PLAYER RE-IDENTIFICATION ACROSS MULTIPLE CAMERA VIEWS USING RESNET50

Introduction

In this project, I aimed to address the challenge of player re-identification across different camera views — specifically between a broadcast view and a tacticam view. My goal was to extract robust feature embeddings from detected player crops and use those to find cross-view matches with high confidence. The problem is inherently complex due to variations in lighting, resolution, player pose, and viewpoint changes between camera feeds.

To achieve this, I implemented a modular pipeline that integrates YOLOv11 for player detection and ResNet50 for deep feature extraction, followed by cosine similarity matching for re-identification. While the pipeline performs reasonably well, the process is still in early stages and offers many avenues for refinement and extension.

Techniques and Methodology

1. Player Detection (YOLOv11)

I used YOLOv11 to detect players from two different camera angles. Detected bounding boxes were cropped using the --save-crop flag. Two crop folders were generated: predict_broadcast for broadcast view and predict_tacticam for tacticam view.

2. Feature Extraction (ResNet50)

A pre-trained ResNet50 model (with the classification head removed) was used to extract deep features. Each crop was resized to 256x128, normalized according to ImageNet standards, and passed through the network. The extracted features are 2048-dimensional vectors, stored in .npz files for each view.

3. Matching via Cosine Similarity

Cosine similarity was computed between each feature vector from the broadcast crops and all tacticam crops. For each broadcast player, the tacticam crop with the highest cosine similarity was considered the best match. The resulting matches and similarity scores were saved in matched_players.csv.

Outcomes and Results

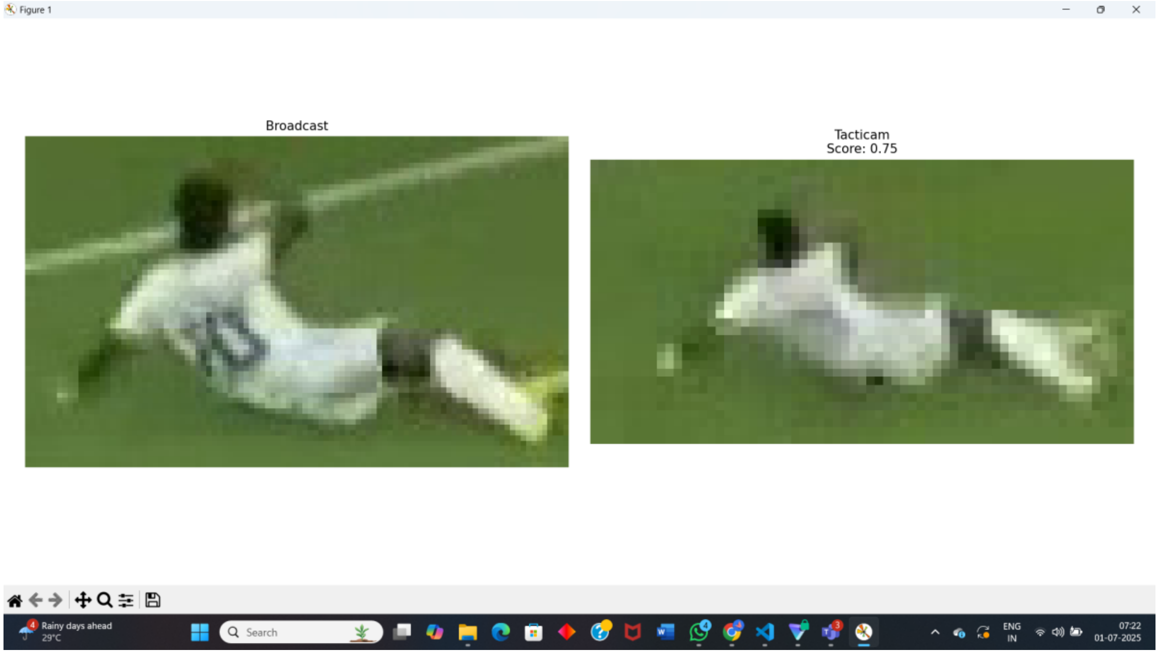
Using the provided CSV (matched_players.csv), I inspected the top matches obtained by the system. Here is a summary of the first few predictions:

#	Broadcast Image	Tacticam Image	Cosine Similarity
1	00000000.jpg	00000014.jpg	0.9053
2	00000001.jpg	00000015.jpg	0.8804
3	00000002.jpg	00000016.jpg	0.8610
4	00000003.jpg	00000017.jpg	0.8578
5	00000004.jpg	00000018.jpg	0.8436

These similarity values suggest that the model is able to identify some correct or semi-correct player matches across views. Matches with cosine similarity greater than 0.85 appear quite reasonable visually in many cases.

