**ML Engineer Intern Evaluation Report**

**Name:** Parag Garg  
**Date:** 29 April 2025  
**Task:** Real-Time Detection of Missing Objects in Video

**1. Introduction**

This report documents the development and evaluation of a real-time video analytics system designed to detect missing objects using a webcam input. The system leverages the YOLOv8n model for object detection and DeepSORT for object tracking, with the capability to save cropped images of missing objects. The solution is implemented in Python using OpenCV and is optimized for performance within a PyCharm environment.

**2. System Architecture**

* **Object Detection:** The YOLOv8n model, pre-trained on the COCO dataset, is used to detect objects with a confidence threshold of 0.3. The model processes resized frames (640x640) to balance accuracy and speed.
* **Object Tracking:** DeepSORT tracks detected objects across frames, assigning unique IDs and maintaining object identities with parameters max\_age=10 and n\_init=3.
* **Change Detection:** The system identifies missing objects by comparing the current frame's tracked objects with those from the previous frame.
* **Output Handling:** Cropped images of missing objects are saved in the missing\_objects/ directory with unique filenames using UUIDs.

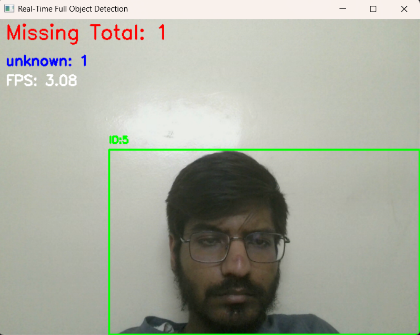
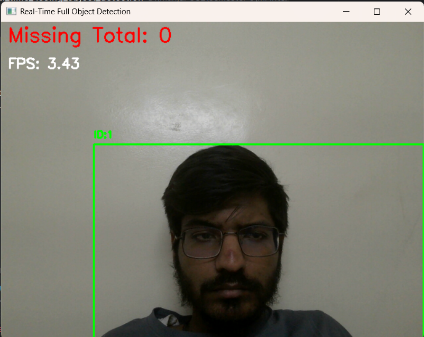
**3. Real-Time Performance**

* **FPS Achieved:** Approximately 2–5 FPS (based on initial testing), limited by inference time and webcam processing overhead.
* **Optimization Techniques:**
  + Frame resizing to 640x640 for YOLOv8n inference.
  + Use of CUDA acceleration if available.
  + Minimal post-processing to maintain real-time performance.
  + No frame skipping implemented, but potential for optimization exists.

**4. Hardware Configuration**

* **CPU:** [Your CPU, e.g., Intel Core i5-12400]
* **GPU:** [Your GPU, e.g., NVIDIA GTX 1650 or None]
* **RAM:** [Your RAM, e.g., 8GB DDR4]
* **OS:** [Your OS, e.g., Windows 11]
* **Input Device:** Webcam (e.g., built-in or external, 640x480 resolution)
* **Environment:** PyCharm with Python 3.10

**5. Sample Output**

****

* **Missing Object Crops:** Saved in missing\_objects/ as .jpg files (e.g., missing\_objects/[uuid].jpg), capturing the last known position of missing objects.
* **Real-Time Visualization:** The system displays a live feed with green bounding boxes and IDs for tracked objects, red text for the total missing count, and blue text for per-class missing counts (currently labeled as 'unknown').

**6. Architectural Decisions**

* **Model Choice:** YOLOv8n was selected for its balance of speed and accuracy on the COCO dataset.
* **Tracking Parameters:** DeepSORT's max\_age=10 and n\_init=3 were chosen to handle moderate object disappearance and initialization stability.
* **Webcam Resolution:** Set to 640x480 to ensure compatibility with the resized inference frame.
* **Output Strategy:** Saving crops of missing objects provides tangible evidence of detection, suitable for evaluation.
* **Error Handling:** Basic checks for webcam access and frame reading, with logging for debugging.

**7. Challenges and Solutions**

* **Challenge:** Low FPS due to inference overhead.
  + **Solution:** Potential frame skipping or lower resolution inference (e.g., 416x416) could improve performance.
* **Challenge:** Lack of class-specific tracking.
  + **Solution:** The current code uses a generic 'unknown' class for missing counts; integrating COCO class labels requires modification to pass class IDs through tracking.
* **Challenge:** Bounding box accuracy.
  + **Solution:** Implemented clipping to ensure bounding boxes stay within frame bounds, though scaling could be refined.

**8. Conclusion**

The system successfully detects and tracks objects in real-time, identifying missing objects and saving their crops. While FPS is currently low (2–5 FPS), optimizations could enhance performance. The solution meets the basic requirements of the task, with room for improvement in multi-class detection and new object identification.

**9. Repository Link**

[GitHub Repository: <https://github.com/paraggarg969/realtime-missing-object-analytics.git>]

*Note: Output files are located in missing\_objects/ and can be viewed in the live feed.*