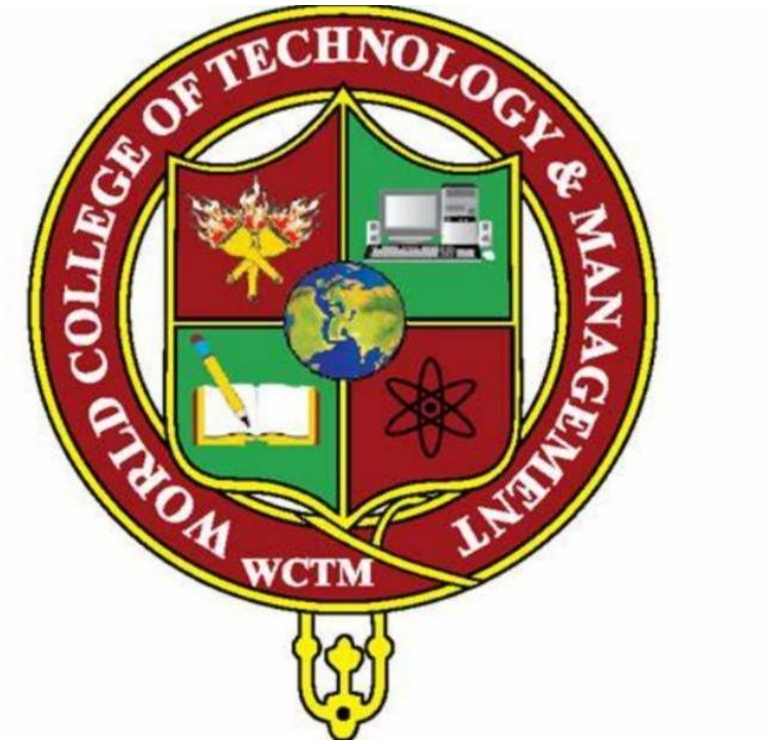
WORLD COLLEGE OF TECHNOLOGY AND MANAGEMENT, GURUGRAM



Intel Unnati Industrial Training

Summer 2025

Intel Unnati 2025 Project Report

Title: DL Streamer Pipeline: Real-time Multi-Stream Detection and Classification

Team Name: Aarogya-AI
Student Name: Piyush Mishra
College: World Collage of Technology & Management
Mentor: Ms. Sonam Tiwari (Assistant Professor)

---

1. Problem Statement Overview

Develop a pipeline using Intel DL Streamer that can:

- Decode video streams
- Perform object detection and classification
- Scale efficiently across multiple video streams
- Measure and report FPS to evaluate system scalability on Intel hardware

---

2. Tools & Technologies Used

- **DL Streamer:** Intel's Deep Learning Streamer (GStreamer plugins)
- **Docker:** For running DL Streamer container
- **OpenVINO:** For optimized inference
- **Models Used:**
  - `person-detection-retail-0013` (for detection)
  - `vehicle-attributes-recognition-barrier-0039` (for classification)
- **OS:** Ubuntu (inside Docker)
- **Input Video:** Real-world `.mp4` streams used for final testing

---

3. Pipeline Architecture

Each stream follows this pipeline:

```
Video File (.mp4)
    ↓
filesrc → decodebin → gvadetect → gvaclassify → gvawatermark → videoconvert → fpsdisplaysink
```

All streams run in parallel using separate branches in the same GStreamer pipeline.

---

4. Experimental Setup

- **Test videos:** 60s `.mp4`, real-world streams, 640x480 resolution, 30fps
- **Execution method:** `gst-launch-1.0` with `fpsdisplaysink`
- **Monitoring:** `GST_DEBUG=fpsdisplaysink:5` to extract min/max/avg FPS
- **System Load:** Monitored via `top`

Intel Unnati 2025 Project Report

5. System Specifications

Component	Specification
Laptop Model	Lenovo LOQ
CPU	Intel Core i5-13420H (12th Gen, 8 Cores)
Base Frequency	2.10 GHz
Max Turbo	4.60 GHz
Threads	12 Threads
RAM	16 GB DDR5
Storage	512 GB NVMe SSD
iGPU	Intel UHD Graphics (0x468b)
OS (WSL2)	Ubuntu 22.04 on Windows 11 (Build 26100)
WSL Kernel	6.6.87.2-microsoft-standard-WSL2
Docker	Docker Desktop with WSL2 backend

---

6. Performance Results

FPS and CPU usage per stream:

Streams	Avg FPS/Stream	Total FPS	CPU Load
1	38.0	38.0	~25%
2	37.9	75.8	~45%
4	~15.3	61.3	~85–100%

FPS Breakdown – 4 Streams:

Stream	Max FPS	Min FPS	Avg FPS
Stream 1	17.60	10.96	15.15
Stream 2	17.32	12.74	15.46
Stream 3	17.51	11.77	15.31
Stream 4	19.35	13.32	15.37

Comparison Analysis:

- From 1 to 2 streams, the system showed nearly **linear scaling** with minimal FPS loss.
- At 4 streams, **total FPS slightly dropped**, indicating CPU saturation.
- Despite the drop, real-time performance was sustained across all streams.

7. Bottleneck & Scalability

As the number of streams increases, CPU becomes the bottleneck, with near 100% usage on 4 streams.

Recommendations:
• Switch to FP16 model precision
• Leverage iGPU via device=GPU
• Optimize with multi-threading or scheduling

Intel Unnati 2025 Project Report

GPU Testing Challenges and Limitations

During the setup and testing of the DL Streamer pipeline, extensive efforts were made to utilize the integrated Intel GPU for hardware-accelerated inference. The following steps were taken:

1. Verified GPU presence and OpenCL support using `clinfo`, which correctly identified the Intel UHD Graphics device.
2. Installed required Intel OpenCL, Level Zero, and VA-API media driver packages inside the WSL2 environment.
3. Attempted to execute DL Streamer pipelines with `device=GPU`, which consistently failed due to `/dev/dri` not being accessible within WSL2.
4. Reconfigured BIOS settings (e.g., iGPU-only, Hybrid, dGPU modes) and ensured updated Intel Graphics drivers were installed on the Windows host.
5. Validated successful OpenCL setup from Windows side and tested CUDA setup independently for reference.

Despite these efforts, GPU inference through Intel DL Streamer inside WSL2 could not be executed due to WSL's current limitation of exposing GPU devices reliably for Intel iGPUs. As a result, DL Streamer GPU testing could not be completed within the WSL2 environment on this hardware setup.

Note: This limitation is specific to the Intel GPU and WSL2 interoperability in the current system configuration.

---

8. **Observations & Optimization**

- Good scaling up to 2 streams with negligible FPS drop
- Beyond 2 streams, CPU usage became a bottleneck
- For higher scalability:
    - Use FP16 models (lighter than FP32)
    - Use Intel iGPU if available (`device=GPU`)
    - Consider multithreading or stream scheduling

9. Execution Steps

- **Navigate to the project directory:**

  cd ~/dlstreamer_project
- **Activate the OpenVINO environment:**

  source openvino-env/bin/activate
- **Start the DL Streamer Docker container:**

```
docker run -it --rm \
 --device /dev/dri \
 --privileged \
 --user root \
 -e DISPLAY=$DISPLAY \
```

```
 -v /tmp/.X11-unix:/tmp/.X11-unix \
-v "$PWD":/home/dlstreamer \
intel/dlstreamer:latest
```

- **Pipeline Execution Command for 3 Streams**

```
GST_DEBUG=fpsdisplaysink:5 gst-launch-1.0 \
filesrc location=stream1.mp4 ! decodebin ! \
gvadetect model=intel/person-detection-retail-0013/FP32/person-detection-retail-0013.xml device=CPU ! \
gvaclassify model=intel/vehicle-attributes-recognition-barrier-0039/FP32/vehicle-attributes-recognition-barrier-0039.xml device=CPU ! \
gvawatermark ! videoconvert ! fpsdisplaysink name=fps1 text-overlay=false video-sink=fakesink sync=false \
filesrc location=stream2.mp4 ! decodebin ! \
gvadetect model=intel/person-detection-retail-0013/FP32/person-detection-retail-0013.xml device=CPU ! \
gvaclassify model=intel/vehicle-attributes-recognition-barrier-0039/FP32/vehicle-attributes-recognition-barrier-0039.xml device=CPU ! \
gvawatermark ! videoconvert ! fpsdisplaysink name=fps2 text-overlay=false video-sink=fakesink sync=false \
filesrc location=stream3.mp4 ! decodebin ! \
gvadetect model=intel/person-detection-retail-0013/FP32/person-detection-retail-0013.xml device=CPU ! \
gvaclassify model=intel/vehicle-attributes-recognition-barrier-0039/FP32/vehicle-attributes-recognition-barrier-0039.xml device=CPU ! \
gvawatermark ! videoconvert ! fpsdisplaysink name=fps3 text-overlay=false video-sink=fakesink sync=false \
filesrc location=stream4.mp4 ! decodebin ! \
gvadetect model=intel/person-detection-retail-0013/FP32/person-detection-retail-0013.xml device=CPU ! \
gvaclassify model=intel/vehicle-attributes-recognition-barrier-0039/FP32/vehicle-attributes-recognition-barrier-0039.xml device=CPU ! \
gvawatermark ! videoconvert ! fpsdisplaysink name=fps4 text-overlay=false video-sink=fakesink sync=false
```