Matched Player Examples

Below is a visual comparison of matched players as determined by the cosine similarity metric.



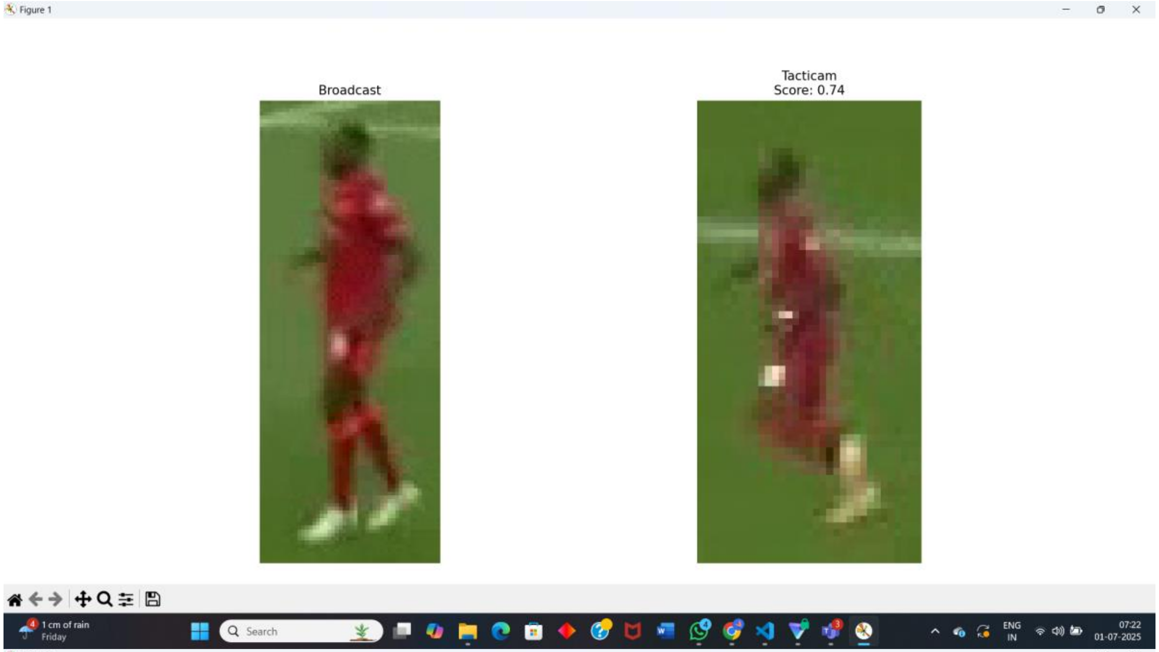


Figure 1
Broadcast
Tacticam
Score: 0.74
1 cm of rain
Friday
Search
ENG
IN
07:22
01-07-2025

