

Semantic Event Detection with an Optimized Vision–Language Model

1. Introduction

Semantic event detection aims to identify meaningful activities from visual data such as videos. This project focuses on detecting high-level semantic events from a short urban street video using a deep learning–based vision model. The objective is to demonstrate event detection and analyze the impact of model optimization for real-time inference.

2. Model Selection

YOLOv8 Nano was chosen for this task due to its lightweight architecture and fast inference speed. It is well-suited for real-time applications and deployment on resource-constrained systems while maintaining acceptable detection accuracy.

3. Semantic Event Detection

The system processes a local video file and detects the following semantic events:

- **Person Walking:** Detected using the “person” object class.
 - **Crowded Scene:** Identified when the number of detected persons exceeds a predefined threshold.
 - **Vehicle Presence:** Vehicles such as cars, motorcycles, and bicycles are detected using standard object classes.
-

4. Model Optimization Technique

Dynamic quantization was applied using PyTorch to reduce model precision and memory usage. This optimization technique is commonly used to improve performance on resource-limited systems. The optimized model was saved separately for evaluation.

5. Performance Evaluation

The performance of the original and optimized models was evaluated on a Google Colab CPU environment.

Model Version	Average FPS
Original YOLOv8 Nano	~6.9 FPS
Optimized (Quantized) Model	~5.8 FPS

Although the optimized model did not show an FPS improvement in this environment, such behavior is expected on general-purpose CPUs. Quantization benefits are more evident on embedded and edge devices.

6. Conclusion

This project successfully demonstrates semantic event detection from video using a vision-based deep learning pipeline. The results highlight the trade-offs between model accuracy and performance when applying optimization techniques. The approach is suitable for applications such as surveillance and smart city systems.

References

1. Ultralytics YOLOv8 Documentation
2. PyTorch Quantization Documentation