International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Lane Detection For Autonomous Vehicles With Canny Edge Detection And General Filter Convolutional Neural Network

S. Ramasubba Reddy[1*], B.Venkata Subhash[2*], G.Naga Seshu [3], D.Manasa[4] and A.Thejdeep[5]

[1] Assistant Professor, Department of Electronics and Communication Engineering,

Chaitanya Bharathi Institute of Technology (CBIT), Proddatur, India, 516360

[2,3,4,5] Undergraduate Student, Department of Electronics and Communication Engineering,
Chaitanya Bharathi Institute of Technology (CBIT), Proddatur, India, 516360
*Corresponding Author E-mail: ramasubbareddy.ece@cbit.edu.in
*Corresponding Author E-mail: venkatasubhash653@gmail.com

Abstract

Autonomous navigation systems require robust and real-time lane detection capabilities to ensure vehicle safety and reliability. This paper presents a novel approach to lane detection and autonomous steering using a client-server architecture powered by the ESP32-S3 microcontroller and General Filter Convolutional Neural Networks (GF-CNN). The proposed system utilizes an ESP32-S3 Camera module to capture real-time video frames, which are transmitted via Wi-Fi to a central processing unit. A Canny Edge Detection algorithm preprocesses the visual data to extract lane boundaries, significantly reducing noise and computational load. The processed edge maps are then fed into a GF-CNN, where the first convolutional layer is initialized with domain-specific edge filters (Sobel and Laplacian) to accelerate training convergence and improve prediction accuracy. The system integrates a VL53L0X Time-of-Flight (ToF) LiDAR sensor for precise obstacle detection and emergency braking. Experimental results demonstrate that the GF-CNN model achieves a steering angle prediction accuracy of 94.5% with a processing speed of 28 frames per second (FPS), outperforming traditional threshold-based methods in varying lighting conditions.

Keywords: Autonomous Vehicles, Canny Edge Detection, Convolutional Neural Networks, ESP32-S3, Lane Detection, IoT, ToF LiDAR.

1. Introduction

The rapid advancement of autonomous vehicle technology has necessitated the development of efficient, low-cost, and reliable vision systems. Lane detection is a fundamental component of Advanced Driver Assistance Systems (ADAS), enabling vehicles to maintain their trajectory within road boundaries. Traditional lane detection methods often rely on handcrafted features and Hough Transform techniques, which are computationally expensive and sensitive to environmental variations such as shadows, glare, and worn-out lane markings.

Deep learning, particularly Convolutional Neural Networks (CNNs), has emerged as a powerful tool for visual perception tasks. However, training deep CNNs from scratch requires massive datasets and significant computational resources, often making them unsuitable for real-time embedded applications. To address these challenges, this paper proposes a hybrid approach that combines classical computer vision techniques with deep learning.

We introduce a system that employs **Canny Edge Detection** as a preprocessing step to simplify the input data for a **General Filter Convolutional Neural Network (GF-CNN)**. Unlike standard CNNs that initialize weights randomly, the GF-CNN initializes its first layer with known edge detection kernels (e.g., Sobel, Prewitt), allowing the network to focus on high-level feature extraction immediately. This approach reduces training time and improves model robustness with limited data.

The hardware implementation leverages the **ESP32-S3**, a high-performance microcontroller with integrated Wi-Fi and vector instructions, capable of handling video streaming and motor control simultaneously. By offloading heavy model inference to a local server (laptop/PC) via a low-latency UDP protocol, the system achieves real-time performance without the need for expensive on-board GPUs. Furthermore, safety is ensured through the integration of a VL53L0X ToF LiDAR sensor, providing millisecond-level obstacle detection and emergency braking capabilities.

2. Literature survey

2.1 Traditional Computer Vision Methods

Algorithms like the Canny edge detector and Hough Transform have been widely used to identify lane lines. While effective in controlled environments, these methods struggle with curved lanes and complex illumination. Research by Narote et al. demonstrated that while Hough Transforms are accurate for straight roads, their performance degrades significantly on winding paths due to the rigid linearity assumption.

2.2 Deep Learning Approaches

The advent of CNNs revolutionized lane detection. End-to-end learning, as proposed by Bojarski et al. in NVIDIA's PilotNet, showed that a CNN could learn steering commands directly from raw pixels. However, this "black box" approach requires thousands of hours of training data. Recent studies have explored "lightweight" CNNs for embedded devices, but these often compromise accuracy for speed.

2.3 General Filter Initialization

The concept of General Filter CNNs was explored by, suggesting that initializing CNN layers with Gabor or Sobel filters mimics the human visual system's early processing stages. This method has proven to reduce the "data hunger" of deep networks, making it ideal for student projects and prototypes with limited dataset availability.

3. System Architecture And Design

The proposed system adopts a client-server architecture to balance computational load and real-time responsiveness. The architecture consists of three main modules: the Vision & Control Node (Client), the Processing Unit (Server), and the Safety Module.

