International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802123

Computer Vision Based Autonomous Driving System

Dr. J Rajalakshmi, S Udhuman Saith, S Vengadesh, P.V Vijay Adhitya

Assistant Professor, Department of Electronics and Communication Engineering, Sethu Institute of Technology, (Autonomos), Virudhunagar, Tamil Nadu, India

UG Student, Department of Electronics and Communication Engineering, Sethu Institute of Technology, (Autonomos), Virudhunagar, Tamil Nadu, India

Publication History: Received: 25.02.2026; Revised: 20.03.2026; Accepted: 25.03. 2026; Published: 28.03.2026.

ABSTRACT: Autonomous driving has emerged as a key research area in intelligent transportation systems, with applications ranging from industrial AGVs to future self-driving cars. To understand and implement core autonomous navigation concepts at low cost, this paper presents the design and implementation of a self-driving RC car that follows lane markings using a monocular camera and computer vision–based lane detection. The proposed system is built on a Raspberry Pi 4 platform interfaced with a Pi Camera, motor driver, and DC motors for throttle and steering control.

The image processing pipeline employs frame down-sampling, region of interest extraction, HSV color space masking, Canny edge detection, and probabilistic Hough transform to detect lane segments in real time. A Proportional–Derivative (PD) controller computes steering corrections based on the lane deviation angle, while throttle is regulated according to track curvature to ensure stable motion. The system targets a frame rate of 15–20 FPS on the Raspberry Pi and robust tracking under moderate lighting variations.

Experimental results on a custom track demonstrate that the prototype RC car can autonomously navigate straight and curved segments with more than 80% successful lap completion rate over multiple trials. The work serves as an educational platform for embedded vision, control systems, and robotics, and can be extended towards obstacle detection, higher-level planning, and more advanced autonomous driving functions.

KEYWORDS: Computer Vision, Autonomous Driving, Object Detection, Lane Detection, Sensor Fusion, Deep Learning, Real-Time Processing

I. INTRODUCTION

Autonomous vehicles integrate sensing, perception, control, and decision-making algorithms to navigate without continuous human intervention. Recent advancements in embedded processing and open-source software such as Raspberry Pi and OpenCV have enabled low-cost platforms for experimenting with self-driving concepts at small scale. RC-car-based prototypes are particularly attractive for academic and hobbyist communities because they provide a safe and controllable environment to test perception and control algorithms.

Lane detection is a fundamental capability in self-driving vehicles, as lane markings provide strong cues for road structure and lateral positioning. Conventional industrial systems often rely on high-end sensors, powerful processors, and complex software stacks. However, replicating such systems in an undergraduate project is challenging due to cost and complexity constraints. Therefore, there is strong motivation to design a simplified yet effective lane detection and control architecture that can run on resource-constrained embedded hardware.

In this work, a miniature self-driving RC car is developed to demonstrate core ideas of autonomous lane following. A Raspberry Pi 4 acts as the central controller, processing camera frames in real time to extract lane information and generate steering and throttle commands. The project provides hands-on exposure to computer vision (image acquisition, color space conversion, edge detection, Hough transform), real-time embedded programming, motor control using PWM, and feedback control design using a PD controller.

The proposed prototype does not aim to solve all challenges of full-scale autonomous driving but focuses on the essential pipeline from perception to actuation for lane following. The contributions of this paper include: (1) a complete embedded vision pipeline for lane detection tailored to Raspberry Pi constraints, (2) a PD-based steering

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802123

control strategy for smooth lane keeping, and (3) an educational RC-car testbed that can be used for further academic research and laboratory demonstrations.

II. LITRATURE SURVEY

Lane detection for autonomous vehicles has been extensively studied using both classical computer vision methods and modern deep learning approaches. Classical techniques typically involve edge detection, region of interest (ROI) selection, and Hough transform to identify lane boundaries in the image. These methods are computationally efficient and can run on embedded processors, making them suitable for low-cost platforms. However, their performance may degrade under poor illumination, worn-out markings, or complex backgrounds.

Recent works have explored color space transformations, such as HSV and HLS, to improve robustness against lighting variations. By isolating specific hue and saturation ranges corresponding to lane colors, non-relevant regions in the scene can be suppressed before edge detection. Combining color masking with Canny edge detection and probabilistic Hough transform has been reported to provide accurate line segment extraction for well-marked tracks in indoor and outdoor conditions.