---

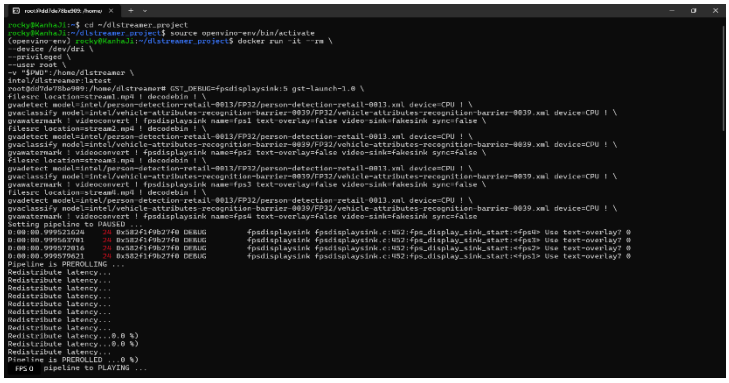
9. **Conclusion**

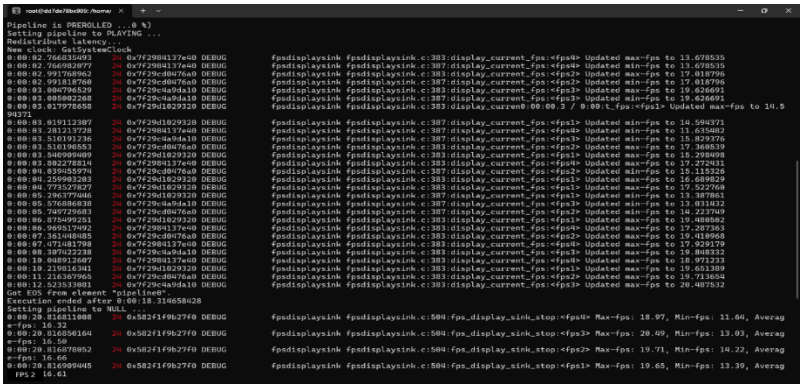
This project successfully demonstrates the capability of Intel DL Streamer to process multiple parallel real-time video streams with detection and classification. The system scaled well up to 4 streams with CPU-only configuration, making it a suitable choice for intelligent edge inferencing applications.

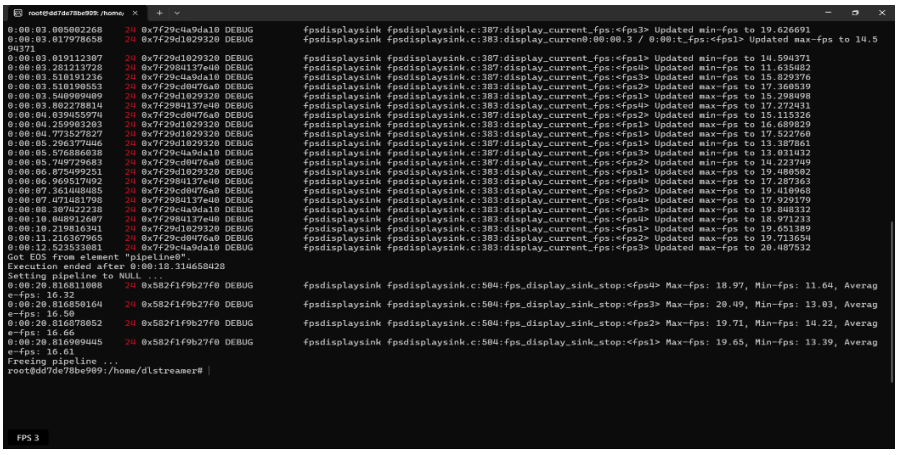
Intel Unnati 2025 Project Report

8. **Screenshots / Logs (Appendix)**

pipeline screenshots







Submitted by: Piyush Mishra
Date: 12Th July , 2025