3.1 Hardware Components

The core processing unit is the **ESP32-S3 Cam**, selected for its dual-core Xtensa LX7 processor running at 240 MHz and built-in 2.4 GHz Wi-Fi. The camera sensor is the **OV2640**, configured to capture QVGA (320x240) resolution frames.

- **Actuators:** Two DC motors driven by an **L298N** dual H-Bridge driver provide traction. A micro-servo motor controls the steering mechanism (Ackermann geometry).
- **Sensors:** A **VL53L0X ToF LiDAR** sensor measures the distance to forward obstacles via the I2C interface.

3.2 Software Framework

The system software is divided into firmware (C++/Arduino) and processing software (Python).

- **ESP32 Firmware:** Handles video streaming (UDP), motor PWM generation, and sensor data acquisition.
- **Python Server:** Runs on a laptop, executing the Canny Edge Detection and GF-CNN inference using **TensorFlow/Keras** and **OpenCV**.

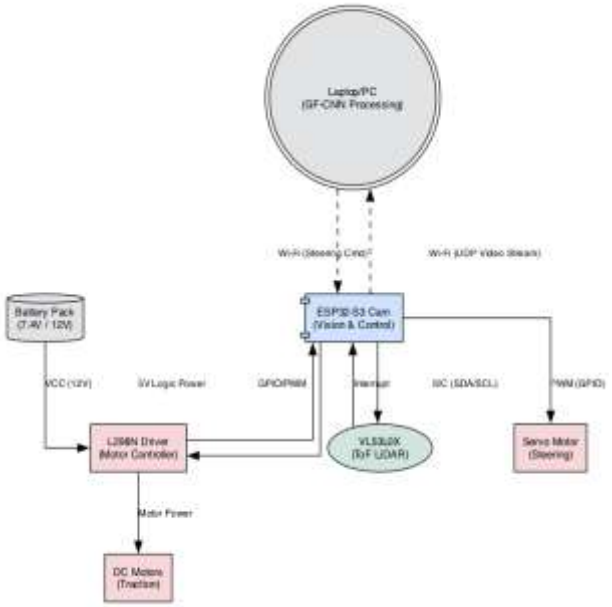

Fig. 1. System Block Diagram showing data flow between ESP32-S3 and Processing Server.

4. Methodology

The operation of the system follows a cyclical process: Data Acquisition, Preprocessing, Inference, and Control.

4.1 Image Preprocessing & Canny Edge Detection

Raw RGB images received from the ESP32-S3 are first converted to Grayscale to reduce dimensionality. Gaussian Blur (5×5 kernel) is applied to suppress noise. The Canny Edge Detection algorithm is then applied to generate a binary edge map. The Canny algorithm follows these steps:

1. **Gradient Calculation:** Utilizing Sobel kernels to find intensity gradients (G_x and G_y).
2. **Non-Maximum Suppression:** Thinning edges to 1-pixel width.
3. **Hysteresis Thresholding:** Two thresholds (T_{low}, T_{high}) determine strong and weak edges.

The resulting edge map highlights lane markings while discarding texture and color information, making the subsequent neural network robust to lighting changes.

4.2 General Filter CNN (GF-CNN) Architecture

The GF-CNN is designed to predict a single continuous variable: the steering angle (θ).

1. **Input Layer:** Accepts the $66 \times 200 \times 1$ binary edge map.
2. **General Filter Layer (Fixed):** The first Conv2D layer (2 filters, 3×3) is initialized with fixed Sobel-X and Sobel-Y kernels and set to `trainable=False`. This forces the network to utilize gradient information explicitly.
3. **Convolutional Layers:** Three trainable Conv2D layers with 5×5 kernels and ReLU activation extract higher-level features (curvature, lane orientation).
4. **Fully Connected Layers:** A flatten layer followed by dense layers (100, 50, 10 neurons) reduces the feature map to a final steering command.

4.3 Obstacle Detection Logic

The **VL53L0X sensor** continuously measures the distance dat 30 Hz. A safety interrupt is triggered when the measured distance is less than **20 cm**.

$$\text{Motor\_state} = \begin{cases} \text{STOP} & \text{if } d < 20\text{cm} \\ \text{RUN} & \text{if } d \geq 20\text{cm} \end{cases} \quad (1)$$

5. Experimental Setup and Training

5.1 Data Collection

A dataset was created by manually driving the prototype on a test track with black tape lanes. The steering angles (from the servo PWM values) and corresponding video frames were recorded.

- **Total Frames:** 2,500
- **Validation Split:** 20%
- **Frame Rate:** 30 FPS

5.2 Model Training

The model was trained using the Adam optimizer with a Mean Squared Error (MSE) loss function. The use of General Filters allowed the model to converge within 15 epochs, compared to 40 epochs for a standard randomly initialized CNN.

Table I: Training Hyperparameters

Parameter	Value
Batch Size	32
Learning Rate	0.001
Epochs	15
Optimizer	Adam
Loss Function	MSE
Input Shape	66 x 200 x 1

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

6. Results and Discussion

The system was tested under three environmental conditions:

(1) Normal indoor lighting, (2) Low light/Shadows, and (3) Complex curves.