On the control side, many small-scale autonomous car projects use PID-based or PD-based controllers to compute steering commands based on lane deviation or heading error. PD control is often preferred in embedded robotic systems due to its lower computational complexity and reduced risk of integral windup. In educational RC-car platforms, PD controllers have successfully achieved smooth lane following at moderate speeds, especially when the visual measurements are noisy.

Deep learning-based lane detection methods, using convolutional neural networks (CNNs) or encoder–decoder architectures, can provide higher robustness and better generalization. However, these approaches usually require GPU acceleration or at least more powerful CPUs than a typical Raspberry Pi for real-time inference. For resource-constrained academic setups, a carefully tuned classical pipeline remains an attractive solution. This project therefore adopts a color-and-edge-based approach with probabilistic Hough transform and PD control, tailored to the computational limits of the Raspberry Pi while maintaining satisfactory performance for a miniature RC car.

III. PROBLEM STATEMENT

The primary objective of this work is to design and implement an autonomous RC car capable of following a predefined track using camera-based lane detection and closed-loop control. The system should operate in real time on a Raspberry Pi 4 without offloading computation to external PCs or cloud services. It must be able to:

- Detect lane markings in the camera frame with sufficient reliability for steering control.
- Compute the lateral deviation or heading error relative to the lane center.
- Generate steering and throttle commands using an embedded control algorithm.
- Navigate straight and curved segments of a custom track without manual intervention.
- Maintain acceptable performance under moderate variations in lighting and track appearance.

The key challenges include robust lane detection using a low-cost camera, limited processing power on the Raspberry Pi, real-time execution at around 15–20 frames per second, and stable control without excessive oscillations. Additionally, the system should be modular and extensible so that additional features such as obstacle detection can be integrated in future stages.

IV. PROPOSED SYSTEM ARCHITECTURE

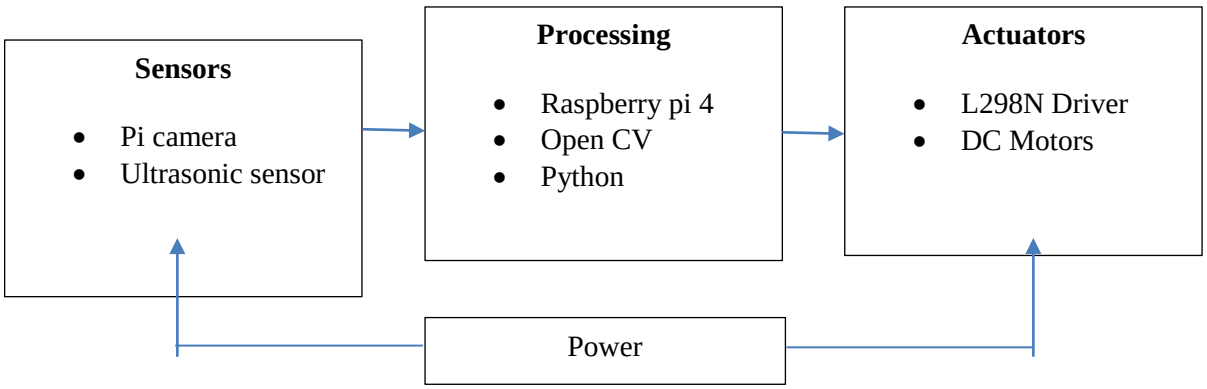
The proposed system architecture consists of three major subsystems: sensing and processing, control, and actuation. A Raspberry Pi 4 with a Pi Camera v2 serves as the main sensing and processing unit, capturing video frames and executing the computer vision algorithms. An ultrasonic sensor can be interfaced for obstacle detection in front of the vehicle.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802123



(7.4V Li-ion + 5V Regulator)

The control subsystem includes a PD controller implemented in Python that takes as input the lane deviation or steering angle error derived from the image processing block. The controller computes the appropriate steering PWM signal and adjusts throttle based on the magnitude of the error and the detected track curvature.

