Player Re-Identification – Single Feed Tracking

GITHUB LINK: https://github.com/AakashYadv/Re-Indetification

Objective

To assign consistent player IDs throughout a 15-second football video using a YOLOv11 detection model. The goal is to ensure that players who go out of the frame and reappear later are tracked with the same identity.

Approach and Methodology

1. **Detection**:
    - Used a fine-tuned YOLOv11 model trained to detect:
        - Players
        - Ball
        - Referees
2. **Tracking**:
    - Integrated **DeepSORT** (or StrongSORT without Re-ID embedder) to maintain object identities over time.
    - The initial few seconds were used to assign IDs based on tracking heuristics.
3. **Frame-by-Frame Processing**:
    - The input video was processed using OpenCV to feed frames into the model and tracker sequentially.
    - The system simulated a real-time tracking pipeline without batching.

⚙ Techniques Tried and Their Outcomes

- **YOLOv11 + DeepSORT (without Re-ID)**:
    - Performed well in most frames where players stayed within camera view.
    - Tracked players consistently when their movement was smooth and unobstructed.
    - ID assignment broke occasionally if players exited for long durations.
- **No appearance-based Re-ID (e.g., osnet_x1_0)**:
    - As Re-ID was not used, ID re-association relied solely on motion and proximity, reducing accuracy in crowded or occluded scenarios.

Challenges Encountered

- **No GPU**:
    - Inference was slower on CPU, especially for real-time frame-by-frame tracking.
- **Re-ID limitations**:
    - Without embedding vectors, tracking IDs could not recover properly after long occlusions or when players swapped positions.
- **Model integration**:
    - Ensuring compatibility between YOLOv11 outputs and DeepSORT input formats required custom mapping and filtering.

Incomplete or Future Work

If given more time or compute resources:

1. **Add Appearance Embedding (Re-ID)**:
    - Integrate `osnet_x1_0.pth` for stronger identity consistency.
    - Improve player re-identification after occlusion or long exits.
2. **Optimization**:
    - Use frame skipping or async pipelines to improve CPU performance.
    - Deploy using FastAPI or Render for web-based testing.
3. **Visualization**:
    - Improve ID annotations on video (color coding, trails, etc.)

Reproducibility Note

This submission is self-contained and includes:

- `main.py` with full tracking code
- Pretrained model placeholder (`models/yolov11_player.pt`)
- Dependency list (`requirements.txt`)
- Setup instructions in `README.md`

You can run the system by placing the input video and model file correctly and executing:

```
python main.py
```