

Real-time multi-class object detection for

ADAS use cases.

Project description

The project aims to develop a real-time multi-class object detection system for ADAS. The system shall utilize Image processing, and deep learning models to detect and classify various objects (e.g. Car, Pedestrian, Cyclist) in a live feed. The objective is to enhance automotive systems' situational awareness and safety features by providing accurate and timely object detection and classification.



Project requirements:

1. Technical stack:

- Image processing & OpenCV: Implement pre-processing algorithms to handle different lighting conditions, weather scenarios, and camera angles
- Deep learning: Utilize a DL framework (e.g. Tensorflow, PyTorch) to train and deploy a multi-class object detection model.

2. Functional requirements:

- Real-time processing: Ensure the system can process video frames in real time with minimal latency
- Multi-class detection: The system should accurately detect and classify at least five different classes of objects relevant to automotive safety (E.g. Cars, Trucks, Pedestrians, and bicycles)
- Accuracy & Robustness: The model should achieve an accuracy of at least 90% precision & recall under various environmental conditions.

3. Non-functional requirements:

- Usability: The frontend interface should be intuitive and provide clear visual indicators of detected objects.
- Performance: The solution should be optimized for performance to ensure smooth and real-time detection without significant lag.

Project expectations:

1. Planning and Design:

- Conduct a thorough requirements analysis and create a detailed project plan.

- Design the system architecture, including data flow, component interactions, and integration points.

2. Development and Implementation:

- Develop and integrate the frontend application using a suitable framework (e.g., React, Angular).
- Implement image preprocessing techniques using OpenCV to enhance the quality of input data.
- Train and fine-tune a deep learning model using a suitable dataset (e.g., IDD (Indian Driving Dataset)).
- Implement real-time object detection and classification logic.

3. Testing and Validation:

- Validate the model's performance using diverse test scenarios to ensure accuracy and reliability.
- Optimize the system for real-time performance and minimal latency.

4. Deployment and Documentation:

- Deploy the solution on a suitable platform, ensuring compatibility with target hardware.
- Provide comprehensive documentation, including user guides, technical specifications, and maintenance instructions.

5. Presentation and Review:

- Prepare a detailed presentation showcasing the project's objectives, implementation, results, and future improvements.