**Real-Time Detection of Object Missing and New Object Placement in Video**

This project focuses on developing a **real-time object detection and tracking system** using **YOLOv8** (You Only Look Once, version 8) for object detection and **DeepSORT** (Simple Online and Realtime Tracking with a Deep Association Metric) for object tracking.  
The objective is to accurately detect multiple objects in each frame of a video or webcam feed and track their movements across consecutive frames by assigning a unique ID to each object.

The system is designed to:

* Detect objects frame-by-frame using a deep learning model (YOLOv8n - Nano version for faster performance).
* Track detected objects over time using DeepSORT, maintaining consistent IDs even when objects temporarily disappear.
* Calculate and display the real-time **FPS (Frames Per Second)** achieved during execution, reflecting system efficiency.
* Log events such as **new object detection** and **missing objects** with timestamps.

The project aims to balance accuracy and speed, achieving decent real-time performance even on modest hardware setups.

**FPS Achieved (Real-Time Performance)**

The system calculates FPS dynamically during execution.

* **Average FPS over runtime:**
  + 5.1 to 5.9 FPS (based on different runs)
* **FPS samples recorded during testing:**
  + 5.63 FPS
  + 5.91 FPS
  + 4.36 FPS
  + 4.81 FPS
  + 6.82 FPS

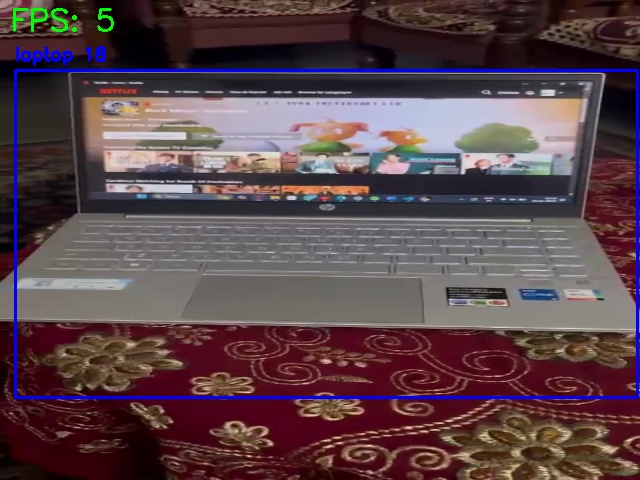
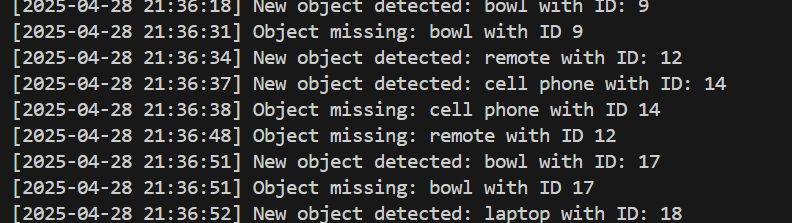
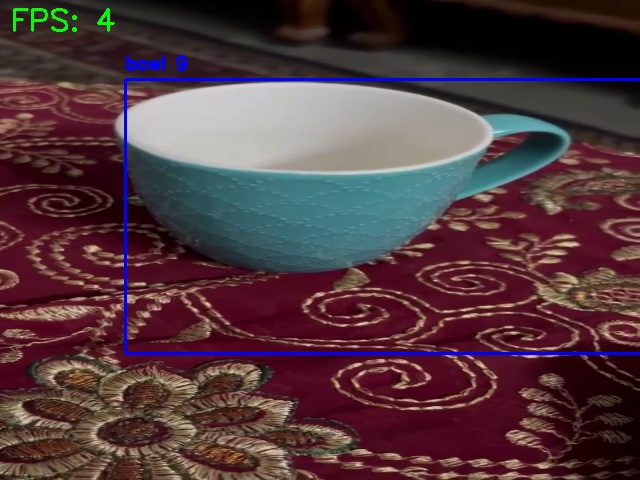
**Minimum FPS observed:** 4.36

**Maximum FPS observed:** 6.82

**Average FPS (approximate overall):** **~**5.2 **FPS**

This performance indicates that the system is capable of handling real-time object detection and tracking at moderate speeds, suitable for basic real-time applications on CPU-only environments.

**Sample Outputs:**

********

**Hardware Configuration Used for Testing**

| **Component** | **Specification** |
| --- | --- |
| **Processor (CPU)** | Intel Core i5 (10th Gen) |
| **Graphics (GPU)** | Integrated Intel UHD Graphics (No dedicated GPU) |
| **RAM** | 8 GB DDR4 |
| **Storage** | 512 GB SSD |
|  |  |

All processing was done on **CPU** without GPU acceleration.

The YOLOv8 **Nano model (yolov8n.pt)** was used to optimize speed.

**Additional Techniques and Architectural Decisions**

* **Model Selection:**  
  Used yolov8n.pt (Nano version) for faster inference on low-resource hardware.
* **Frame Resizing:**  
  Input frames were resized to 640x480 resolution to speed up detection without significant loss in accuracy.
* **DeepSORT Configuration:**  
  Set max\_age=30 in DeepSORT to allow minor occlusions without losing track IDs.
* **Dockerization:**  
  The project includes a **Dockerfile** for containerized deployment to ensure environment consistency across different systems.
* **Logging:**  
  Console logs were added for tracking new and missing objects with precise timestamps.