Challenges Faced While Working on the Project

- The first pipeline idea was to actually have **YOLO + FairMOT**.
- Due to ambiguity in the FairMOT documentation, I transitioned to the **YOLO + BOTSORT + OSNET** pipeline:
    - **YOLO** for identification
    - **BOTSORT** for tracking
    - **OSNET** for re-ID
- BOTSORT turned out to be more challenging to integrate than expected, which led me to fall back to **ByteTrack** for the tracking component.
- For the re-identification task:
    - I had to search the net and finally found something obvious: the **TorchREID** module, which helps in feature extraction and mapping.
- Had many branched ideas, like:
    - Custom training the **Torchreid engine (OSNet)** on **SoccerNetv3_reid** for more accurate mapping.
    - Had to **backtrack** due to time constraints on training and data preprocessing.

Challenges Overcome

- The first run had an execution time of **~20 mins (abysmal time)**:
    - Caused by an **inefficient pipeline** maintaining `global_tracks`, which was iterated through for every new frame.
    - Time complexity was **exponential** (imo).
    - Also caused **all players to have the same GID of 0**.
- Solved by:
    - Maintaining a **sliding window approach** using `deque` of limited size reducing time to about ~110 seconds.
    - Tuning the **threshold hyperparameter** for proper working.

- During final runs:
    - The model struggled to identify the **goalkeeper and the referee** in YOLO tracking.
    - This led to `global_id_tracks` reassigning **90+ new IDs** to just the goalkeeper (and also to the ball).
- Solved by:
    - Changing the logic in the **track function** to:
        - Map **referee** and **goalkeeper** to player ID.
        - **Ignore ball detection** to reduce redundant re-IDs.

Final Result Obtained

Due to video quality or my own lack of skills:

- In the final video:
    - **Players who exited and reentered the frame** were given new global IDs.
- Count of Global IDs Assigned:
    - 🟦 **Blue Team**: 32 global IDs
    - 🔴 **Red Team**: 13 global IDs

Side Notes

1) Thought Process for Task 1

I have already thought of a general pipeline for the multi-angle player mapping using the **pre-existing `tracker.py` module class** itself:

- Generate `tracks`, `global_id_tracks`, etc., for the **first video**.
- Only generate tracks for the **second video** and:
    - Use **TorchReID** to perform **similarity matching** with the first video's **last set of `global_id_tracks`**.
- Of course, save both the videos.

2) Personal Note

- On the same day the assignment was released, I suffered a **physical setback (injury in the gym)**.
- This made me **unable to sit and work** on the assignment for more than **30–45 minutes at a time** without major discomfort.

🙏 Closing Note

Thank you for the opportunity to work on this task. I thoroughly enjoyed:

- Building this pipeline
- Exploring new tools
- Iterating on ideas

I look forward to hearing from you and hope to contribute further in the future.