Lane Detection for Autonomous Vehicles Using Computer Vision

Prepared by:
Chellouche Oumaima Chaima
Student at the National Higher School of Autonomous Systems Technology
Algiers, Algeria
Email: mimachellouche@gmail.com

Company:
CodeAlpha
Program: Robotic and Automation
Location: Lucknow, India
Official Website: codealpha.tech

Submission Date:
2025

Abstract

Autonomous vehicles rely on AI to perceive their environment and make safe driving decisions. Cameras, LiDAR, and radar sensors provide essential data for detecting lanes, obstacles, and traffic signs. By combining these sensors, vehicles gain a reliable understanding of their surroundings. This project focuses on **camera-based lane detection**, demonstrating how classical computer vision techniques can identify lane markings in images and video, forming a foundation for autonomous driving applications.

---

1. Introduction

Autonomous vehicles are at the forefront of artificial intelligence and intelligent transportation systems. They rely heavily on perception systems to understand their environment and make safe driving decisions. Among the most critical perception tasks is **lane detection**, which allows a vehicle to identify road lanes and remain within its designated path. Lane detection is a key component of **Advanced Driver Assistance Systems (ADAS)** and fully autonomous driving systems. This project focuses on the **implementation and simulation of a lane detection system** using classical computer vision techniques. The system is implemented in **Python using OpenCV**, and it is tested on both images and video streams to simulate real-world driving conditions.

---

2. Artificial Intelligence in Autonomous Vehicles

2.1 Perception Systems

Autonomous vehicles rely on AI to understand their environment. **Cameras, LiDAR, and radar sensors** provide data that the vehicle uses to detect lanes, obstacles, and traffic signs.

- **Cameras** capture lane markings, vehicles, and signs; they are high-resolution but sensitive to lighting.
- **LiDAR** creates 3D maps and measures distances accurately, useful for detecting obstacles.
- **Radar** detects moving objects and works well in poor weather conditions.

By combining these sensors, known as **sensor fusion**, the vehicle gains a reliable understanding of its surroundings, enabling safe navigation and decision-making. In this project, lane detection is achieved using **camera-based perception**, forming the foundation for autonomous driving tasks.

2.2 Lane Detection Techniques

Lane detection in autonomous vehicles can be approached using **two main categories** of methods: classical computer vision and deep learning–based approaches. Each method has distinct characteristics, advantages, and limitations.

2.2.1 Classical Computer Vision Methods

Classical computer vision techniques rely on **manually designed algorithms** to extract lane information from images. Typical steps include:

1. **Edge Detection** – Algorithms such as the **Canny edge detector** identify abrupt changes in pixel intensity, highlighting lane markings that often appear as bright lines against the road surface.
2. **Color Filtering** – Some methods isolate lanes based on their color (e.g., white or yellow markings) to distinguish them from the surrounding pavement.
3. **Geometric Transformations** – Techniques such as perspective transformation (bird's-eye view) and region-of-interest selection help focus on relevant parts of the image, reducing noise from irrelevant areas like sidewalks, cars, or vegetation.
4. **Line Detection** – The **Hough Transform** is commonly used to detect straight lines, representing the left and right lanes on the road.

Advantages:

- Computationally efficient and suitable for real-time applications
- Fully interpretable and easy to debug
- Works well under normal lighting and clearly marked lanes

Limitations:

- Less robust under challenging conditions: shadows, curved lanes, occlusions, faded markings
- Requires manual tuning of parameters such as thresholds for edge detection and line detection

2.2.2 Deep Learning–Based Methods

Deep learning methods, particularly **Convolutional Neural Networks (CNNs)**, learn to detect lanes directly from large datasets of labeled images. Unlike classical methods, these algorithms **do not require manual feature design**. Key characteristics include:

1. **Feature Learning** – CNNs automatically extract relevant features for lane detection, capturing subtle patterns in the image such as lane curvature, intersections, and dashed lines.
2. **Robustness** – Deep learning models can handle challenging scenarios, including poor lighting, complex road layouts, and partial occlusions.
3. **End-to-End Prediction** – Some models can take an image as input and directly output lane coordinates or segmentation maps, simplifying the pipeline.

Advantages:

- Highly accurate and adaptive to complex road conditions
- Can generalize to various environments without manual tuning

Limitations:

- Requires large labeled datasets for training
- Computationally intensive; may need GPU acceleration for real-time performance
- Less interpretable than classical methods

3. System Architecture and Simulation Environment

3.1 System Overview

The lane detection system processes images or video frames from a front-facing camera. Each frame undergoes preprocessing, edge detection, and lane extraction, simulating a vehicle's perception pipeline.

3.2 Simulation Tools

- **Programming Language:** Python
- **Libraries:** OpenCV, NumPy
- **Input Data:** Static images and dashcam videos

3.3 Sensors

- **Camera:** Provides visual input for lane detection
- **LiDAR (optional):** Can enhance lane detection in autonomous vehicles but was not used in this project

4. Lane Detection Methodology

The lane detection methodology is structured as a **step-by-step pipeline**, designed to process visual input from a front-facing camera and detect lane markings reliably. Each step is crucial for transforming raw image data into usable lane information for autonomous driving.

