Player Re-Identification and Tracking System

Objective

The goal of this project was to build a **player re-identification and tracking system** for a single football video feed. The system should consistently assign the **same ID to each player**, even when they leave and re-enter the frame.

Approach & Methodology

1. Detection

- Used **YOLOv11** (Ultralytics) as the base object detection model.
- Model was **fine-tuned** specifically to detect **players** and the **ball** from football match footage.

2. Tracking Pipeline

- Combined **motion-based tracking** with **appearance-based tracking**:
    1. **Kalman Filter:** For predicting the next position of players based on previous motion.
    2. **IoU Matching:** Bounding box overlap used for short-term frame-to-frame association.
    3. **Re-Identification (ReID) Model:**
        - Used **TorchReID** with **OSNet** architecture to compute feature embeddings from player crops.
        - Appearance-based matching enabled re-association even when players disappeared and reappeared.

3. ID Assignment

- Each player was assigned a **human-readable ID** like Player 1, Player 2, etc.

- IDs were retained throughout the video unless the player permanently left the frame.

4.Output

- Generated a processed .mp4 video with:
  - Bounding boxes around each player.
  - Persistent labels (Player N) displayed on each player.

Techniques Tried & Their Outcomes

Technique	Outcome
YOLOv11 + IoU Only (SORT-like)	IDs frequently switched when players crossed paths or occluded.
YOLO + StrongSORT (off GitHub)	Setup was heavy, reproducibility issues in Colab/Kaggle.
YOLO + DeepSORT with Kalman	Improved but still failed with long occlusions or exits.
YOLO + TorchReID + Kalman + IoU	**Final approach — stable IDs, works even with occlusion or re-entry.**
Post-processing with smoothing	Reduced jitter in bounding boxes, improved visual quality.

Challenges Encountered

1. ID Switching Problem

- **Cause:** IoU-only matching failed when players crossed, overlapped, or when their appearance was similar.
- **Solution:** Added ReID feature embedding to complement IoU.

2. Ghost Boxes & False Tracks

- Some detections were lost temporarily, resulting in ghost tracks or blank boxes.

- Resolved by tuning:
  - MAX_MISSING_FRAMES
  - Kalman filter noise parameters
  - Minimum confidence thresholds for YOLO

3. Compute Limitations (GPU Availability)

- Faced GPU restrictions in **Google Colab** and **Kaggle**.
- Solution: Shifted processing to **local machine with GPU support (VSCode + CUDA)**.

4. ID Complexity

- UUID-based IDs were unreadable for humans (9a3f8e1b).
- Solved by mapping UUIDs to simple labels like Player 1, Player 2.

5. File Corruption Issues

- Some .mp4 files wouldn’t play due to improper codec settings in OpenCV.
- Fixed by switching to 'mp4v' codec and correct VideoWriter settings.

Key Outcomes

- ✅ Achieved persistent tracking with readable labels.
- ✅ Significantly reduced ID switching compared to naive IoU-only tracking.
- ✅ System works smoothly for **single-camera sports videos.**

Future Improvements

- Add player trajectory visualization (lines showing movement).
- Export logs as CSV for analytics.
- Support for multi-camera ReID tracking.

- Deploy as a real-time application or web dashboard.

Conclusion

The final pipeline, combining **YOLOv11 + TorchReID + Kalman Filter + IoU**, successfully tracks players with stable IDs in football videos. The system balances accuracy and computational efficiency while being robust against occlusions and re-entries.

-Ch.M.Karhthikeya