**Project Report Submission**

**Intel Unnati Industrial Training 2025**

**Project Title: Create pipeline (detect, decode and classification) using DL Streamer, define system scalability for Intel HW**

**Team Members**

**Idhikash Raja, 2nd year, B.Tech/ECE**

**Manaas M, 2nd year, B.Tech/ECE**

**Lalit Kishore V, 2nd year, B.Tech/ECE**

**B S Abdur Rahman Crescent Institute of Science and Technology**

**Team Mentor**

**Dr V Jean Shilpa**

**Associate Professor**

**Department of ECE**

**B S Abdur Rahman Crescent Institute of Science and Technology**

**Introduction**

This project aims to build a real-time deep learning video pipeline using **Intel® DL Streamer**, focusing on three stages: video decoding, object detection, and classification of detected objects. The system is evaluated for scalability on Intel hardware platforms (CPU and GPU) by analysing frame rates, the number of video streams handled, and resource utilization.

**Tools and Technologies Used**

* **DL Streamer** (Intel® Video Analytics Framework)
* **OpenVINO™ Toolkit** (Optimized model inference)
* **GStreamer** (Multimedia pipeline framework)
* **Ubuntu 22.04**, **Python**, **FFmpeg**, **VAAPI**
* **Intel® Core™ i7 CPU**, **Intel® Iris® Xe GPU**

**Models Employed:**

* Detection: person-vehicle-bike-detection-crossroad-0078
* Classification: vehicle-attributes-recognition-barrier-0039

**Pipeline Architecture**

**Stages:**

Input Video → Decode → Object Detection → Object Classification → Watermark → Output

**DL Streamer Elements:**

Decodebin / vaapidecodebin – Decoding input video streams

Gvadetect – Object detection (OpenVINO model)

Gvaclassify – Classification using OpenVINO model

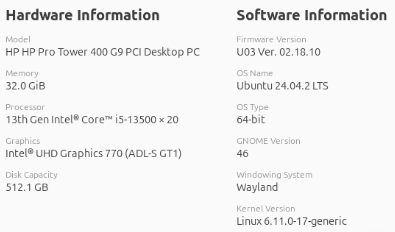
Gvawatermark – Adds labels/boxes to video

Fpsdisplaysink – Outputs video and displays FPS

**Model Information**

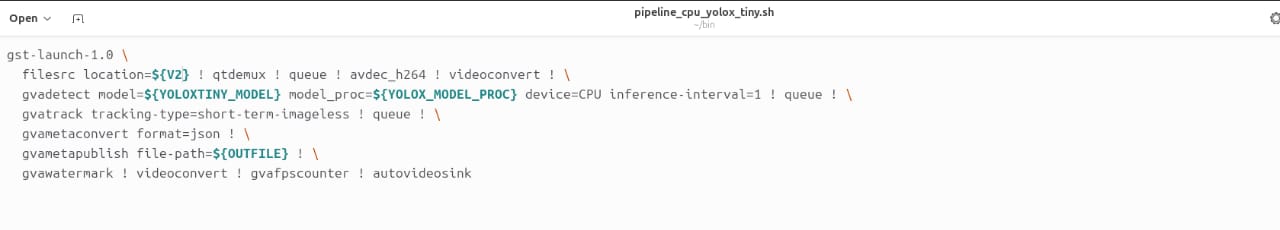
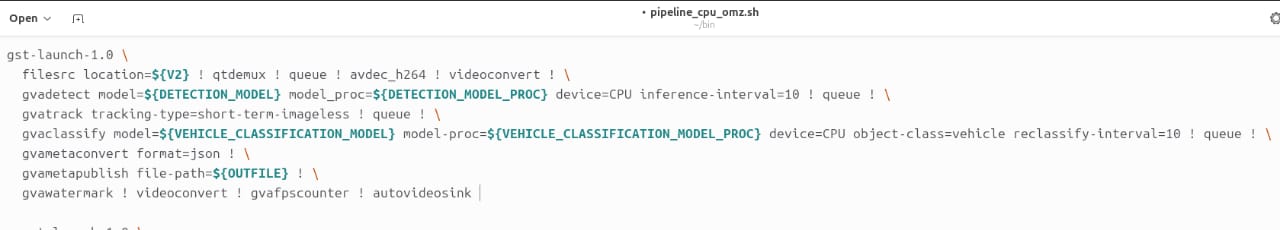
|  |  |  |  |
| --- | --- | --- | --- |
| Task | Model Name | Format | Precision |
| Detection | person-vehicle-bike-detection-crossroad-0078 | IR | FP32/FP16 |
| Classification | vehicle-attributes-recognition-barrier-0039 | IR | FP32/FP16 |

