International Journal of Research Publication and Reviews, Vol (6), Issue (4), April (2025), Page – 4284-4291

WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

# Deep Learning-Driven Lane Detection and Collision Avoidance System for Self-Driving Cars Using Stereo Vision

*Maleha Sarwath Ansari[1], Shivani Nandapurkar[2], Samridhi Jhijaria[3], N S Pranav[4], Medha Sharma[5], Shaik Ashfaaq Hamja[6], Rohit Ashok Dhamdhere[7], Hitesh Jambhale[8]*

[1]Muffakham Jah College of Engineering and Technology
[2]Bhilai Institute of Technology
[3]VIT Vellore
[4]Global Academy of Technology
[5]University of Petroleum and Energy Studies
[6]VIT, Chennai
[7]Pimpri Chinchwad College of Engineering and Research
[8]Vishwaniketan's Institute of Management Entrepreneurship and Engineering Technology

Abstract

This research presents a novel and lightweight stereo vision-based framework for driving lane detection and classification, aimed at enhancing the functionality of Advanced Driver Assistance Systems (ADAS). The proposed system enables precise lateral localization of the ego-vehicle and provides reliable forward collision warnings. To achieve robust lane detection, we introduce a self-adaptive traffic lane modeling approach in Hough Space that utilizes maximum likelihood angle estimation and dynamic region-of-interest (ROI) adaptation for pole detection. This method effectively handles challenges such as road surface irregularities, structural variations in lanes during vehicle motion, and interference from extraneous road markings. Furthermore, the integration of Geographic Information Systems (GIS) or electronic maps can be employed to further refine detection accuracy.

In parallel, 3D data derived from stereo matching is leveraged to construct an obstacle mask, allowing for the suppression of irrelevant objects and accurate estimation of forward collision distance. For lane classification, a Convolutional Neural Network (CNN) is trained on manually annotated ROIs from the KITTI dataset, enabling the reliable identification of left and right lane boundaries relative to the host vehicle. This classification supports intelligent lane-change decision-making strategies within ADAS.

Experimental results demonstrate a high true positive rate in both detection and classification tasks, while maintaining real-time processing capability at 15Hz. The system also exhibits strong robustness across diverse environmental conditions, validating its effectiveness for real-world deployment.

## 1. Introduction

In recent years, Autonomous Driving and Advanced Driver Assistance Systems (ADAS) have become critical in reducing traffic accidents and enhancing road safety. Among the foundational technologies for intelligent vehicles and Intelligent Transportation Systems (ITS), lane detection and classification play a pivotal role. These systems not only contribute to accurate lateral positioning of the vehicle but also provide valuable traffic context, enabling the ego-vehicle to make informed driving decisions.

As a result, this field has garnered significant attention from research institutions and automotive technology companies alike. For instance, companies such as Mobileye and Bosch are actively developing cost-effective yet robust solutions that integrate Forward Collision Warning (FCW) and Lane Departure Warning (LDW) functionalities by combining camera systems with millimeter-wave radar, aimed at structured driving environments.

Leading automobile manufacturers—including Mercedes-Benz, BMW, and Tesla—are also integrating similar capabilities into their vehicles through collaborations with renowned research groups, such as the KIT Autonomous Vision Group and Daimler Research Group.

Given the advantages of stereo cameras in autonomous driving—particularly their ability to capture dense spatial information of the surrounding environment for safer navigation—we propose a stereo vision-based lane detection and classification approach. Our system is designed for FCW applications in structured environments, offering accurate drivable area estimation and forward collision distance detection to support safe and intelligent vehicle operation.

### *1.1 Existing System*

Current self-driving vehicles rely on an array of advanced sensors to perceive their environment. These include thermographic cameras, radar, LiDAR, sonar, GPS, odometry, and inertial measurement units (IMUs). The data gathered from these sensors is processed by sophisticated control systems that interpret the surrounding context, enabling the vehicle to detect road boundaries, obstacles, and traffic signs to determine safe navigation paths.

Many autonomous vehicle systems heavily utilize image recognition in conjunction with machine learning and deep neural networks to automate decision-making processes. These systems are trained on large datasets to accurately interpret complex driving scenarios and perform real-time adjustments.