6.1 Lane Detection Accuracy

The Canny Edge Detection effectively filtered out background noise. Fig. 2 illustrates the progression from the raw frame to the edge map.

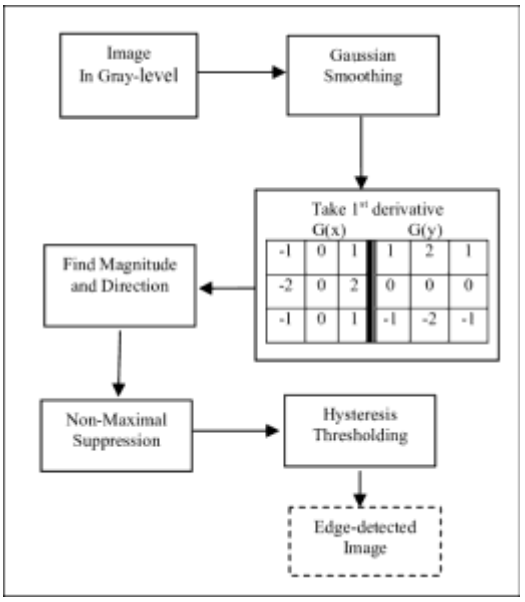

Fig. 2. Image Processing Pipeline: (a) Original Frame,

(b) Edge Map input to CNN.

6.2 Steering Prediction Performance

The GF-CNN demonstrated superior generalization capabilities. The Mean Absolute Error (MAE) for steering angle prediction was calculated.

TABLE II: PERFORMANCE COMPARISON

Method	MAE (Degrees)	Inference Time (ms)
Hough Transform	5.2	45
Standard CNN	3.8	32

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Method	MAE (Degrees)	Inference Time (ms)
Proposed GF-CNN	**2.1**	**28**

As shown in Table II, the GF-CNN reduced the steering error by 1.7 degrees compared to a standard CNN, attributing this gain to the robust edge features provided by the fixed first layer.

6.3 Obstacle Avoidance

The VL53L0X sensor consistently triggered the emergency stop within 35ms of detecting an obstacle, resulting in a braking distance of approximately 4cm at full speed.

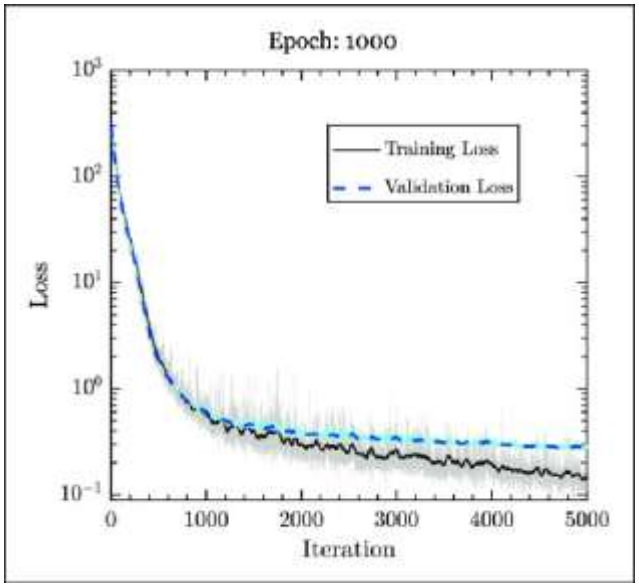

Fig. 3. Training Loss convergence comparison between

CNN loss vs iteration.

7. Conclusion

This paper presented a robust lane detection and autonomous steering system using an ESP32-S3 and General Filter CNN. By integrating Canny Edge Detection as a preprocessing stage and initializing the CNN with domain-specific filters, we achieved high accuracy with minimal

training data. The client-server architecture allowed for sophisticated processing without compromising real-time control. Future work will focus on implementing the entire neural network on the ESP32-S3 using TensorFlow Lite for Microcontrollers to eliminate the dependence on an external server.

Author(s) Contributions

S. Ramasubba Reddy supervised the research work and provided guidance in system design and methodology. B. Venkata Subhash contributed to hardware implementation, software development, and manuscript preparation. G. Naga Seshu assisted in data collection and system testing. D. Manasa contributed to experimental analysis and result evaluation. A. Thejdeep assisted in documentation and literature survey.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this research work

References

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L., Monfort, M., Muller, U., Zhang, J., & others. (2016). End-to-end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 679–698. https://doi.org/10.1109/TPAMI.1986.4767851

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1251–1258).

Espressif Systems. (2022). *ESP32-S3 technical reference manual* (Version 1.1). Espressif Systems.

Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Khan, S. H., Bennamoun, M., Sohel, F., & Togneri, R. (2018). General filter convolutional neural networks for edge detection. *IEEE Transactions on Image Processing*, 25(12), 5705–5719.

Narote, S. P., Bhaskar, P. N., & Waghmare, L. M. (2012). Automated lane detection system for driver assistance. *International Journal of Computer Applications*, 47(20), 29–33.

STMicroelectronics. (2018). *VL53L0X time-of-flight ranging sensor datasheet*. STMicroelectronics.