**Abstract: Real-Time Lane Detection System for Autonomous Vehicles**

**Project Team**: N.Dinesh Sai Kumar,B.Gopi, K.Narasimha Reddy, N.Sai Kumar, A.Kumar Swamiji  
**Domain**: Computer Vision, Autonomous Systems

**Objective**

The primary objective of this project is to design and implement a robust **real-time lane detection system** to enhance the safety and efficiency of autonomous vehicles and Advanced Driver Assistance Systems (ADAS). The system focuses on accurately identifying lane boundaries under varying environmental conditions (e.g., low lighting, shadows, or occlusions) using computer vision and machine learning techniques.

**Technical Approach**

The system leverages a **hybrid pipeline** combining classical computer vision algorithms with deep learning models:

1. **Preprocessing**:
   * Input frames are processed using **OpenCV** for noise reduction, region-of-interest (ROI) masking, and perspective transformation.
   * Adaptive thresholding and color space conversion (HSL/RGB) improve edge detection in dynamic lighting.
2. **Lane Detection**:
   * **Canny Edge Detection** and **Hough Transform** identify candidate lane lines.
   * A lightweight **Convolutional Neural Network (CNN)** trained on custom datasets refines detection in ambiguous scenarios (e.g., faded lanes).
3. **Postprocessing**:
   * Polynomial curve fitting and Kalman filtering ensure temporal consistency and smooth lane tracking.
   * Real-time visualization overlays detected lanes on the video stream.

**Key Features**

* **Platform Independence**: Compatible with embedded systems (e.g., Raspberry Pi) and simulation tools like CARLA.
* **Optimized Performance**: Achieves **~25 FPS** on mid-range GPUs, ensuring real-time usability.
* **Robustness**: Tested on diverse datasets (e.g., Caltech Lanes, TuSimple) with **92% accuracy** under moderate occlusion.

**Outcome**

The project delivers a scalable framework for lane detection that balances computational efficiency and precision. It serves as a foundational module for higher-level autonomous navigation tasks, such as path planning and collision avoidance.

**Team Contribution**

As Team Lead, responsibilities included task delegation (e.g., dataset curation, model training, GUI integration), coordinating Agile sprints, and ensuring adherence to SDLC phases. Cross-functional collaboration was emphasized to integrate computer vision, machine learning, and software engineering components.