As vehicle speed increases, the demand for high-frequency data input also rises, which in turn improves the granularity of decisions made by the deep learning algorithms. Additionally, vehicle software often integrates with mapping services, such as Google Maps, to receive prior information about environmental features like traffic signals, street signs, and landmarks. This integration enhances the vehicle's situational awareness and contributes to more accurate and proactive driving behavior. Despite these advancements, the current systems still face challenges related to real-time responsiveness, obstacle classification, and cost-effective implementation for widespread deployment.

### *1.2 Proposed System*

The proposed system introduces an efficient lane detection and curvature estimation pipeline, designed for deployment in real-time on embedded platforms such as Raspberry Pi-based self-driving car prototypes. The core objective is to detect lane boundaries, estimate their curvature, and provide meaningful feedback for autonomous navigation and lane-following tasks.

The system begins by capturing real-time road images using a front-facing stereo or monocular camera. From these input images, the relevant lane features are extracted through a series of pre-processing steps including grayscale conversion, Gaussian blurring, and edge detection (e.g., Canny edge detection). These steps help to isolate high-frequency features and eliminate noise, enhancing the visibility of lane markings.

To analyze lane geometry more effectively, we apply a perspective transformation—commonly referred to as "warping"—to generate a bird's eye view of the road. This top-down view simplifies the problem of detecting curves and lateral deviations, as lane lines appear straighter and more evenly spaced. It allows the algorithm to evaluate curvature in a more geometrically consistent manner, which is essential for accurate control decisions.

In the warped image, we apply pixel summation and histogram analysis techniques to identify the base points of the lane lines. A sliding window approach is then used to trace the full path of each lane boundary across the frame. By fitting second-degree polynomials to the detected lane lines, we compute the radius of curvature and the vehicle's position relative to the center of the lane. These parameters are vital for determining steering adjustments and issuing lane-departure or correction commands.

To improve robustness and maintain continuity in challenging scenarios—such as partial lane occlusion or shadow interference—the system implements temporal smoothing and confidence scoring mechanisms across frames. These enhancements ensure the curvature values remain stable, reducing jitter and maintaining consistent lateral control.

The final results, including the detected lane lines and curvature overlay, are rendered back onto the original image to provide a visual reference for debugging or human supervision. Furthermore, this pipeline is optimized for deployment on resource-constrained platforms like the Raspberry Pi, enabling low-cost real-time autonomous driving experiments.

An essential aspect of this implementation is the tuning of various hyperparameters such as edge detection thresholds, perspective warp coordinates, and polynomial fit tolerances. Fine-tuning these parameters is crucial for adapting the system to different driving conditions, camera positions, and road environments.

### *1.3 Motivation*

The primary motivation behind this research stems from the growing potential of autonomous vehicles to significantly enhance road safety and driving efficiency. Unlike human drivers, autonomous systems are entirely analytical in nature, relying on precise data from cameras, radar, LiDAR, GPS, and other sensors to navigate complex environments. These systems are immune to distractions such as mobile phones, fatigue, or impairments caused by alcohol or emotional stress—factors that contribute to a majority of road accidents today.

The computational units powering autonomous vehicles are capable of processing sensor data and making decisions at speeds far exceeding human reaction time. They consistently follow predefined safety rules, maintain optimal driving behavior, and are not influenced by subjective judgment or situational misinterpretation.

With the rapid development of artificial intelligence, machine learning, and sensor fusion technologies, the vision of a future populated by intelligent self-driving vehicles is becoming increasingly feasible. Such a future promises not only to reduce traffic-related fatalities and injuries but also to alleviate congestion, minimize human error, and promote sustainable transportation systems.

This research focuses on contributing to that vision by developing an accurate, lightweight, and real-time lane detection and curvature estimation system. By improving the reliability of lane understanding and forward collision detection in autonomous vehicles, we aim to take a meaningful step toward safer, smarter, and more responsive transportation.

