

# Report on Player Re-Identification in Soccer Video

## My Approach

### 1. Object Detection

I used the provided YOLOv8 model. I used it to detect players frame-by-frame in the video. I saved all the detections (bounding boxes) in a file called yolo\_detections.json.

### 2. Re-Identification

To identify and track players across frames, I used a Vision Transformer model (google/vit-base-patch16-224) from Hugging Face. This model was used to extract visual features from each player crop. These features were then compared using cosine similarity to decide if the current player matched any existing player in the database.

I also stored multiple features per player over time (temporal history) and used a weighted average of recent features to make the matching more stable.

If a player matched an existing one (based on similarity score and motion), I assigned the same ID. Otherwise, I assigned a new ID and saved the features.

### 3. Motion Check

To avoid wrong matches, I added a motion consistency check. If a player's movement between frames was too much, I reduced the similarity score.

---

## Techniques I Tried

1. **ViT feature-based re-ID:** This worked decently and helped identify players across frames.
2. **Jersey number recognition using ViT:** I tried using another vision transformer (from timm) to extract jersey numbers, but this did not work properly.

---

## What Did Not Work

- Jersey number detection failed on the video because the input quality was low, and the resolution was not enough for reading numbers.
  - Also, this method was very slow and took more than 4 hours to run. I could not fine-tune the model due to time and resource limitations.
-

## Challenges I Faced

1. Low-resolution video made feature extraction harder.
  2. I could not train or fine-tune a model due to lack of time and GPU power.
  3. Jersey number detection didn't work well, so I had to rely mostly on visual feature similarity.
- 

## What I Would Improve With More Time

If I had more time and better resources, I would do the following:

- I would fine-tune a Vision Transformer on soccer player data to extract jersey numbers. Then I would integrate it into my existing code. The idea is: if a jersey number is detected, I would store it in a database. Later, if the same jersey number appears again, I would check the database. If it exists, it means this is an old player, so the same ID will be assigned.
  - If the jersey number is not found in the database, then I would extract the visual features of the player and compare them with existing features in the database using cosine similarity or another metric. If the similarity is above a defined threshold, the player will be assigned the ID of the most similar existing player. If the jersey number was also detected, the database would be updated with that number against the assigned ID. If there is no similar match, a new ID will be assigned, and the player's features (and jersey number if available) would be stored in the database.
- 

## Summary

Overall, I was able to build a working pipeline that detects players using YOLOv11 and tracks them using visual similarity from a ViT model. Though jersey number detection didn't work, the base system gives decent tracking results. With more time, I could improve it further by adding jersey number recognition and optimizing the speed.