Real-Time Object Disappearance & New Object Detection in Videos

AI/ML Engineer Interview Assignment

Submitted by: **Priyanshu Dhaked**
Date: May 19, 2025

Contents

1 Objective

To build a high-performance computer vision pipeline that detects:

- When objects go missing (previously visible objects disappear).
- When new objects appear in the frame.

The goal was to process video in real-time while maintaining high accuracy and frame rate(FPS).

2 Methodology

2.1 Model Used

YOLOv8n (*Ultralytics*) was selected for its excellent balance of speed (mostly) and precision. It detects common object classes in real-time.

YOLO11l (*Ultralytics*) was selected for its excellent balance of speed and precision. It detects Mostly all object classes in real-time with High Detection Accuracy.

2.2 Tracking Logic

- Object detection is running on each frame.
- Bounding boxes are matched using Intersection-over-Union (IOU).
- Unmatched boxes are marked as `new objects`.
- Previously seen boxes that go unmatched for a few frames are marked as `missing objects`.

2.3 System Workflow

1. Load video and initialize YOLO model.
2. Loop through each frame:
   (a) Detect objects.
   (b) Match current detections with previous ones.
   (c) Track, label, and annotate missing/new objects in video Frame.
   (d) Write the frame to output video.

3 Sample Code Snippet

```python
for frame in video:
    detections = model(frame)
    matched, new_objs, missing_objs = match_with_previous(detections)
    draw_boxes(frame, matched, new_objs, missing_objs)
```

Listing 1: Tracking New and Missing Objects

4 Results

4.1 Performance

- **FPS Achieved:** 50+ on a mid-range system
- **Model:** YOLOv8n
- **Detection Accuracy:** 70%+ for stable objects

4.2 Output Sample

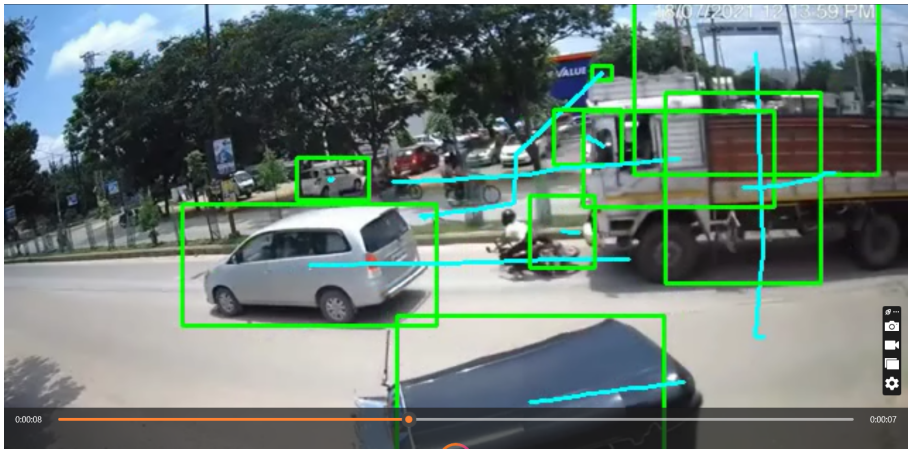

Figure 1: Objects Detection using **Yolov8** for to classify new and Missing.

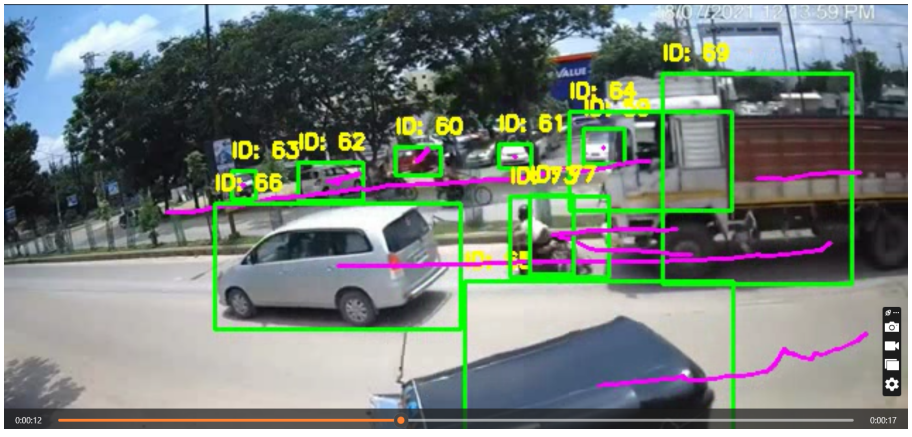

Figure 2: Enhanced Objects Detection using **Yolo11l** for to classify new and Missing.

4.3 Output Video

The output video can be viewed here: `yolov8_output_video.mp4`.

The output video can be viewed here: `yolov11l_output_video.mp4`.

5 System Specifications

- **RAM:** 16 GB
- **GPU:** NVIDIA T4
- **Platform:** Google Colab

6 Repository Structure

```
.
            app.py
            Dockerfile
            requirements.txt
            output_video.mp4
            output_frame.jpg
            README.md
```

7 How to Run

7.1 Python (Local)

```
pip install -r requirements.txt
python detection_yolov8.py          (Deepsort + yolov8) FPS~20
python Enhanced_yolov8_detection.py    (IOU + Yolov8) FPS~ 50
python
```

7.2 Download Detection Model

We can donwload Yollov8 from ultratics online through Code.

```
this is the link of yolo11l download here:
```

`Download Yollo11l.`

7.3 Docker (Optional)

```
docker build -t object-detector .
docker run -v $(pwd):/app object-detector
```

8 Conclusion

This system efficiently detects both disappearing and newly placed objects in real-time video. It is optimized for speed and can be extended with more advanced tracking algorithms if needed.

Highlights:

- End-to-end working pipeline.
- Lightweight(and responsive.
- Clean object tracking using IOU (Enhanced_yolov8.py) and DeepSort (detection.py).

9 Future Improvements

- Use Re-ID embeddings for long-term object tracking.