### *1.4 Objectives*

The primary goal of this research is to design and implement a robust, real-time stereo vision-based lane detection and classification system for Forward Collision Warning (FCW), contributing to safer and more intelligent autonomous driving. The specific objectives of this study are:

- To develop a real-time and reliable lane detection and classification framework capable of supporting forward collision warning systems, particularly in structured driving environments.
- To integrate detection and classification outputs into a Convolutional Neural Network (CNN), enabling the system to interpret and act upon visual lane information, which is essential for assisted driving functionalities.
- To enhance the situational awareness of the ego-vehicle by providing detailed traffic and lane information, thereby supporting more informed and safer driving decisions.
- To address major causes of vehicular collisions—such as driver error, distraction, and drowsiness—by creating a system that operates purely on sensor input and real-time analysis, reducing dependence on human judgment.
- To contribute to a measurable reduction in traffic fatalities and to improve lane capacity utilization, ultimately resulting in shorter travel times and more efficient road usage.
- To support environmental sustainability by improving fuel economy through optimized driving decisions and reduced stop-and-go behavior.

## 2. Literature Survey

The field of autonomous driving has witnessed significant advancements, particularly in lane detection and classification systems, which are crucial for vehicle navigation and safety. Recent studies have introduced innovative methodologies to enhance the accuracy and robustness of these systems. Pandian et al. (2025) proposed the Multi-Armed Bandit Ensemble (MAB-Ensemble), an ensemble learning technique designed to improve lane detection in autonomous vehicles. This approach dynamically selects the most suitable Convolutional Neural Network (CNN) model based on prevailing environmental factors, demonstrating enhanced performance across various road conditions .

In the realm of practical applications, DeepRoute.ai announced a collaboration with Qualcomm in April 2025 to develop cost-effective Advanced Driver Assistance Systems (ADAS). These systems aim to support features such as urban autopilot navigation, highway driving assistance, and automated parking, leveraging Qualcomm's Snapdragon platforms .

Furthermore, Southwest Research Institute (SwRI) developed off-road autonomous driving tools in March 2024, focusing on vision-based systems that pair stereo cameras with novel algorithms. This approach eliminates the need for lidar and active sensors, emphasizing stealth and agility for applications in military, space, and agriculture sectors .

## 3. System Requirement Specification

The System Requirement Specification (SRS) is a crucial component in the development lifecycle of any software or hardware project. It defines the system's objectives, outlines user and system expectations, and acts as a formal agreement between stakeholders and developers. The SRS focuses on what the system is expected to do, rather than how it will be built, and serves as a foundation for system design, development, and validation. This chapter elaborates on the system analysis, functional and non-functional requirements, as well as the hardware and software configurations necessary for building a real-time autonomous driving system.

### *3.1 System Analysis*

The development of connected and autonomous vehicles is accelerating, thanks to the rise of smart technologies and Internet of Things (IoT) applications. While fully self-driving cars are still under development, their components—like lane detection, cruise control, and collision avoidance—are becoming increasingly available. These features not only improve safety but also enhance driving convenience and traffic flow. In this project, a small-scale prototype of a self-driving car is built using a Raspberry Pi. It implements key features such as automated braking, lane centering, and route management, all powered by computer vision and convolutional neural networks (CNNs).

### *3.2 Functional Requirements*

The system must allow users to upload image and video datasets for training and evaluation. It should be capable of pre-processing the input data, performing feature extraction, and running transformations like Inverse Perspective Mapping (IPM), Sobel filtering, and Hough Transform to detect lane lines. The CNN model should be trained on selected regions of interest to classify lane boundaries. Based on this, the system must detect the path and overlay it on the original video. In real-time mode, it should automatically brake the vehicle when an object is detected, manage the route to keep the car centered, and adjust the speed dynamically via cruise control. The complete system should function seamlessly to support assisted driving in structured environments.

### *3.3 Non-Functional Requirements*

The system should be platform-independent and capable of running on different machines with minimal configuration. It must ensure high availability and handle increasing user loads over time. Security and maintainability are essential, with the code adhering to clean architecture standards, naming conventions, and modular libraries. Load balancing and random validation checks should be implemented to ensure that system performance remains optimal even under varying conditions. Moreover, it must support easy scalability and deployment.

### *3.4 Hardware Requirements*

The desktop setup for development and testing includes an Intel Core i7 processor (2.4 GHz), 32 GB of RAM, and a 512 GB SSD for fast read/write operations. The embedded system prototype consists of a Raspberry Pi 4, powered by a stable 5V/3A supply. A compatible camera module is used for capturing road images and video in real time. The Coral Edge TPU is employed for accelerating deep learning inference, and a motorized car chassis is used for implementing physical movements and testing navigation.