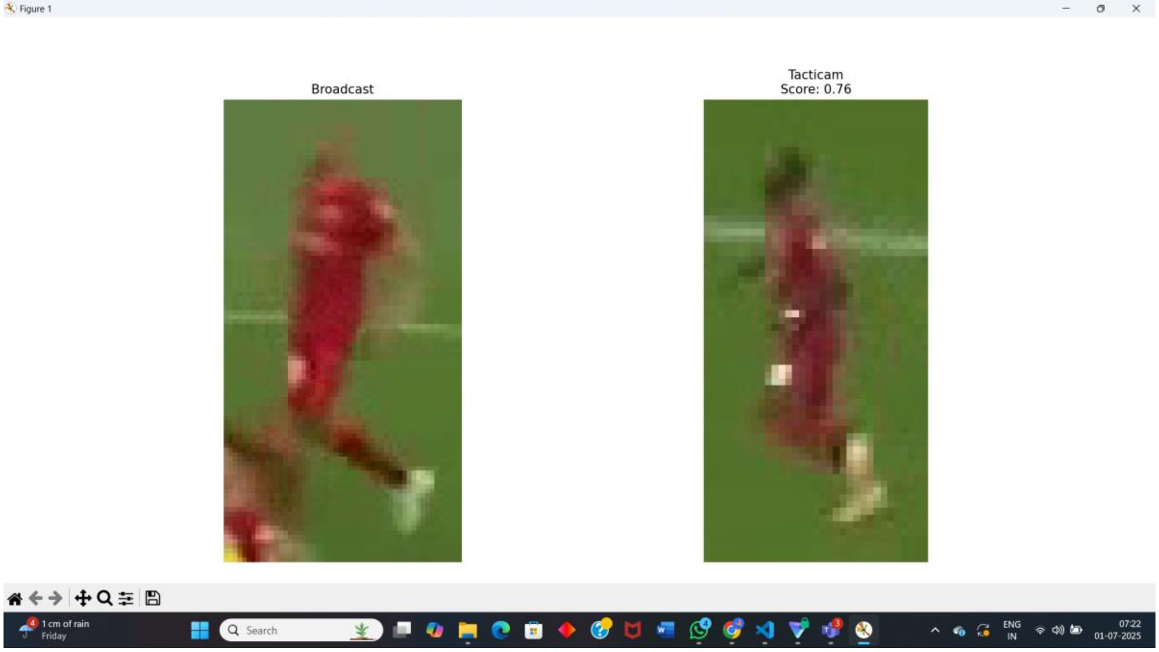
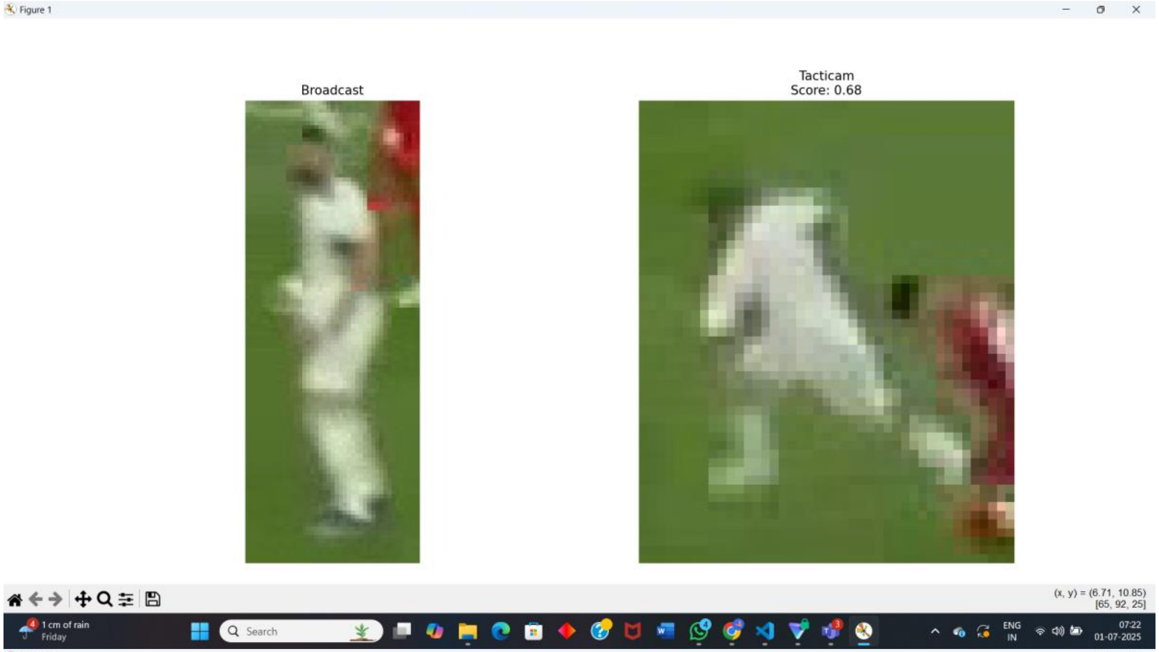
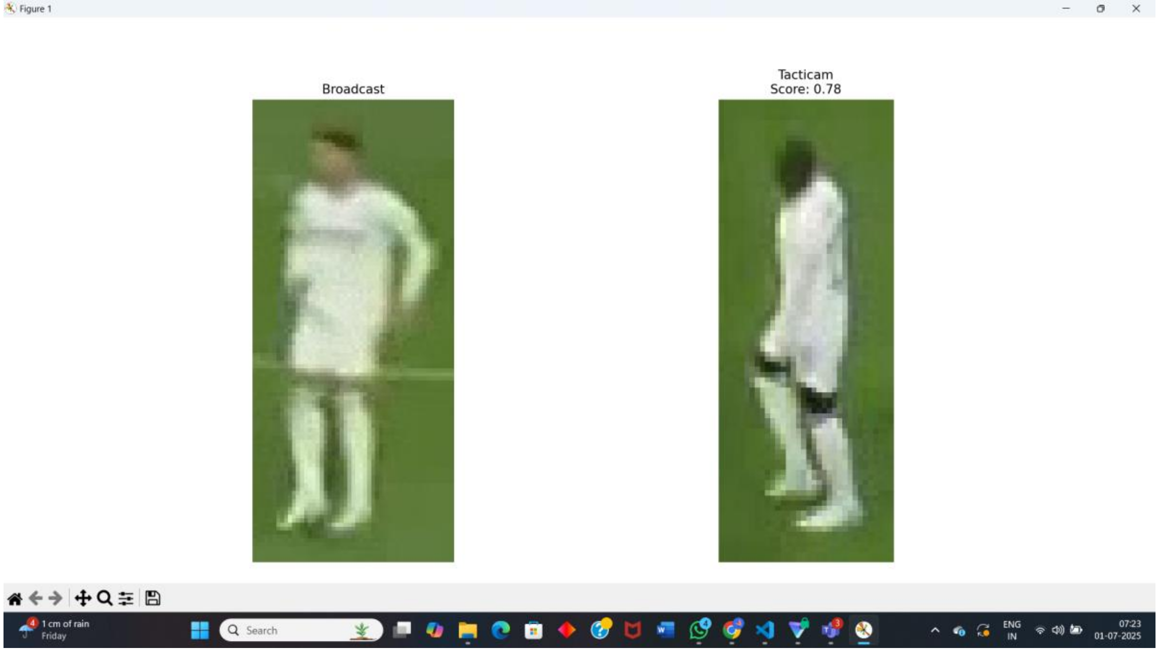