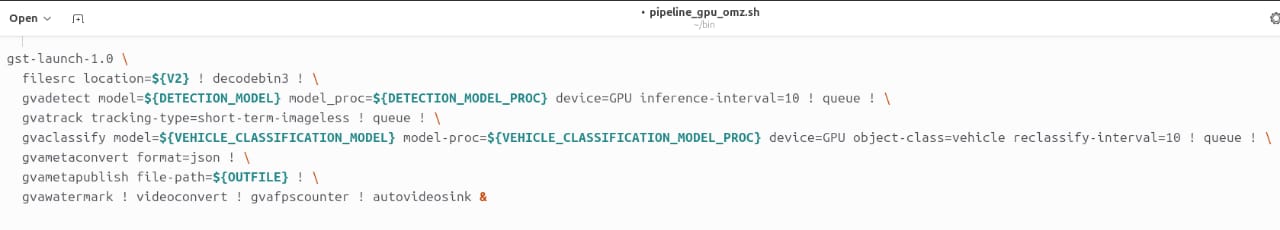
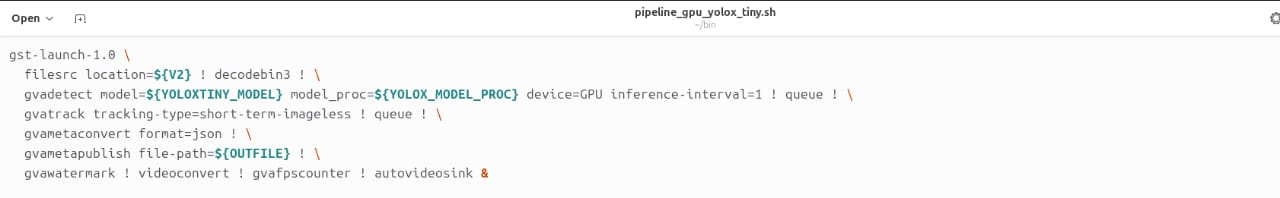
**System Scalability Analysis**



Observation: CPU Performance degrades beyond 2 streams due to compute saturation.

Observation: GPD - Stable performance up to 4 streams





**Bottleneck Summary**

|  |  |  |
| --- | --- | --- |
| **Component** | **Bottleneck** | **Details** |
| CPU | Yes | Maxes out at 2 streams due to decode + infer |
| GPU | No | (up to 4 streams) Performs well with FP16 models |
| I/O | No | Disk/network not a bottleneck |
| Decoder | Potential | Use vaapidecodebin for better e7. |

**Output & Results**

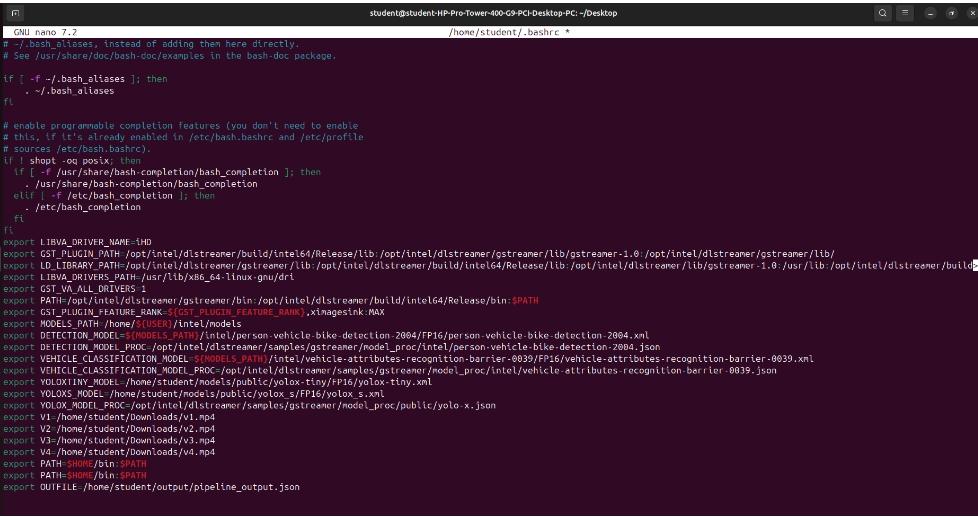
* Sample Log (for one stream):
* [Stream 1] Detected: 2 persons, 1 vehicle
* [Stream 1] Classification: Car – White, Sedan
* FPS: 28

**Visual Output:**

* Bounding boxes around objects
* Labels (e.g., “Car: White SUV”, “Person”)
* FPS counter overlay

**Optimization Techniques**

* Convert models to FP16
* Use hardware decoding (vaapidecodebin)
* Enable batch processing in detection/classification
* Reduce input resolution to support more streams
* Distribute workload across CPU + GPU



**Conclusion**

The project successfully demonstrates an end-to-end real-time video analytics system using DL Streamer and OpenVINO on Intel platforms. While CPU performance is limited to two streams, the GPU enables efficient parallel processing of four full HD streams with stable performance and low latency.

**Future Scope**

Add action recognition or face identification

Deploy to Intel NUC or edge AI kits

Integrate cloud-based dashboards

Use containerized deployment (Docker/Kubernetes) for scaling

**Acknowledgment**

We sincerely thank the Intel® Unnati Program, mentors, and faculty coordinators for their continuous support, guidance, and opportunity to learn cutting-edge AI and video analytics tools.

**References**

1. Intel DL Streamer GitHub
2. OpenVINO Toolkit Docs
3. GStreamer Plugins for DL Streamer
4. Intel Developer Zone