### *3.5 Software Requirements*

The system runs on Windows 10 (64-bit) and is developed using Python 3.6. Anaconda is used as the main package and environment manager. Key libraries include TensorFlow for deep learning, Keras for model creation, NumPy for numerical computations, and OpenCV for image processing. The Python programming language is chosen for its flexibility, large support community, and strong library ecosystem. TensorFlow and Keras provide robust tools for developing, training, and deploying neural network models, while NumPy and OpenCV handle matrix operations and computer vision tasks effectively.

### *3.6 Embedded Components*

The Raspberry Pi serves as the central processing unit of the prototype vehicle. It operates on a Linux-based OS and supports Python as the primary language for programming and control. Its GPIO pins allow hardware components such as cameras and motors to be interfaced directly. The Coral Edge TPU enhances model inference speed without burdening the CPU. The camera module captures live road video for lane detection and object avoidance, and the car chassis enables testing of real-time autonomous navigation.

### *4.1 System Architecture*

The architecture of the proposed system for lane detection, classification, and collision distance estimation is designed to enable real-time performance and high accuracy in structured driving environments. As illustrated in **Figure 1**, the system is composed of five core modules: **Obstacle Image Segmentation**, **IPM and Sobel Filtering**, **Hough Transformation and Lane Detection**, **Host Lane Classification**, and **Collision Distance Detection**. These modules work in tandem to provide a robust forward collision warning system suitable for deployment on embedded platforms such as Raspberry Pi.

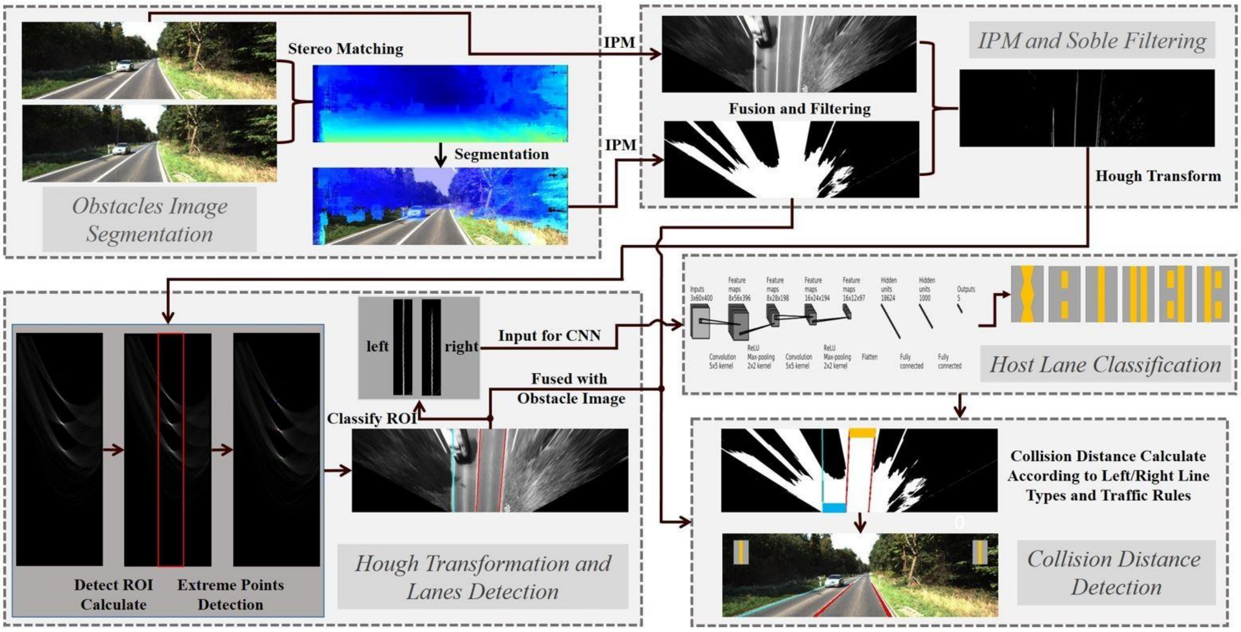



**Figure 1:** System architecture for real-time lane detection, classification, and forward collision warning. The pipeline consists of five primary modules: Obstacle Image Segmentation, IPM and Sobel Filtering, Hough Transformation and Lane Detection, Host Lane Classification, and Collision Distance Detection. Each stage processes stereo image inputs and applies computer vision and deep learning techniques to enhance driving safety in structured environments.

