⚽ Player Re-Identification in Sports Footage

This project focuses on **player detection**, **re-identification**, **team classification**, and **basic action recognition** using a short football match video. It integrates object detection (YOLO), tracking (SORT), vision-language models (CLIP), and 3D CNNs for action detection (R3D).

📁 Repository Contents

```
bash
CopyEdit
.
├── models/
│   └── yolo_players.pt           # YOLOv8 model fine-tuned for player and
ball detection
├── sort/
│   └── sort.py                   # SORT multi-object tracking algorithm
├── videos/
│   └── 15sec_input_720p.mp4      # Input match footage (15 seconds)
├── results/
│   └── final_clip_output.mp4     # Output video with IDs, teams, and actions
├── detect_players.ipynb          # Optional notebook for step-by-step
debugging and visualization
├── Player Re.docx                # This report and project documentation
```

⚙️ Setup Instructions

🔧 Requirements

- Python 3.8+
- GPU (recommended for real-time CLIP processing)

🎁 Install Dependencies

```
bash
CopyEdit
pip install torch torchvision opencv-python numpy pillow scikit-learn
git+https://github.com/openai/CLIP.git ultralytics
```

Ensure the YOLOv8 model is saved at: `models/yolo_players.pt`

▶ Run the Project

```
bash
CopyEdit
python vlm_action_tracking.py
```

This will process the input video and display:

- Persistent player IDs even after re-entry
- Team label based on jersey color
- Action tag (if enough frames available)
- Ball detection with label

📽 Project Objective

To detect and track players and the ball in football footage, assign each player a **persistent identity** (even if they leave and re-enter the frame), classify their **team based on jersey color**, and detect basic **player actions** using temporal frame buffers.

🔍 What I Did

1. Player & Ball Detection

- Used a **YOLOv8 model** trained specifically for detecting football players and the ball.

2. Tracking & Re-Identification

- Implemented **SORT** for short-term object tracking.
- Replaced simple ID assignment with **CLIP-based re-identification**:
    - Each new player's jersey crop is converted to a CLIP embedding.
    - If a visually similar player is already tracked (via cosine similarity), we reassign the same ID.
    - Ensures persistent IDs even after re-entry.

3. Team Classification

- Extracted the top 40% of each player's bounding box (jersey region).
- Used **OpenAI CLIP** with prompts like:
    - `"a football player wearing a red jersey"`
    - `"a football player wearing a blue jersey"`
- Compared the image embedding with prompt embeddings to assign Team A or B.

4. Action Recognition

- Maintained a buffer of the **last 16 cropped frames per player**.
- Passed this through a **pretrained R3D (3D CNN)** model from `torchvision` to predict basic action classes (e.g., movement, idle).

✅ What Worked & What Didn't

Method	Description	Result
HSV Histogram	Basic color segmentation	❌ Too sensitive to lighting
KMeans Clustering	Color-based clustering	❌ Unstable in real match footage
CLIP (VLM)	Natural language + jersey crop embedding	✅ Worked reliably for team IDs
SORT	Kalman filter + IOU matching	✅ Good short-term tracking
CLIP + SORT	Added jersey embeddings for re-identification	✅ Persistent ID worked well
R3D Action Model	Frame buffer-based CNN	✅ Decent action detection on CPU/GPU

⚠️ Problems Faced

- CLIP model was slow on CPU → switched to GPU.
- SORT loses ID on long occlusion → fixed with CLIP-based ID re-assignment.
- R3D requires **exactly 16 frames** to predict → needed consistent buffer management.
- Team color recognition struggles when jersey not visible (e.g., back facing camera).

🔧 Future Improvements

- 🔁 Replace R3D with a more robust temporal model (e.g., SlowFast or TimeSformer).
- 🧍 Add face/jersey number recognition for **known player** identification.
- 🏃‍♂️ Extend team detection to identify **player roles** (e.g., goalkeeper, striker).
- 📊 Add analytics overlay: passes, tackles, or goal attempt detection.
- 🎙️ Add auto-generated **commentary or event tagging** using time + action + team data.

My Experience

This was my first time using **CLIP for visual classification tasks** and combining it with **tracking and detection pipelines**. I have previously worked with OpenCV and YOLO, but integrating temporal models and vision-language models gave me a broader perspective of multimodal AI.

While the system isn't perfect yet, it demonstrates a **real-time sports analytics pipeline** combining:

- Object Detection
- Re-identification
- Language-based Vision Classification
- Action Understanding

Contact

Feel free to reach out for any clarifications, collaboration, or feedback on the project!