4.1 Image Acquisition

The first step involves capturing visual data from either static images or video frames. Static images are used to test lane detection on specific scenes, while video frames allow simulation of real-time detection. Reliable image acquisition ensures that the subsequent processing stages receive high-quality input for analysis.

4.2 Image Preprocessing

Once images are acquired, they are converted to **grayscale**, which reduces the image to a single intensity channel. This simplification decreases computational requirements while retaining essential features, such as lane edges.After grayscale conversion, a **Gaussian blur** is applied to smooth the image and reduce noise. Noise in an image, such as small variations in brightness or texture, can generate false edges that negatively impact lane detection. The Gaussian filter helps eliminate these spurious details while preserving the overall shape of lane markings.

4.3 Edge Detection

The smoothed grayscale image is then processed using an **edge detection algorithm**. Lane markings typically have high contrast with the surrounding road surface. Edge detection highlights these changes in intensity, producing a binary image where lane lines appear as bright edges. This step is essential for isolating the structures in the image that represent potential lanes.

4.4 Region of Interest Selection

Not all edges in an image correspond to lane markings. To focus on the relevant area, a **region of interest (ROI)** is defined, typically as a trapezoidal shape covering the portion of the road ahead of the vehicle. Applying this mask removes irrelevant information from the scene, such as buildings, trees, or other vehicles outside the driving lane, thus reducing false positives and improving detection accuracy.

4.5 Lane Extraction

After isolating edges in the ROI, the system uses a **line detection algorithm** to extract lane markings. Specifically, it detects straight lines representing left and right lanes on the road. This step converts the pixel-based edge information into structured lane representations that can be further used for navigation.

Lane extraction identifies the slope and position of each lane, providing a visual overlay for monitoring purposes. In more advanced systems, this information can directly inform steering controls or autonomous navigation.

4.6 Lane Visualization

The final step overlays the detected lanes on the original images or video frames. Visualization allows for validation and assessment of the detection system’s performance. By continuously displaying the lanes, it simulates how an autonomous vehicle perceives the road in real-time conditions.

5. Experimental Results and Analysis

The system was tested on multiple images and video frames. Key observations:

- **Static Images:** Left and right lane markings were accurately detected under normal lighting.
- **Video Streams:** Lane lines were detected frame by frame, providing a continuous lane overlay on the video.

Limitations

- Curved or faded lanes can sometimes be missed
- Shadows and varying lighting can reduce accuracy

These limitations highlight areas for improvement, such as **using CNN-based lane detection** or combining camera data with LiDAR for enhanced perception.

6. Applications and Future Work

This lane detection system can be applied to:

- **ADAS:** Lane keeping assistance and collision prevention
- **Autonomous Vehicles:** Core component of perception systems
- **Smart Transportation:** IoT-enabled traffic management systems

Future improvements may include:

- Deep learning-based lane detection for more robustness
- Integration with obstacle detection for full driving assistance
- Real-time deployment on embedded platforms such as **Raspberry Pi** or **NVIDIA Jetson**

7. Conclusion

This project demonstrates that **classical computer vision techniques** can effectively detect road lanes from images and videos. Lane detection is a crucial task for autonomous driving and safety systems, providing the vehicle with essential information to maintain proper lane positioning and avoid collisions. It also forms a foundation for more complex AI-based perception pipelines, enabling future integration with obstacle detection, traffic sign recognition, and autonomous navigation.

While classical methods perform well under ideal conditions, they have limitations in challenging scenarios such as curved roads, poor lighting, or partially occluded lane markings. **Future work** can incorporate deep learning models, additional sensors, and sensor fusion to improve robustness and accuracy, moving closer to fully autonomous driving in diverse environments.

Overall, this project highlights the **practical application of AI in vehicle perception**, demonstrating a clear and interpretable approach to lane detection that can serve as a stepping stone for more advanced autonomous driving systems.

The complete implementation, including source code and experimental materials, is available in the associated GitHub repository for reproducibility.

. References

[1] John Canny,*A Computational Approach to Edge Detection*,IEEE Transactions on Pattern Analysis and Machine Intelligence, June 1986.
https://doi.org/10.1109/TPAMI.1986.4767851

[2] Richard O. Duda, Peter E. Hart,*Use of the Hough Transformation to Detect Lines and Curves in Pictures*,Communications of the ACM, January 1972.
https://doi.org/10.1145/361237.361242

[3] Mohamed Aly,
Real-Time Detection of Lane Markers in Urban Streets,
IEEE Intelligent Vehicles Symposium, June 2008.
https://doi.org/10.1109/IVS.2008.4621159

[4] M. Bertozzi, A. Broggi, *Gold: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection*, IEEE Transactions on Image Processing, August 1998.
https://doi.org/10.1109/83.704312

[5] Rafael C. Gonzalez, Richard E. Woods,*Digital Image Processing*,Pearson Education, 4th Edition, 2018.
https://www.pearson.com

[6] Andreas Geiger, Philip Lenz, Raquel Urtasun, *Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite*,
IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2012.
https://doi.org/10.1109/CVPR.2012.6248074

[8] Ian Goodfellow, Yoshua Bengio, Aaron Courville,
Deep Learning,
MIT Press, 2016.
https://www.deeplearningbook.org