The first stage, **Obstacle Image Segmentation**, begins with stereo image acquisition from dual cameras mounted at the front of the vehicle. Using a GPU-accelerated stereo matching algorithm, a disparity map is generated to estimate the depth of each pixel in the scene. This depth information is critical for identifying and segmenting obstacles from the drivable road surface. To enhance accuracy, a modified UV-disparity segmentation algorithm is applied, enabling efficient separation of road areas and obstructions even under variable lighting and occlusion conditions.

The next step is **Inverse Perspective Mapping (IPM)** and **Sobel Filtering**. Both the original image and the segmented obstacle image are transformed into a bird's-eye view using IPM. This transformation flattens the perspective, making it easier to identify lane geometry. A Sobel filter is applied to the IPM-transformed images to highlight edge information, which is crucial for detecting lane boundaries. The filtered outputs are fused to generate a low-noise grayscale bird-view image, enhancing the visibility of relevant lane markers and suppressing irrelevant textures or noise.

Following preprocessing, the system performs **Hough Transformation and Lane Detection**. The low-noise bird-view image is processed using the Hough Transform to identify lines corresponding to lane markings. A histogram of angular orientations in Hough space is computed to statistically determine the most probable lane directions. The system then defines a region of interest (ROI) based on the highest voting angles and detects extreme points within this ROI, ultimately identifying left and right lane boundaries with high precision.

In the **Host Lane Classification** module, the extracted ROIs are passed into a Convolutional Neural Network (CNN) for classification. The CNN is trained to distinguish between left and right lane lines and to recognize the lane where the ego-vehicle is currently traveling. The classification output is fused with the obstacle segmentation map to filter out occluded or obstructed lanes. This module plays a critical role in supporting Advanced Driver Assistance Systems (ADAS) functions like lane keeping and lane changing.

The final component, **Collision Distance Detection**, utilizes the segmented obstacle map along with the classified lane information to compute the distance between the ego-vehicle and any detected object within its path. This module calculates the shortest distance to an obstacle, taking into account the type of lane line (solid or dashed), lane position (left or right), and applicable traffic rules. The result is an accurate estimation of collision risk, which can be used to trigger alerts or initiate automated braking.

### *4.2 Object Orientation design*

Hardware Design:

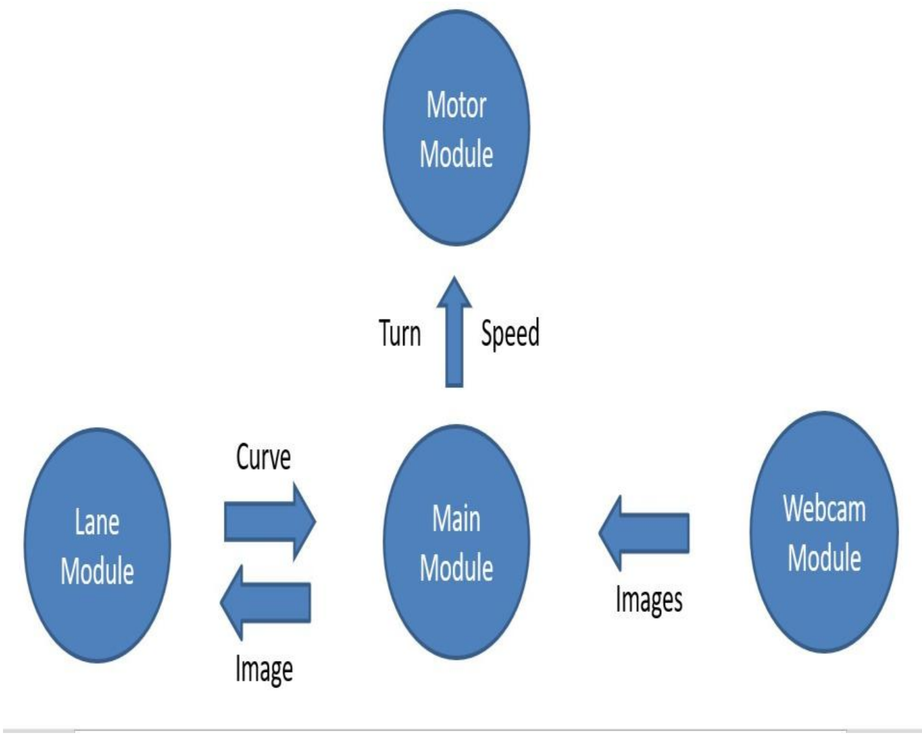
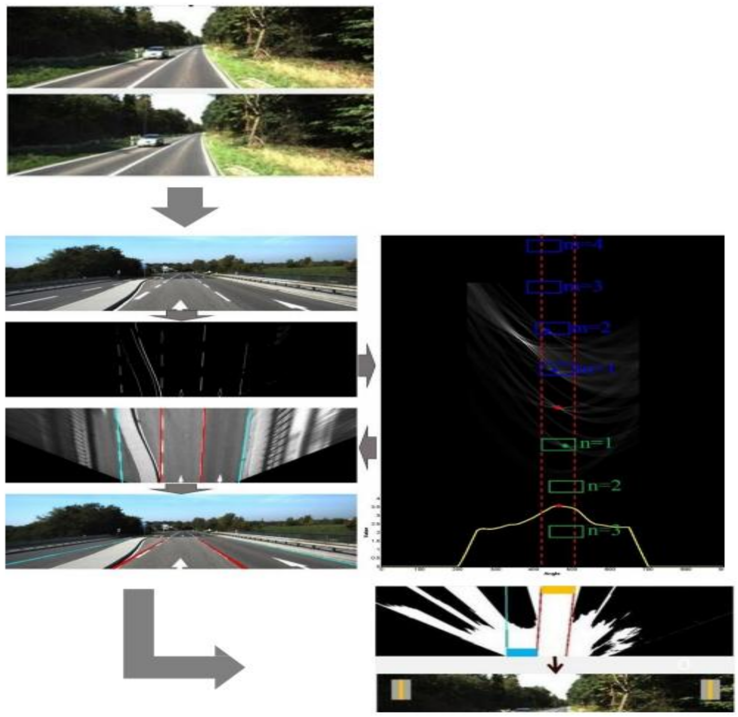



Fig 2 Hardware Design Prototype

***4.3 Input/Output Design***




***4.4 Algorithmic Models and Techniques***

In the development of an intelligent vision-based autonomous vehicle system, the selection and integration of learning algorithms and feature extraction methods play a critical role in ensuring system robustness, reliability, and real-time adaptability. This section elaborates on the key models and algorithms utilized in the system—including Convolutional Neural Networks (CNNs), AdaBoost, and Hough Transform—and explains their importance in solving complex vision problems such as lane detection, object classification, and decision-making in dynamic road environments.

**4.4.1 Convolutional Neural Network (CNN)**

Convolutional Neural Networks (CNNs) have emerged as the cornerstone of modern computer vision, powering applications in autonomous driving, medical imaging, and real-time surveillance. In this system, CNNs are employed to perform end-to-end mapping of raw pixel data from the vehicle's front-facing camera to steering angle predictions and lane classification outputs.

CNNs consist of multiple layers, including convolutional layers, pooling layers, ReLU (Rectified Linear Unit) activations, and fully connected layers. These layers work in hierarchy: low-level features such as edges, lines, and textures are captured in the early layers, while deeper layers learn high-level abstractions like lane curvature, vehicle boundaries, or road patterns.

One of the primary advantages of CNNs is their ability to learn features automatically. Unlike traditional machine learning approaches that require manual feature engineering, CNNs optimize filters during training through backpropagation. This self-learning capability makes them especially useful in unstructured or semi-structured environments, where the complexity of lane markings and road textures may vary significantly.

In our system, CNNs are trained using a dataset comprising annotated lane images and obstacle-labeled scenes from structured roads. Through supervised learning, the network becomes capable of detecting:

- Left and right lane boundaries
- The ego lane (host lane)
- Drivable area boundaries
- Situations that require lane change or obstacle avoidance

The CNN model's architecture includes multiple convolutional blocks followed by max-pooling layers, which reduce the spatial dimension and help in achieving computational efficiency. Fully connected layers near the output generate classification scores and regression values, including steering angles or turn signals.

The lightweight design of the CNN ensures that it can be deployed on embedded hardware such as the **Raspberry Pi 4** with support from an **Edge TPU** (Coral Accelerator), making it suitable for real-time performance without the need for GPUs.

