Project Report: Cross-Camera Player and Referee Detection with Stable ID Assignment

1. Introduction

This project addresses the challenge of detecting and uniquely identifying football players and referees from match footage using a deep learning-based object detection system. The primary objective is to detect these individuals across frames and assign them consistent IDs using visual similarity and spatial heuristics.

Two main tasks were executed:

- **Option 1**: Cross-camera player mapping between two different video feeds (broadcast and tacticam).
- **Option 2**: Re-identification and consistent tracking in a single video feed using tacticam.

This report outlines the methodology, key techniques, results, encountered challenges, and future improvements.

---

2. Approach and Methodology

The solution involves multiple stages in a computer vision pipeline:

2.1 Object Detection

- Used a **YOLOv11** model, trained to detect two custom classes: player and referee.
- The model provides high-accuracy bounding boxes and classification confidence scores.

2.2 Feature Extraction

- Once an object is detected, its bounding box is used to extract the cropped image of the player or referee.
- The cropped image is resized to **64×128**, flattened, and normalized to form a simple feature vector.
- This vector acts as a visual signature for the individual.

2.3 ID Matching Using Similarity

- Extracted features are compared across frames using **cosine similarity**.
- If a new detection matches a previously seen individual (based on feature similarity > 0.85 and IoU > 0.4), the same ID is retained.
- If no match is found, a new ID is assigned.

2.4 Spatial Filtering (Field Detection)

- Applied **HSV color thresholding** to isolate the green field area in the frame.
- This prevents false detections of spectators, camera crew, or substitutes sitting outside the field.

2.5 Cross-Camera Mapping (Option 1)

- Feature vectors from both the broadcast and tacticam videos are extracted for top visible players.
- A similarity matrix is computed to map players from one feed to the other based on visual likeness.

2.6 Output Generation

- Bounding boxes are drawn with unique, consistent IDs.
- Video is saved in MP4 format and made compatible with web browsers via FFmpeg re-encoding.

---

3. Techniques Tried and Their Outcomes

Technique	Purpose	Outcome
YOLOv11	Real-time object detection	Effective for detecting player and referee in most frames
Flattened pixel vector (64x128)	Feature representation for ID matching	Fast to compute, good for basic visual similarity comparison
Cosine similarity	Compare current detection with previous features	Gave reasonably stable ID assignment across frames
HSV green masking	Remove out-of-field detections	Helped reduce false positives from non-field areas
IoU thresholding	Improve match reliability in spatial dimension	Prevented wrong ID matches due to occlusion or perspective shift
Frame skipping	Performance optimization	Maintained speed while preserving sufficient tracking accuracy

---

4. Challenges Encountered

4.1 Detection of Non-Players

- Players outside the main field (bench players, media, spectators) were sometimes detected.
- HSV-based field masking helped reduce these errors, but wasn't perfect under poor lighting.

4.2 Feature Representation Simplicity

- The use of raw pixel features lacks robustness to changes in pose, scale, or lighting.
- More advanced embedding methods like ReID models could significantly improve reliability.

4.3 Player Occlusion and Overlap

- When players overlapped or occluded each other, bounding boxes were imperfect.
- This affected both detection and visual similarity-based matching.

4.4 Performance and Latency

- Processing full-resolution frames with YOLO and cosine similarity in each frame was computationally heavy.
- Optimization was done by resizing inputs and skipping frames periodically.

---

5. If Incomplete, What Remains

Although the project achieved the stated goals, a few areas could be further improved with additional time and resources:

5.1 Advanced Re-Identification Models

- Use **pretrained ReID models** (e.g., OSNet, MobileNet+Triplet Loss) for better feature embedding.
- This would handle pose, angle, and lighting changes better than raw pixel-based vectors.

5.2 Integration with Tracking Algorithms

- Incorporate **Deep SORT** or **ByteTrack** for motion-assisted multi-object tracking.
- Would allow smoother ID continuity even when players disappear and reappear due to occlusion.

5.3 Better Field Detection

- Use **semantic segmentation** models (like DeepLab) to segment the football field precisely.
- Prevents misclassification of non-field entities.

5.4 Speed and Deployment

- Move to a **GPU-based processing pipeline** to handle real-time or near real-time use cases.
- Add options for running on video streams instead of offline files.

---

6. Results and Conclusion

The final system successfully:

- Detected players and referees from both tacticam and broadcast videos.
- Assigned **consistent and stable IDs** across frames using visual features and spatial validation.
- Filtered out many non-player entities using HSV field masking.
- Demonstrated player matching between two camera feeds based on visual similarity.
-

Conclusion

This project effectively combines deep learning-based detection and traditional feature comparison to tackle a real-world sports analytics problem. With further optimization and enhancements, it can be deployed as a robust module in sports broadcasting, player tracking, or tactical analysis pipelines.

Note

For complete implementation, refer to the GitHub notebooks:

- Task_Option_1_Cross_Camera_Player_Mapping_.ipynb
- Task_Option_2_Re_Identification_in_a_Single_Feed.ipynb
  Repo: github.com/Arish005/Soccer-Player-Re-Identification-Assignment

Done By,
Arish K S
Department of Artificial Intelligence & Data Science
Karpagam College of Engineering
Email: arishsmart2005@gmail.com
GitHub: https://github.com/Arish005