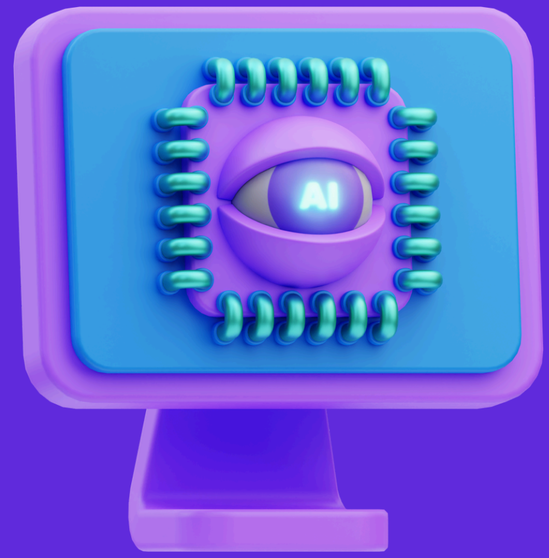
Lane Lines Detection using Python and OpenCV

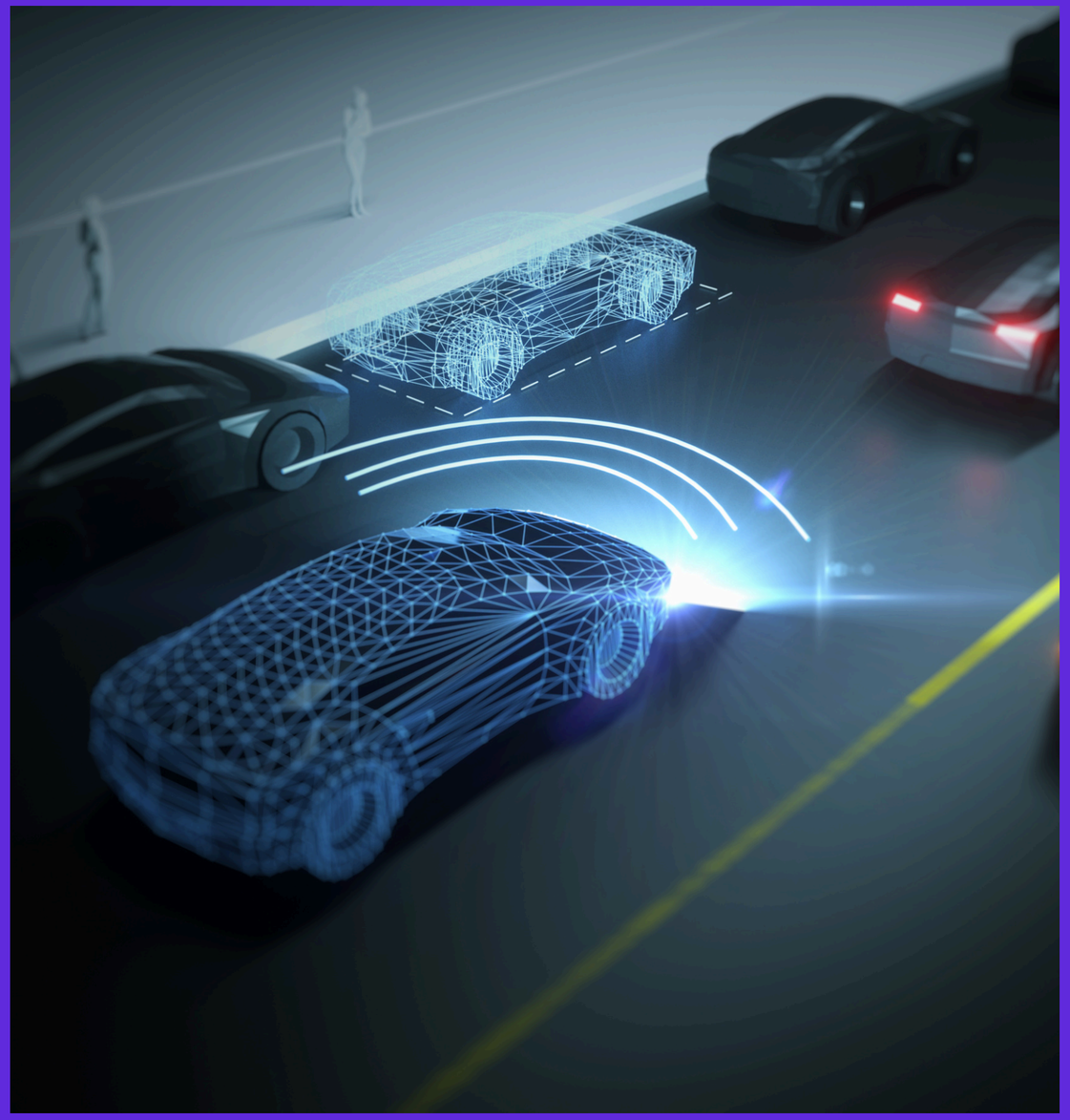
Leveraging Computer Vision for identifying lane markings on the road.