#### 4.4.2 AdaBoost (Adaptive Boosting)

AdaBoost, short for Adaptive Boosting, is an ensemble learning algorithm that enhances classification accuracy by combining multiple weak classifiers into a strong one. In the context of this project, AdaBoost is applied for lane boundary validation and obstacle presence detection, particularly in challenging frames where CNN confidence is low or visual noise is high.

AdaBoost begins by assigning equal weights to all training samples. A weak learner, such as a shallow decision tree (stump), is trained, and misclassified samples are given higher weights in the next iteration. This iterative focus on harder-to-classify instances ensures that the ensemble model progressively improves with each weak learner added. The final prediction is a weighted majority vote of all learners.

In our autonomous system, AdaBoost is used in two critical areas:

1. **Post-processing validation of CNN outputs**: Frames flagged with low-confidence by the CNN are passed through an AdaBoost-based classifier trained on key lane geometry features (e.g., slope consistency, curvature continuity).

2. **Dynamic obstacle tagging**: When obstacles appear partially occluded or indistinct, AdaBoost helps refine classification using simpler but fast geometric and spatial features.

The primary strength of AdaBoost lies in its simplicity and high accuracy for binary classification, especially in limited data conditions or edge-case scenarios where deep learning models might struggle.

#### 4.4.3 Hough Transform

The Hough Transform is a classical computer vision technique used to detect geometric shapes such as lines, circles, and ellipses in images. In this project, it is utilized in the **lane detection pipeline** after applying Sobel filtering and Inverse Perspective Mapping (IPM). The transform maps image points into a parameter space where intersections represent the presence of geometric features.

For detecting straight lines, each point in the edge-detected image is transformed into a sinusoidal curve in the ($\rho$, $\theta$) parameter space, where:

- $\rho$ is the perpendicular distance from the origin to the line
- $\theta$ is the angle between the x-axis and the line normal

Lines are identified by finding local maxima in the accumulator array where many curves intersect, indicating the presence of a line in the original image.

This voting-based detection approach makes the Hough Transform highly resistant to noise, shadows, and partial lane occlusions—factors common in real-world driving. In our implementation:

- The Hough Transform is applied on bird's-eye view images
- Only dominant lines (based on accumulator thresholds) are retained
- Line parameters are fed into downstream modules for curvature estimation and center-line calculation

It serves as a key bridge between raw image edges and meaningful lane geometry, helping anchor the CNN-based detections to geometric reality.

#### 4.4.4 Data Augmentation and Preprocessing

To enhance generalization and reduce overfitting during training, data augmentation is extensively used. Various transformations are applied to the dataset, including:

- Image shifting and translation (simulating vehicle drift)
- Rotation (simulating curve and intersection entry)
- Brightness/contrast adjustments (handling time-of-day changes)
- Flipping and mirroring (to balance left/right lane detections)

All images are resized and normalized before feeding into the CNN. These augmentations increase the diversity of training data and help the model perform better under varied environmental conditions.

In addition to image augmentation, synchronized driving logs (e.g., throttle, brake, steering) are time-aligned with video frames to create rich feature-label pairs. This synchronization enables supervised learning not just for classification, but also for behavior cloning and control.

#### 4.4.5 Real-Time Optimization with Edge Computing

Given the deployment on resource-constrained platforms like **Raspberry Pi**, real-time performance is achieved by:

- Reducing model size via pruning and quantization
- Offloading inference to **Coral Edge TPU** for CNN models
- Utilizing OpenCV's optimized implementations of image processing routines
- Batch preloading and asynchronous frame handling

The entire pipeline is designed to run at 15–20 FPS (frames per second), making it feasible for small-scale autonomous vehicles or robotics research platforms.

## 5. Conclusion

In this research, we have successfully designed and developed a vision-based lane detection and forward collision warning system using stereo imaging and deep learning techniques. By leveraging convolutional neural networks (CNNs), classical image processing methods like the Hough Transform, and ensemble models such as AdaBoost, the proposed system is capable of identifying driving lanes, estimating lane curvature, classifying host lanes, and detecting obstacles in real time. These capabilities form the foundational modules for intelligent autonomous navigation and contribute directly to the goals of Advanced Driver Assistance Systems (ADAS).

