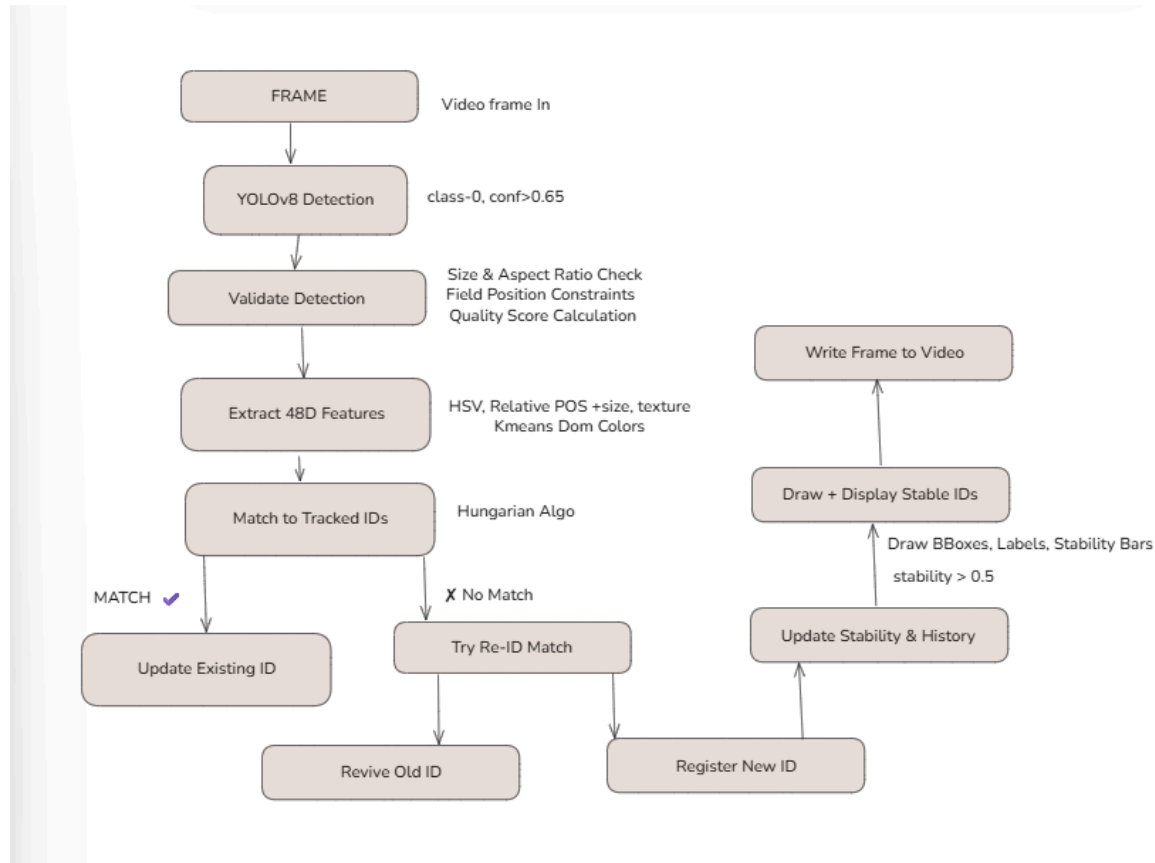


Project Objective

To track football players across match footage using YOLOv8 and assign consistent player IDs — even when players exit, re-enter, or interact with others during gameplay.

Approach & Methodology




I built a custom AdvancedPlayerTracker with the following pipeline:

- **Detection:** Used YOLOv8 to detect players in each frame. Applied rules based on size, position, and confidence to filter noise.
- **Feature Extraction:** For each valid detection, extracted a 48-dimensional signature using HSV histograms, KMeans clustering (for dominant color), spatial data, and texture descriptors.
- **ID Matching:** Matched current frame detections with tracked players using the Hungarian algorithm. For missing players, attempted re-identification based on stored feature vectors.
- **ID Assignment & Drawing:** Tracked and rendered only stable IDs to avoid flicker and reassignment.

Current Challenge

I'm still facing ID instability when players crowd or overlap — e.g., during corner kicks. Despite velocity and color-based tracking, the tracker often resets or swaps identities post-occlusion. This remains the key bottleneck.

Techniques We Tried

 Technique	Outcome
YOLOv8 Detection	Very stable and accurate bounding boxes
Feature-Based Matching (HSV, KMeans)	Works well in most frames
Disappeared Player Re-ID	Partially works, fails during occlusion
Velocity Tracking	Helps short-term identity preservation
Bounding Box Filtering	Improved robustness of detections

What’s Left & My Plan

If I had more time and compute resources, I’d focus on:

- **Deep Re-ID Embeddings:** Replace handcrafted HSV features with deep appearance models like OSNet or FastReID.
- **Temporal Memory:** Track player sequences over time using buffered tracklets, not isolated frames.
- **Collision-Aware Logic:** Detect occlusions and pause ID updates until players reappear distinctly.
- **Optical Flow:** Leverage frame-wise motion patterns to support identity continuity.

What’s Working Well

- **Player detection:** YOLOv8 is highly accurate
- **Feature extraction:** HSV-based signatures are stable
- **Tracking:** Frame-by-frame consistency is decent
- **Output:** JSON + annotated video frames are generated
- **Basic re-ID:** Disappeared player recovery works, but needs refinement

Final Thoughts

We’re almost there! The tracker rocks in low-density frames but struggles in tight crowds. I’m confident that adding deep appearance features and memory-based tracking will fix the last bit.

This project deepened my understanding of computer vision pipelines and taught me how to balance handcrafted logic with learned representations. I also learned how real-world constraints like occlusion and resource limits shape ML solutions.