Name: Sanskar Singh
Batch: 24
Group: 3



Introduction

This project outlines and implements a computer vision solution to detect lane lines on roads. Lane line detection is a critical component in Advanced Driver Assistance Systems (ADAS) and autonomous vehicle technology, contributing to safer and more efficient navigation by helping vehicles maintain their position within a lane. This system will process video streams or images to identify and highlight lane markings.



Automation & Vision-Based Lane Detection

This project focuses on automating the process of identifying lane markings from a visual input.

Vision-based lane detection utilizes computer vision techniques to analyze images or video frames and identify lane lines.

Methodology

1. **Image Acquisition:** Begin with loading diverse road images or individual frames from a video stream.
2. **Preprocessing** - Color Selection:
3. **Preprocessing** - Edge Detection Preparation
4. **Edge Detection** - Canny Algorithm
5. **Region of Interest** (ROI) Masking
6. **Line Detection** - Hough Transform
7. **Line Post-processing** - Averaging & Extrapolation:
8. **Visualization & Output**
9. **Video Processing Application**

System Architecture Overview

The system architecture for lane line detection typically involves the following stages:

- **Input Stage:** Acquisition of video frames from a camera (either live or pre-recorded) or static road images. This serves as the primary data source.
- **Processing Pipeline:** A series of image processing steps are applied to each frame/image to identify lane lines. This includes preprocessing, edge detection, region of interest selection, line segment detection, and lane line fitting/extrapolation.
- **Output Stage:** The detected lane lines are typically overlaid onto the original video frames or images, providing a visual representation of the detected lanes. This output can also be used to calculate parameters like lane curvature or vehicle offset from the lane center.

Tools and Technologies

- Programming Language: Python
  - Chosen for its extensive libraries for image processing, scientific computing, and ease of prototyping.
- Primary Library: OpenCV (Open Source Computer Vision Library)
  - Utilized for fundamental image processing operations such as:
    - Image/video loading and manipulation
    - Color space conversions
    - Gaussian blurring
    - Canny edge detection
    - Image masking and bitwise operations
    - Hough Line Transform
    - Drawing functions
- Supporting Libraries:
  - NumPy Matplotlib,

Conclusions

This project successfully demonstrated a Python and OpenCV-based lane detection system. A robust pipeline was established for processing both images and real-time video, effectively utilizing HSL color segmentation, Canny edge detection, ROI focusing, and the Hough Transform for straight lane identification, thereby meeting the objective of creating a foundational computer vision application. Future enhancements will focus on advanced capabilities such as curved lane detection and adaptive parameter tuning. Significant efforts will also be directed towards improving robustness against environmental challenges like shadows, poor markings, and adverse weather. Further development will explore advanced tracking with temporal filtering, integration with other sensors like LiDAR or RADAR for a more comprehensive ADAS, and performance optimization for faster real-time processing.