Figure 1
Broadcast
Tacticam
Score: 0.76
1 cm of rain
Friday
Search
ENG
IN
07:22
01-07-2025





Limitations and Reflections

While the pipeline provides a structured and reproducible approach, I encountered some key limitations that affected overall performance:

1. Low-Quality YOLOv11 Crops:

Many of the cropped player images produced by YOLOv11 were blurry or poorly framed. This greatly reduced the discriminative quality of the features extracted by ResNet50. I observed that crops varied in size, resolution, and clarity, which is detrimental for re-identification.

2. Lack of Temporal Information:
The current system only uses static appearance-based features. Incorporating temporal continuity (e.g., using a short sequence of frames per player) could provide richer cues for identification. Temporal context can disambiguate players with similar appearances who move differently.

3. No Spatial-Aware Features:
Spatial orientation or position of the player in the frame was not considered. Integrating spatial priors or pose-based embeddings could strengthen match confidence, especially when the background is cluttered.

Future Improvements

Based on the challenges above, I would like to propose the following directions for future work:

- Improve Crop Quality: Manually refining detection outputs or using high-resolution models can significantly enhance crop quality.
- Multi-Frame Integration: Averaging features over multiple consecutive frames or using temporal networks (e.g., 3D CNNs or LSTMs) can help with more robust identity encoding.
- Pose-Based Features: Using body pose estimators (like MediaPipe or OpenPose) to add skeletal keypoint features may improve match accuracy in cluttered scenes.
- Hard Negative Mining: Introducing negative samples during training or re-ranking steps could improve discriminative ability of the matching.

Acknowledgments

This project was built using publicly available tools including YOLOv11 for detection and PyTorch with Torchvision's ResNet50 model for feature extraction. My work stands on the shoulders of these open-source efforts and I am grateful for the community support.

Final Thoughts

This project was a valuable learning experience that deepened my understanding of re-identification systems. While the current performance is modest and still under development, I believe that with improvements in input data quality and the use of temporal and spatial modeling, the system can be taken much further.

I humbly acknowledge the current limitations of the system and look forward to iterating on it with more robust strategies.

Author: Anisha Singh

GitHub: https://github.com/anisha-singh-2004