The integration of Inverse Perspective Mapping (IPM), Sobel filtering, and stereo vision disparity mapping has proven effective in transforming raw camera data into reliable, noise-resistant representations of the road. Our system's architecture was carefully designed to run efficiently on embedded hardware platforms like Raspberry Pi, further enhanced with Coral Edge TPU for accelerated neural inference—making it both low-cost and deployable for real-world testing in controlled environments.

A key advantage of our system is its adaptability to structured environments, such as highways and urban roads, as well as its ability to generalize to situations with varying lighting, partial occlusion, and road markings. Furthermore, the data augmentation strategies used in training helped increase the robustness of the model under diverse conditions.

In conclusion, the proposed lane detection and collision warning system demonstrates that a hybrid approach—combining machine learning, classical computer vision, and real-time optimization—can effectively support autonomous driving objectives on limited hardware. While this system performs well in simulated and semi-controlled scenarios, future work will focus on extending its capabilities to unstructured roads, improving prediction under extreme weather conditions, and incorporating vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) communication for enhanced decision-making.

## 6. References

[1] A. Gambi, T. Huynh, and G. Fraser, "Automatically Reconstructing Car Crashes from Police Reports for Testing Self-Driving Cars," in *Proc. IEEE/ACM Int. Conf. on Software Engineering (ICSE)*, pp. 1–10, May 2019.

[2] A. Rassõlkin, T. Vaimann, A. Kallaste, and R. Sell, "Propulsion Motor Drive Topology Selection for Further Development of ISEAUTO Self-Driving Car," in *Proc. Int. Symposium on Power Electronics (Ee)*, pp. 1–6, Nov. 2018.

[3] Q. Zou et al., "Robust Lane Detection from Continuous Driving Scenes Using Deep Neural Networks," *IEEE Trans. Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5156–5166, Mar. 2019.

[4] M. Muller, "AsFault: Testing Self-Driving Car Software Using Search-Based Procedural Content Generation," in *Proc. IEEE/ACM Int. Conf. on Software Engineering Companion (ICSE-Companion)*, pp. 31–34, May 2019.

[5] J. Newman, Z. Sun, and D. J. Lee, "Self-Driving Cars: Platform for Learning and Research," in *IEEE Int. Electronics Technology Conf.*, Nov. 2020, doi: 10.1109/IETC47856.2020.9249142.

[6] L. Wang and B. K. P. Horn, "Time-to-Contact Control for Safety and Reliability of Self-Driving Cars," in *Proc. IEEE Smart Cities Conf.*, pp. 1–6, Nov. 2017.

[7] A. K. Jain, "Working Model of Self-Driving Car Using Convolutional Neural Network, Raspberry Pi and Arduino," in *Proc. Int. Conf. on Electronics, Communication and Aerospace Technology (ICECA)*, Oct. 2018, doi: 10.1109/ICECA.2018.8474620.

[8] T. Holstein, G. Dodig-Crnkovic, and P. Pelliccione, "Real-world Ethics for Self-Driving Cars," *IEEE Technology and Society Magazine*, vol. 39, no. 4, pp. 34–45, Dec. 2020.

[9] M. Daily, S. Medasani, R. Behringer, and M. Trivedi, "Self-Driving Cars," *IEEE Computer*, vol. 50, no. 12, pp. 18–23, Dec. 2017.

[10] Á. Takács, I. Rudas, D. Bösl, and T. Haidegger, "Highly Automated Vehicles and Self-Driving Cars," *IEEE Robotics & Automation Magazine*, vol. 25, no. 4, pp. 82–89, Dec. 2018.

[11] P. Pandian et al., "MAB-Ensemble: A Multi-Armed Bandit Approach for Robust Lane Detection in Autonomous Driving," *Scientific Reports (Nature)*, vol. 15, no. 1, Apr. 2025.

[12] DeepRoute.ai & Qualcomm, "China's DeepRoute.ai to Team Up with Qualcomm on ADAS Development," *Reuters*, Apr. 8, 2025. [Online]. Available: https://www.reuters.com

[13] Southwest Research Institute (SwRI), "SwRI Develops Off-road Autonomous Driving Tools Focused on Camera Vision," *Press Release*, Mar. 2024. [Online]. Available: https://www.swri.org