AI-POWERED BLIND SPOT DETECTION SYSTEM FOR FOGGY AND RAINY CONDITIONS

1 Objective

Participants will develop a blind spot detection system for vehicles operating in foggy and rainy weather conditions. The challenge involves designing a virtual multi-sensor system, implementing sensor fusion techniques, and using machine learning (ML) algorithms to improve object detection and risk assessment. Since this is an online hackathon, all implementations will be simulated using software-based frameworks and datasets.

2 Problem Breakdown (Four Subparts)

2.1 Hardware Part (Simulation-Based)

2.1.1 Sensor Fusion and Virtual Data Acquisition System

Problem Statement:

Vehicles rely on multiple sensors to detect blind spots, but their effectiveness reduces in adverse weather conditions like fog and rain. Your task is to simulate multi-sensor data acquisition and fusion in such environments.

Tasks:

1. **Sensor Simulation:**

- Use tools like CARLA, Unreal Engine, ROS, or MATLAB Simulink to simulate the environment.

- Integrate Radar, LiDAR, SWIR (Short-Wave Infrared), Ultrasonic Sensors, and Camera for realistic blind spot coverage.

2. **Weather Condition Simulation:**

- Generate synthetic fog, rain, and low-visibility conditions within the simulation tool.

- Add realistic sensor noise to mimic actual weather effects.

3. **Sensor Fusion Implementation:**

- Design an algorithm to combine data from multiple sensors to improve object detection accuracy.

- Use Kalman Filters or Deep Learning-based fusion methods.

4. Dataset Generation:

- Capture and preprocess simulated sensor data for further AI-based processing.

2.1.2 Edge Processing & Embedded System Virtualization

Problem Statement:

Traditional blind spot detection systems rely on cloud processing, which introduces latency issues. The goal is to simulate an edge computing system that processes sensor data in real time within the vehicle.

Tasks:

1. Edge Computing Simulation:

- Use Docker, Virtual Machines, or cloud-based IoT platforms to simulate an embedded AI system inside a vehicle.
- Implement low-latency data processing algorithms for real-time sensor fusion.

2. Data Transmission Optimization:

- Develop an efficient communication protocol to relay sensor data to the vehicle's warning system with minimal delay.
- Optimize bandwidth usage while maintaining high detection accuracy.

3. Computational Load Balancing:

- Implement a resource-efficient model that balances accuracy vs. processing power constraints.
- Use techniques like quantization and model pruning for lightweight AI deployment.

2.2 Software Part (AI-Based Processing & Detection)

2.2.1 Machine Learning-Based Object Detection in Foggy and Rainy Conditions

Problem Statement:

Visibility is significantly reduced in foggy and rainy conditions, making traditional object detection methods unreliable. The goal is to develop an AI-based detection system that enhances vehicle safety under such conditions.

Tasks:

1. Dataset Selection & Augmentation:

- Use datasets like BDD100K, Foggy Cityscapes, or generate synthetic data.

- Implement data augmentation techniques (adding synthetic fog, blurring, contrast enhancement, etc.).

2. Object Detection Model Development:

- Train and test a Deep Learning model using YOLO, Faster R-CNN, or Transformer-based models.

- Fine-tune the model for low-visibility scenarios by improving contrast, edge detection, and depth estimation.

3. Preprocessing & Enhancement:

- Implement Dehazing, Rain Removal, and Noise Reduction using Deep Learning-based Image Processing techniques.

- Compare performance using traditional filters (CLAHE, Bilateral Filtering) vs. AI-based methods.

4. Performance Evaluation:

- Compare the detection accuracy in normal vs. foggy conditions using metrics like mAP (Mean Average Precision) and IoU (Intersection over Union).

- Optimize the model for real-time performance in embedded/edge computing environments.

2.2.2 AI-Powered Risk Assessment and Adaptive Decision-Making System

Problem Statement:

Detecting objects alone is not sufficient—a system must assess collision risk and trigger warnings or adaptive vehicle responses in real-time. The task is to design an AI-based risk assessment model that classifies potential threats based on sensor data.

Tasks:

1**. Risk Assessment Algorithm Development:**

- Train an AI model to predict potential collisions using parameters like object distance, velocity, movement trajectory, and vehicle speed.

- Implement techniques like Reinforcement Learning, Bayesian Networks, or Decision Trees for real-time risk prediction.

2. **Adaptive Warning Mechanism:**

- Develop a smart alert system based on real-time object proximity and motion analysis.
- Simulate alerts like audio-visual signals, dashboard notifications, or virtual haptic feedback.

3. **Scenario-Based Simulation:**

- Test the system with different driving scenarios (e.g., merging lanes, sharp turns, and highway driving in fog/rain).
- Implement real-time decision-making for vehicle assistance (e.g., automatic braking, lane departure warnings).

3 Deliverables

Participants should submit:

☑ Simulation Demonstration: A video or live demo showcasing their simulated system in action.

☑ Technical Report: A detailed document explaining their approach, methodology, algorithms used, and performance evaluation.

☑ Code Repository: A GitHub or cloud-based repository containing their code, dataset processing scripts, and model files.

Expected Outcome

By the end of the hackathon, participants should have developed a fully functional AI-based blind spot detection system capable of detecting objects and assessing risks in low-visibility weather conditions using simulated multi-sensor data and machine learning models.