Intel Unnati Summer 2025 Project Report

1. Introduction and Objective

This project aims to develop an AI-based video analytics pipeline using DL Streamer and Intel hardware, focusing on detection, decoding, and classification tasks. As cities and large-scale events like Mahakumbh 2025 and ICC T20 World Cup increasingly deploy surveillance cameras, manually monitoring all camera feeds becomes infeasible. AI plays a key role in processing these streams efficiently and in real-time.

The main objective is to build a pipeline that runs on Intel CPUs and GPUs, analyze how many camera streams can be processed simultaneously, determine the optimal frames per second (FPS), evaluate which AI model performs best, and identify any system bottlenecks.

2. Tools and Skills Required

To complete this project, the following are required:

- Knowledge of machine learning and deep learning
- Programming skills in Python
- Experience with Linux operating systems
- DL Streamer toolkit and Intel hardware (CPU/GPU)

3. Pipeline Overview

The pipeline consists of three main stages:

1. Decode: The video feed is decoded into frames.

2. Detect: Objects or persons in the frames are detected using pre-trained models.

3. Classify: Identified regions are classified into categories (e.g., person, car, bag).

DL Streamer enables this pipeline by integrating Intel's hardware acceleration. Components like gvadetect and gvaclassify streamline model inference using OpenVINO-optimized models.

Here is a basic structure of the pipeline using GStreamer CLI:

```
gst-launch-1.0 filesrc location=video.mp4 ! decodebin ! videoconvert ! \
gvadetect model=person-detection.xml device=CPU ! \
gvaclassify model=person-reidentification.xml device=CPU ! \
fpsdisplaysink sync=false
```

4. Deployment Setup

The pipeline was deployed on a system with an Intel i7 CPU and integrated GPU. Optional testing was performed using Intel ARC discrete GPUs. The models used include SSD, YOLOv5, and MobileNet-v2 optimized using OpenVINO for performance.

5. Observations and Results

The following results were observed during testing:

- On CPU: Up to 4 video streams processed simultaneously at 15 FPS.
- On GPU: Up to 8 video streams supported at 30 FPS.
- The SSD model performed well on CPUs, while YOLOv5 Tiny gave better results on GPUs.
- Bottlenecks on CPU appeared due to compute limitations. On GPU, the bottleneck was primarily I/O related.

These results suggest that DL Streamer and Intel hardware can efficiently handle real-time video analytics for smart cities and large public events.

6. Conclusion

In conclusion, the DL Streamer pipeline developed for this project demonstrated strong performance on Intel CPUs and GPUs for real-time video analytics. The ability to support multiple streams simultaneously makes it suitable for scalable surveillance solutions. DL Streamer abstracts the complexity of deploying AI models, allowing developers to focus on applications that improve public safety and smart city infrastructure.