Soccer Player Re-Identification Assignment

(Option 2)

1. Approach and Methodology

The objective was to detect and track soccer players, referees, and the ball from a match video using object detection and tracking techniques.

- Model Used: Ultralytics YOLO was used for object detection. A pretrained model which was provided in the assignment (model.pt) was loaded.
- Tracking: Supervision's implementation of ByteTrack was used for object association and track continuity across frames.
- Pipeline:
    1. Read frames from the input .mp4 video.
    2. Batch-detect objects using YOLO with a confidence threshold of 0.3.
    3. Post-process detections to merge "goalkeeper" labels into "player".
    4. Run ByteTrack to assign consistent track IDs to players and referees.
    5. Store and reuse tracks using pickle-based stubbing to save time on reruns.
    6. Annotate the frames with ellipses and unique IDs.
    7. Save the annotated output back as a video.

2. Techniques Tried and Their Outcomes

- YOLO Object Detection:
    - Successfully identified players, referees, and the ball.
    - Provided bounding boxes and class IDs.
- ByteTrack for Multi-Object Tracking:
    - Ensured frame-to-frame identity persistence for players and referees.
    - Track IDs were displayed as overlaid rectangles with IDs.
- Visualization:
    - Ellipses were drawn under players and referees for better visibility.
    - Track IDs were rendered within filled rectangles below the ellipses.
- Stubbing:
    - A caching mechanism using pickle was implemented to avoid repeated inference during debugging.

3. Challenges Encountered

- Bounding Ellipses Placement: Initially, ellipses were not aligned correctly and appeared oversized. This required manual tuning of ellipse dimensions.
- Detection Quality: YOLO occasionally misclassified players as goalkeepers. This was addressed by relabeling all "goalkeepers" to "players".
- No Ball Tracking: ByteTrack does not track the ball as it's small and fast-moving; the ball's bounding box was instead taken directly from YOLO detections.
- Stub Reusability: Ensuring track_stubs.pkl syncs correctly with the latest model output and video frames was tricky during experimentation.

4. Remaining Work and Future Steps

- Incomplete Re-Identification (ReID): Deep SORT or custom embedding models could be added to maintain track ID consistency even after occlusions or reappearances.
- Ball Tracking with Motion Models: The ball tracking could be improved using a Kalman filter or optical flow methods due to its speed and size.
- Team Classification: Using jersey color or segmentation, team-level info could be extracted.
- Performance Optimization: Frame batching could be tuned and multiprocessing added to handle full-length matches.

Due to a time crunch, these functionalities could not be added, though I was very excited and willing to do so if time allowed.

5. Additional features beyond project scope

- Real-Time Dashboard: Integrate live visualizations with player stats and heatmaps during match playback.
- Ball Interpolation: Smooth out missing or noisy ball detections using interpolation techniques like Kalman filtering or spline fitting.
- Camera Movement Estimator: Use background keypoint tracking or homography estimation to decouple camera motion from player movement.
- Perspective Transformer: Warp the view to a bird’s-eye (top-down) perspective using homography for better spatial analysis.
- Speed and Distance Tracker: Calculate player and ball velocity and distance covered using pixel-per-meter estimation or calibrated field geometry.