DOCUMENTATION

ROAD LANE DETECTION

**Scope and Objective**

This project has the following scopes-

* *Image and Video Processing*: The project will focus on processing both images and video feeds to detect lane lines, catering to applications requiring real-time analysis of road scenes.
* *2D Lane Detection*: The scope will encompass detecting and highlighting lane lines in two-dimensional space, primarily focusing on lane markings visible from the vehicle's perspective.
* *Single and Multi-Lane Detection*: The system will be designed to detect both single and multiple lanes on the road, accommodating scenarios such as highways, urban roads, and intersections.
* *Variability in Road Conditions*: The project will address variations in road conditions, including different lighting conditions (daytime, nighttime, shadows), road types (highways, urban streets), and weather conditions (rain, snow).
* *Platform Independence*: The lane detection system will be platform-independent, allowing deployment on different hardware configurations and operating systems commonly used in autonomous vehicles and driver assistance systems.
* *Integration Capabilities*: While the primary focus is on lane detection, the project will consider integration capabilities with other components of autonomous vehicle systems, such as obstacle detection, navigation, and control systems.
* *Performance Metrics*: Evaluation of the system will include metrics such as accuracy, precision, recall, and computational efficiency, ensuring that the system meets performance requirements for real-world deployment.

Objectives of the project are-

* *Develop a Robust Lane Detection System*: The primary objective is to design and implement a lane detection algorithm capable of accurately identifying and highlighting lane lines in various road scenarios, including different lighting conditions, road types, and weather conditions.
* *Real-Time Performance*: Ensuring that the lane detection system operates in real-time, allowing for timely responses in autonomous vehicle navigation or driver assistance systems.
* *High Accuracy and Reliability*: Aiming for high accuracy and reliability in lane detection to enhance the safety and efficiency of autonomous vehicles or driver assistance systems.
* *Scalability and Adaptability*: Designing the system to be scalable and adaptable to different environments and road conditions, accommodating future advancements and variations in road infrastructure.
* *User-Friendly Interface*: Providing a user-friendly interface for interacting with the lane detection system, allowing users to visualize lane markings and adjust detection parameters as needed.
* *Documentation and Deployment*: Ensuring comprehensive documentation of the project, including code documentation, model architecture, training procedures, and deployment instructions. Facilitate seamless deployment of the lane detection system in various applications and environments.

**Data Collection**

Gather a diverse dataset of images or videos containing road scenes with different lighting conditions, weather, and road types. Include annotations for lane markings.

First, I am trying to build this detector without using machine learning algorithms and just by using OpenCV.

But, later for improvisations I will take this [kaggle dataset](https://www.kaggle.com/datasets/kpgeek/kitti-roadlane-detection-dataset-224-x-224).

**Preprocessing**

Preprocess the data by resizing images, normalizing pixel values, and augmenting the dataset to improve model generalization.

**Model Selection**

Choose an appropriate computer vision model for lane detection. Common choices include Convolutional Neural Networks (CNNs) like U-Net, semantic segmentation models, or traditional image processing techniques like Hough Transform.

**Training**

Split the dataset into training, validation, and test sets. Train the chosen model using the training data and validate its performance using the validation set. Fine-tune hyperparameters as necessary.

**Evaluation**

Evaluate the model's performance on the test set using metrics such as accuracy, precision, recall, and F1-score. Iterate on the model architecture and training process if necessary to improve performance.

**Deployment**

Deploy the trained model as an application or service that can process images or video feeds in real-time. Ensure scalability and efficiency for deployment on different platforms.

**END OF DOCUMENTATION**