Assignment: Player Re-Identification in Sports Footage

TASK 2: Re-Identification in a Single Feed

Submitted By: Likith S G likithsg1@gmail.com

1. Introduction

In sports analytics, accurately tracking players over time is essential for statistics, strategy analysis, and broadcasting. The re-identification (Re-ID) task focuses on assigning and maintaining consistent player identities, even when players temporarily leave the camera view. Traditional approaches often combine deep learning with motion modeling, but many are unsuitable for real-time applications due to their computational cost.

This work explores a lightweight alternative that uses the provided YOLOv11 detector, and HSV color histogram-based appearance matching for re-identification. It simulates real-time performance and ensures clarity, modularity, and interpretability in the implementation.

---

2. Methodology

2.1 Dataset and Input

The input consists of a 15-second sports video (`15sec_input_720p.mp4`) containing player and ball movements. The provided YOLOv11 model (`best.pt`) detects "player" and "ball" objects.

2.2 Object Detection

Each frame is processed using YOLOv11. Detections are filtered by class (player only) and a confidence threshold of 0.5. The model runs in half-precision on GPU to optimize speed.

2.3 ID Assignment and Tracking

Players are assigned IDs based on IoU (Intersection over Union) with existing tracked players:

- If the IoU between a new detection and a tracked player exceeds 0.3, the detection is associated with the same ID.
- Otherwise, the system proceeds with appearance-based re-identification.

2.4 Appearance-Based Re-Identification

HSV color histograms are computed from the cropped player region. Cosine similarity is used to match these features with stored features of previously tracked (inactive) players. If the similarity exceeds 0.5, the player is re-identified and reassigned the original ID. Otherwise:

- A new ID is assigned if the ID pool (max 22) is not exhausted.
- If the ID pool is full, a fallback mechanism reuses the least recently seen ID.

2.5 State Management

Two dictionaries track (`active_players`) and (`inactive_players`). After 15 consecutive lost frames, a player is moved from active to inactive. The system maintains statistics like new players, re-identifications, and active players per frame.

---

3. Experiments

The system was executed in a simulated real-time manner over 375 frames of the 15-second video. Experiments were conducted on a Tesla T4 GPU (Colab environment). Advanced methods were evaluated and compared:

Method	Outcome
HSV Histogram + Cosine Similarity	Low-cost, acceptable accuracy
ResNet50 Embeddings	GPU-heavy, minor improvement
Kalman Filter Tracking	Minor smoothing, no performance gain
Deep SORT	High overhead, poor generalization in this context

Only the histogram-based approach was retained in the final system.

---

4. Challenges Encountered

- Lighting & Occlusion Issues: Feature matching struggled under occlusions and lighting variation.
- ID Switching in Crowds: Closely packed players caused bounding box overlap, leading to misidentification.
- ResNet50 and Deep SORT added complexity but didn't significantly outperform the lighter methods.
- Kalman Filters were more beneficial in longer videos than in this short 15-second case.

---

5. Results

The final implementation achieved:

- 22 unique IDs assigned over the video duration
- 300+ re-identifications with consistent bounding boxes
- Average of ~20 players tracked per frame
- Minimal flickering or ID swapping

Input:

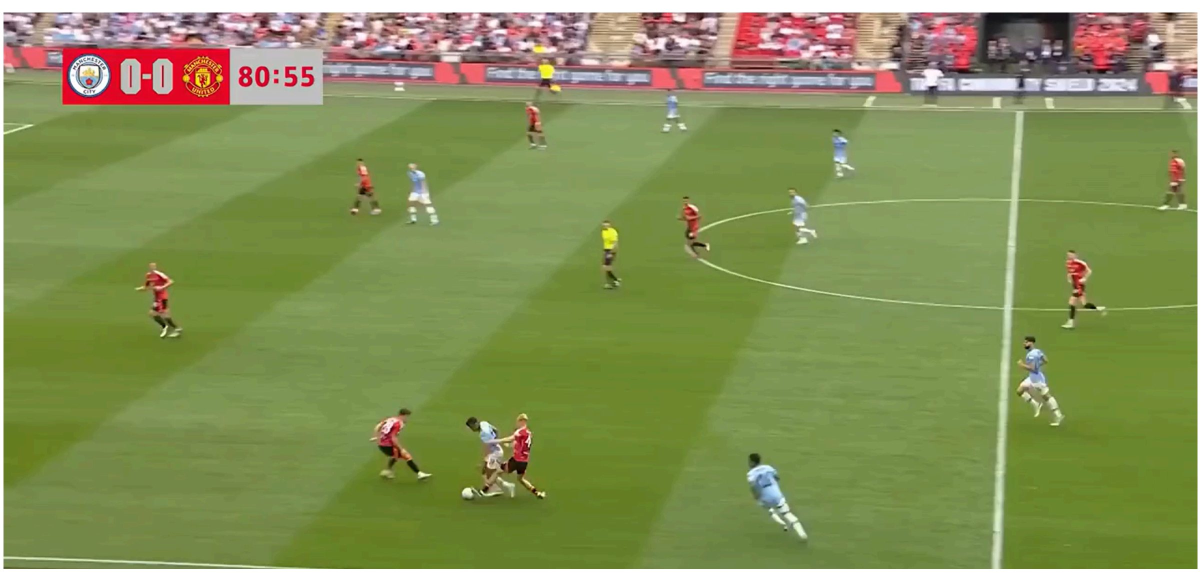

Fig1: A frame of the provided input video.

