***Project Report: Player Re-Identification Across Camera Views***

**1. Approach & Methodology**

The objective was to ensure consistent player identities across two video feeds (broadcast.mp4 and tacticam.mp4) of the same match captured from different camera angles. The task involved detecting players in both videos and re-identifying them using a combination of visual appearance features and similarity matching.

The pipeline was structured into three key phases:

* Detection: Detect players in each frame using a YOLOv11-based model (best.pt).
* Feature Extraction: Extract appearance embeddings using a ResNet50 CNN from cropped player images.
* Player Matching: Use cosine similarity between embeddings and apply the Hungarian algorithm to establish player correspondences.

**2. Techniques Used & Their Outcomes**

| Phase | Technique | Description | Outcome |
| --- | --- | --- | --- |
| Detection | YOLOv11 (Ultralytics, fine-tuned) | Detected player class in each frame of both videos | Successfully detected players in most frames with confidence scores |
| Feature Extraction | ResNet50 + Torch | Extracted 2048-dimensional appearance embeddings from cropped player regions | Produced robust visual features for similarity matching |
| Matching | Cosine Similarity + Hungarian Algorithm (scipy) | Computed cost matrix between tacticam and broadcast embeddings; solved optimal assignment | Correctly matched most players across views with consistent player\_id assignment |

The entire pipeline was implemented using Python, leveraging torch, torchvision, ultralytics, scipy, and opencv

**3. Challenges Encountered**

File Size Limits on GitHub

* Issue: best.pt (≈186MB) exceeded GitHub’s 100MB file limit.
* Solution: Integrated Git LFS and rewrote Git history using git filter-repo to clean large file traces before pushing.

Video Frame Alignment

* Issue: Videos captured from different angles may not be frame-synchronized.
* Mitigation: Assumed approximate frame alignment based on temporal continuity; future enhancement could include timestamp syncing or optical flow.

Visual Similarity Variance

* Issue: Varying lighting, angles, and occlusions between views made re-identification harder.
* Solution: Used deep feature embeddings instead of raw pixel similarity; results improved significantly using ResNet50.

**Final Result**

* The system achieved stable player re-identification across both feeds for most frames.
* The pipeline is modular and can be extended with player tracking, ball detection, or real-time deployment.