**Assignment Report:**

**1. Objective**

To detect and track multiple footballs during player drills and dynamically classify each ball as either:

* **Stationary** – used as regional markers
* **Action** – actively involved in the drill

The final output overlays bounding boxes, object IDs, and classification states (ACTION/STATIONARY) on the video.

**2. Models Used**

* **YOLOv8s (yolov8s.pt)**: For detecting "sports ball" objects in each frame.
* **ByteTrack (bytetrack.yaml)**: For object tracking and maintaining consistent object IDs across frames.
* **YOLOv8s-pose (yolov8s-pose.pt)** *(optional)*: For pose estimation and overlaying skeleton keypoints on players (not mandatory for the task).

**3. Methodology**

**3.1. Object Detection & Tracking**

* Frames are passed to the YOLOv8s model to detect all instances of sports ball.
* The **Ultralytics track()** API integrates **ByteTrack** to assign unique IDs and track the same ball across multiple frames.
* The bounding box, confidence score, and tracking ID are captured for every detection.

**3.2. Velocity-Based State Classification**

To distinguish between ACTION and STATIONARY balls:

* For each detected ball ID, the centroid (x, y) of the bounding box is stored in a **deque** of fixed size (e.g., 10 frames).
* The **average velocity** is computed using Euclidean distance over stored centroids.
* A simple threshold-based logic is used:
  + **Velocity ≥ 2.0 px/frame** → ACTION
  + **Velocity < 2.0 px/frame** → STATIONARY

This approach generalizes well across different videos and does not rely on object class changes or external context.

### 3.3. Challenges Faced

• **Initial Heuristics Attempted:**

* **Direction Shift:** Detected large angular deviations in trajectory across 3 consecutive frames.
* **Area Variation:** Observed rapid fluctuations in bounding box size, indicating possible occlusion or misclassification.  
  • A **scoring mechanism** (e.g., score ≥ 2) was introduced based on these heuristics to classify a ball as ACTION.  
  • While these checks helped refine edge cases (e.g., a stationary ball nudged briefly), they were **ultimately replaced** by a **pure velocity-based approach** for simplicity, better generalization, and reduced false positives.

**3.4. Visualization and Overlay**

Each frame is rendered with:

* Bounding boxes around balls
* Ball ID, STATE (ACTION/STATIONARY), and Velocity
* *Tried* Pose skeletons for players using yolov8s-pose.pt

Final videos are saved as annotated .mp4 outputs.

**4. Tracking Techniques & ID Consistency**

* **ByteTrack** effectively maintained object ID consistency even during:
  + Temporary occlusion
  + Ball overlap
  + Fast movement across frames
* To reduce noise and false detections:
  + Objects with **confidence < 0.4** were ignored
  + Objects with **area < 8000 pixels** were filtered out
* These steps prevented ID flickering and ensured stable tracking across sequences.

### ****5. Observations & Known Limitations****

While the overall tracking and classification performed consistently across both test videos, the following edge cases were observed:

* **Bounding Box Fluctuation (Stationary Ball - Left Side):**  
  In certain frames, a stationary sports ball on the left side exhibited minor bounding box jitter. This is attributed to subtle lighting or background texture changes, leading to occasional detection instability.
* **Labeling Error (Terminal Output):**  
  In a few instances, the action ball was momentarily misclassified in the terminal logs (e.g., labeled as “sheep”). This likely stems from the pre-trained YOLOv8s model’s default class vocabulary, where certain white/round objects might be matched to similar shapes under confidence thresholds.

⚠️ Note: These were not reflected in the visual overlays and did not significantly impact ID tracking or classification.

6. File Structure

├── test.py # Main script for detection, tracking, and classification

├── videos/

│ ├── test.mp4 # Input video 1

│ ├── test2.mp4 # Input video 2

├── outputs/

│ ├── output1\_annotated.mp4 # Resulting annotated video

│ ├── output1\_annotated.mp4

├── yolov8s.pt # YOLOv8s model for ball detection

├── yolov8s-pose.pt # (Optional) YOLOv8s for pose estimation

├── README.md # Project instructions

├── Report.doc #Project documentation