identification system.

- One major improvement would be the integration of **deep learning-based Re-ID models**, trained specifically on sports datasets. These models can handle complex scenarios like occlusions and similar player appearances more effectively than pose-based logic alone. GPU acceleration would also enable real-time performance even with heavy models.
- Additionally, collecting and fine-tuning on **domain-specific datasets** (e.g., football, basketball footage) would increase accuracy in player detection, tracking, and re-identification. Quantitative evaluation using metrics like IDF1 and mAP could help benchmark the system.
- In future iterations, **action recognition** could be added, allowing the system to detect movements like running, shooting, or passing. This would open doors to tactical analysis and richer player profiling.
- Another key upgrade would be building an **interactive analytics dashboard** to display trails, movement heatmaps, zone-based statistics, and match summaries — beneficial for coaches, analysts, and broadcasters.
- Finally, packaging the solution as a **web or mobile application** would make it accessible to wider users. Deploying on the cloud would also enable remote processing and collaboration.

In short, this system has strong potential to grow into a robust sports analysis platform with deeper intelligence, broader usability, and higher accuracy.