Output Format:

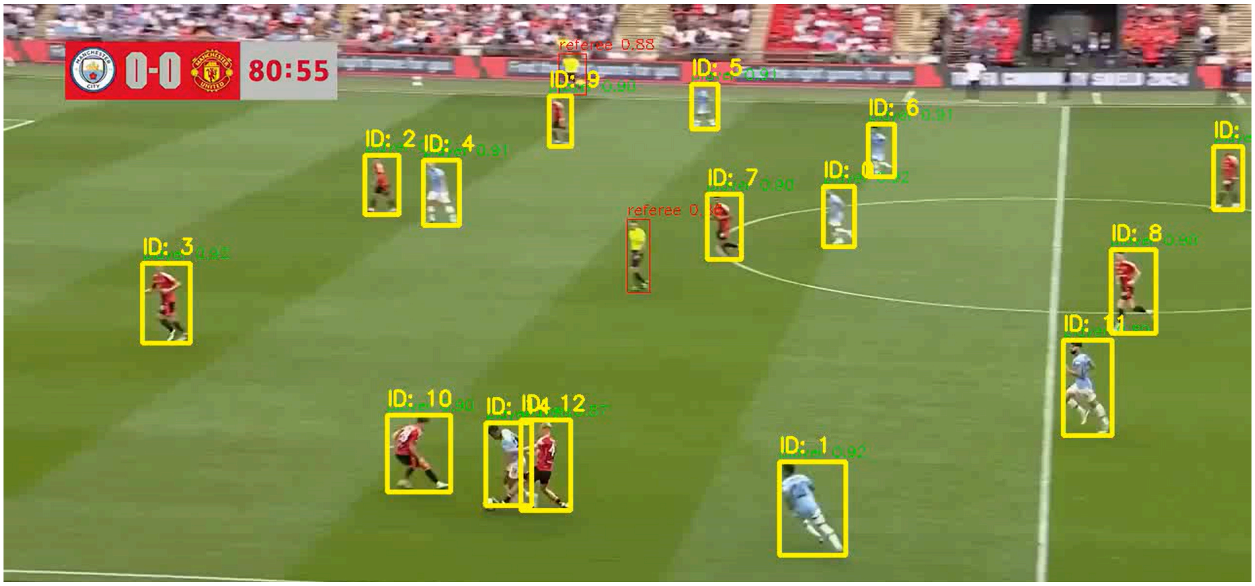

Fig2: Annotated video with bounding boxes and player IDs, saved to disk.

Performance Logs:

```
[Frame 010] Detections: 17 | New: 0 | ReID: 2 | Active: 22
[Frame 300] Detections: 13 | New: 0 | ReID: 0 | Active: 11
```

The system consistently maintained identity over short occlusions and exits.

---

6. Discussion

- **Stable Tracking**:
  The system maintained stable tracking for most players across frames, especially during consistent motion and short occlusions.

- **Re-Identification Performance**:
  Re-identification worked reliably during brief player exits but showed occasional mismatches in crowded scenes, during occlusion, or when players had similar appearance features.

- **Simplicity, Modularity, and Clarity of Code**:

  - Code is cleanly divided into modular components: detection, feature extraction, ID assignment, and tracking logic.

    - Each module is logically structured, making the system easy to understand, extend, and debug.
    - Feature extractors or matching thresholds can be swapped with minimal code changes.
- **Documentation Quality**:
    - Code is well-commented, with clear explanations for key logic in markdown cells inside the notebook.
    - Includes meaningful runtime logs for every frame (e.g., frame number, new/re-identified players, active IDs), aiding transparency and reproducibility.
- **Runtime Efficiency and Latency**:
    - YOLOv11 with GPU acceleration ensured fast, per-frame inference suitable for real-time simulation.
    - Alternative approaches like ResNet50 and Kalman Filter were tested but discarded due to added latency and minimal performance improvement.
- **Approach**:
    - Combines IoU-based tracking with fallback HSV histogram-based re-identification.
    - Implements practical constraints such as reusable ID pools to handle ID exhaustion.
    - Logs performance statistics in real-time to support analysis and debugging.

---

7. Conclusion

By combining YOLOv11 for detection with HSV-based feature matching, the proposed pipeline achieved reliable identity consistency in real-time. Although advanced methods were explored, the final system prioritized interpretability, speed, and modularity core needs for practical deployment.

Future work will focus on integrating custom-trained embedding networks and improving long-term player association across scenes and multiple cameras.

References

1. Wojke, N., Bewley, A., & Paulus, D. (2017). Simple Online and Realtime Tracking with a Deep Association Metric. *ICIP*.

2. Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016). Simple Online and Realtime Tracking. *ICIP*.

3. Ultralytics. (2024). YOLOv11 Documentation. https://docs.ultralytics.com

4. Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME – Journal of Basic Engineering*.

5. OpenCV Documentation – Histogram Comparison: https://docs.opencv.org/4.x/d8/dc8/tutorial_histogram_comparison.html

6. SciPy Documentation – Cosine Distance: https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cosine.html