Computer Vision / Deep Learning Engineer – Hiring Assignment

* Problem Statement

Design a real-time multi-camera intelligent vision system for an industrial or smart-environment use case.

1. USE CASE AND OBJECTIVES:

This system is designed for a factory environment where multiple cameras are used to monitor daily operations. Cameras are placed across the factory floor to observe workers, machines, forklifts, and product movement. The main goal is to improve safety by detecting dangerous situations such as workers entering restricted areas, unsafe behavior, and spills on the floor. Along with safety, the system also helps improve productivity by tracking workflow, counting products, and understanding movement patterns.

Instead of sending all video to the cloud, the system processes video on an edge device. This makes the system faster, reduces internet dependency, and allows real-time responses.

2 . System Architecture and Data Flow

The system follows an edge-based setup where most processing happens close to the cameras. Several IP cameras are installed near machines, work areas, and walkways to continuously stream live video. All camera feeds are sent to an NVIDIA Jetson edge device, which acts as the main processing unit. The Jetson runs computer vision models in real time to detect objects, identify risky zones, and track worker and machine movement.

After processing, the system generates safety alerts and collects useful information such as object counts and activity patterns. Only important data and selected video clips are sent to the cloud for storage and analysis. The cloud is mainly used for dashboards, reports, and improving the system in the future.

3. NVIDIA Platform and Tools Used

To handle multiple camera streams smoothly, the system uses NVIDIA's edge AI platform on the Jetson device.NVIDIA DeepStream is used to manage all video streams and run AI models together in a single pipeline. It also helps with tracking objects and generating real-time analytics.NVIDIA Transfer Learning Toolkit (TLT) is used to fine-tune pre-trained models using factory-specific data. This makes the models more accurate for detecting workers, machines, and safety risks while reducing training time.NVIDIA TensorRT is used to optimize the trained models so they run faster and use less memory on the Jetson device. This ensures real-time performance on edge hardware.

4. AI Modules and Processing Pipeline

The system uses multiple computer vision models working together to understand the factory environment in real time.

Object Detection identifies workers, machines, forklifts, and products in each camera frame.

Semantic Segmentation separates important areas such as floor space, restricted zones, and spill regions so unsafe areas can be clearly detected.

Tracking and Video Analytics follow objects over time to count products, monitor worker movement, and detect unusual behavior.

Image Stitching and Layout Mapping combine multiple camera views into a single factory layout view, making it easier to understand overall activity.

Clustering and Pattern Analysis study past movement data to find congestion areas, repeated unsafe actions, and inefficient workflows.

5. Edge Optimization and Deployment Strategy

Since edge devices have limited resources, the AI models are optimized to run efficiently. Quantization is used to reduce model size and increase speed. Pruning removes unnecessary parts of the model to improve performance.

TensorRT converts trained models into fast inference engines optimized for Jetson hardware. The system runs using Docker containers, making updates and maintenance easier. Model performance is monitored continuously, and updates can be applied remotely.

6. Trade-Off Analysis

Smaller and faster models are used instead of very large networks to ensure real-time performance. While this slightly reduces accuracy, it allows smooth operation on edge devices. Edge processing is preferred over cloud processing to avoid delays and ensure immediate safety alerts. Although cloud systems are powerful, edge processing is more reliable for real-time tasks.

Frame rate is balanced at around 10–15 FPS to save power while still providing enough detail for monitoring. Only important data is sent to the cloud instead of full video streams to reduce storage and network usage.

7. Failure Handling and System Reliability

If the internet connection fails, the edge device stores important data locally and syncs it when the connection is restored.

If a camera stops working, the system detects it and attempts to reconnect automatically. Alerts can be sent for maintenance if the issue continues.

If the edge device crashes, a watchdog system restarts services or reboots the device. Model accuracy is monitored over time, and retraining is triggered when performance drops.

8. Real-World Feasibility and Assumptions

The system is designed to run within the power and compute limits of NVIDIA Jetson devices. Optimized models allow real-time performance across multiple cameras. It assumes a stable power supply, working IP cameras, periodic internet connectivity, and access to labeled factory data for training. The architecture is scalable and can support additional cameras or features as needed.

Conclusion

This system provides a practical multi-camera intelligent vision solution for factory environments. By using edge-based AI processing along with optimized models and cloud analytics, it achieves real-time monitoring with high reliability. The design balances speed, accuracy, and hardware limitations, making it suitable for real-world industrial use.

Thank you,

Nagendra Guptha

9100660479.

nagendra.guptha777@gmail.com