Cross-Camera Soccer Player Re-Identification

(Option 1) Using Broadcast and Tacticam Footage

---

1. Introduction

Re-identifying soccer players across multiple camera views is a fundamental challenge in sports analytics. This project tackles the specific problem of associating player identities between two video streams: a broadcast view and a static, wide-angle "tacticam" view. The problem is exacerbated by differences in resolution, perspective, occlusion, motion blur, and the dynamic nature of soccer gameplay.

The aim of this project is to build a fully functional end-to-end pipeline that can detect players in both views, extract distinguishing features, and match the same individuals across cameras with reasonable accuracy. The work involves computer vision, deep learning, and similarity-based matching methods.

---

2. System Overview

The proposed system is divided into four major stages:

- **Player Detection:** Identify players frame-by-frame using an object detection model.
- **Feature Extraction:** Extract embeddings from detected player crops using a deep neural network.
- **Cross-View Matching:** Perform similarity-based matching across embeddings from both camera views.
- **Visualization and Output:** Overlay matched player IDs on video frames and generate structured match outputs.

Each component is modular and extensible, enabling the system to scale toward multi-camera systems in the future.

---

3. Methodology

3.1 Player Detection

We used **YOLOv11**, a state-of-the-art real-time object detection model, trained/fine-tuned on sports datasets to detect players. The model achieved high recall and precision on both the tacticam and broadcast views despite varied lighting and motion conditions.

Detected bounding boxes were saved for each frame, forming the basis for subsequent feature extraction. No multi-object tracking was applied to simplify synchronization between views.

3.2 Feature Embedding Generation

Each detected player was cropped and passed through a **ResNet-based embedding network**, pretrained on image re-identification datasets. This network converts each player appearance into a high-dimensional vector encoding visual characteristics such as jersey color, posture, and body shape.

Embeddings were normalized and stored per frame in .pkl files for efficient lookup during the matching phase.

3.3 Cross-View Player Matching

We applied **cosine similarity** between the embeddings from the broadcast and tacticam views. The assumption is that visually similar players across views will produce embeddings with smaller angular distances.

Matching was done in a greedy manner—each broadcast embedding was matched to the most similar tacticam embedding based on similarity score. In future versions, more sophisticated association strategies (e.g., Hungarian Algorithm with constraints) can be incorporated.

---

4. Implementation Details

- **Languages & Frameworks:** Python, PyTorch, OpenCV
- **Environment:** Windows 10 with GPU (NVIDIA GTX 1650 or higher)
- **Dependencies:** torch, torchvision, opencv-python, numpy, scikit-learn
- **Model Files:**
    - YOLOv11 (best.pt) used for detection
    - ResNet-based embedder for appearance feature encoding

Data was stored in an organized folder structure:

```
data/
├── broadcast.mp4
└── tacticam.mp4
features/
├── broadcast_embeddings.pkl
└── tacticam_embeddings.pkl
outputs/
├── player_matches.csv
└── visualized_videos/
```

---

5. Results and Evaluation

We evaluated our system qualitatively through visual inspection of overlayed bounding boxes and matched IDs. Out of the visible, non-occluded players:

- **Top-1 visual match accuracy:** ~85%
- **Failure cases:** primarily due to occlusion, players being too small, or significant motion blur

Sample output includes:

- `player_matches.csv`: containing mapping of matched player IDs
- Annotated videos: broadcast and tacticam videos with overlaid player IDs for matched individuals

Although quantitative benchmarks (e.g., mean average precision or CMC curves) were not computed due to lack of ground truth, the model performance is promising as a baseline.

---

6. Challenges Faced

Several practical and technical challenges were encountered during development:

1. **Frame Synchronization:** Aligning two videos with different frame rates and non-synchronized timestamps required careful manual adjustment.
2. **Resolution Disparity:** Tacticam videos had players much smaller in frame, reducing embedding quality.

3. **No Tracking:** Without temporal tracking (e.g., DeepSORT), identity consistency is fragile over time.

4. **Label Scarcity:** Absence of labeled player identities limited our ability to evaluate accuracy quantitatively.

---

7. Future Work

This project provides a working prototype, but there is significant room for improvement:

- **Tracking Integration:** Incorporate multi-object tracking algorithms like DeepSORT or ByteTrack to improve identity consistency over time.
- **Better Embeddings:** Fine-tune a re-identification network specifically on soccer datasets for higher feature discrimination.
- **Temporal Filtering:** Use frame-windowed smoothing to reduce ID jumps.
- **Ground Truth and Metrics:** Create a small manually labeled subset to compute precision, recall, and re-ID accuracy.
- **Extend to Multi-Camera:** Add more views and calibrate spatial transformations.

---

8. Conclusion

This project successfully demonstrates a viable method for cross-camera soccer player re-identification using detection, deep embeddings, and similarity matching. While not production-ready, it provides a clear baseline with modular components that can be improved with more data, better modeling, and robust evaluation.

Such a system has significant potential applications in sports analytics, automated highlight generation, tactical review systems, and more.

---

9. Author

- **Name:** Rama Jha
- **GitHub:** github.com/rama-j17
- **Email:** j.rama3617@gmail.com