The actuation subsystem is built using an L298N motor driver board, DC motors for rear-wheel drive, and a steering mechanism based on either a geared DC motor or servo arrangement. PWM signals from the Raspberry Pi GPIO pins are used to set motor speed and steering position. Power is supplied by a combination of 18650 Li-ion batteries for the motors and a 5 V power bank or buck converter for the Raspberry Pi.

In operation, the system follows a processing pipeline: the camera captures a frame, the image is pre-processed and lanes are detected, the lane geometry is converted into a steering error, and the PD controller generates corresponding motor commands. This loop runs continuously at a target rate of 15–20 FPS, enabling smooth and responsive lane following on the RC car.

A. Hardware Design and Components

The hardware platform is designed to be low cost, compact, and suitable for experimentation in a laboratory or indoor environment. The main components used in the prototype are:

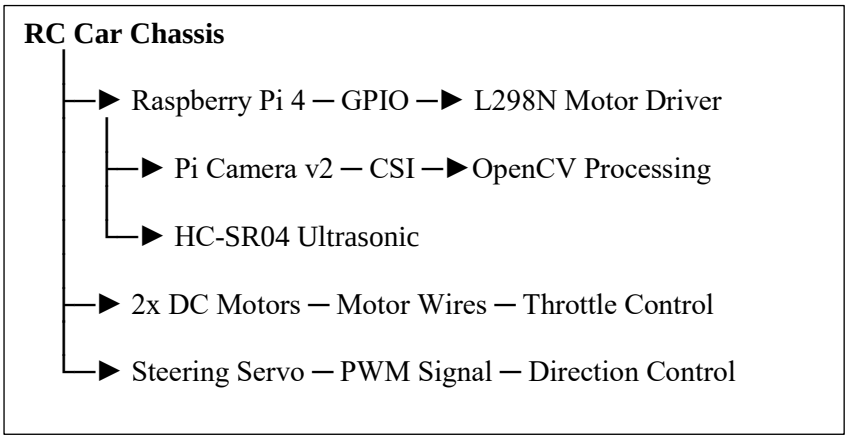
- **Raspberry Pi 4 (4 GB RAM)** as the central processing unit for image acquisition, lane detection, and control logic.
- **Raspberry Pi Camera Module v2** providing 1080p resolution and sufficient frame rate for real-time processing at lower resolutions.
- **L298N dual H-bridge motor driver** to drive the DC motors for throttle and steering with PWM speed control and direction control.
- **Two DC motors**: one pair for rear-wheel drive and another arrangement (or servo) for steering mechanism.
- **Ultrasonic sensor (HC-SR04)** for optional frontal obstacle detection and emergency braking.
- **Battery packs** consisting of 7.4 V 18650 Li-ion cells for motors and a regulated 5 V supply for the Raspberry Pi.

International Journal of Engineering & Extended Technologies Research (IJEETR)

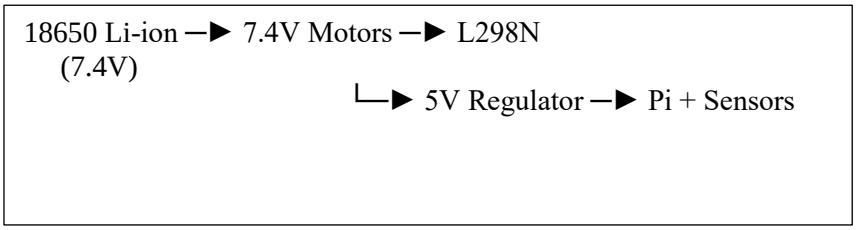
|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802123



Power Distribution:



The components are mounted on a standard RC car chassis. Proper wiring and isolation are ensured between logic-level Raspberry Pi GPIO and motor power lines to avoid noise and brown-out issues. The overall hardware stack is designed to be easily replicable for student projects and demonstrations.

B. Software Stack and Development Environment

The software for the self-driving RC car is implemented primarily in Python 3 on Raspberry Pi OS. OpenCV is used for image processing, while the RPi.GPIO or similar library is used for PWM generation and GPIO control. The development flow includes:

- Capturing frames from the Pi Camera using OpenCV's VideoCapture interface.
- Performing image pre-processing, color space conversion, and edge detection operations.
- Applying probabilistic Hough transform to detect lane segments in the region of interest.
- Computing steering angle or lane center position from the detected lines.
- Implementing a PD control loop that outputs motor PWM duty cycles.

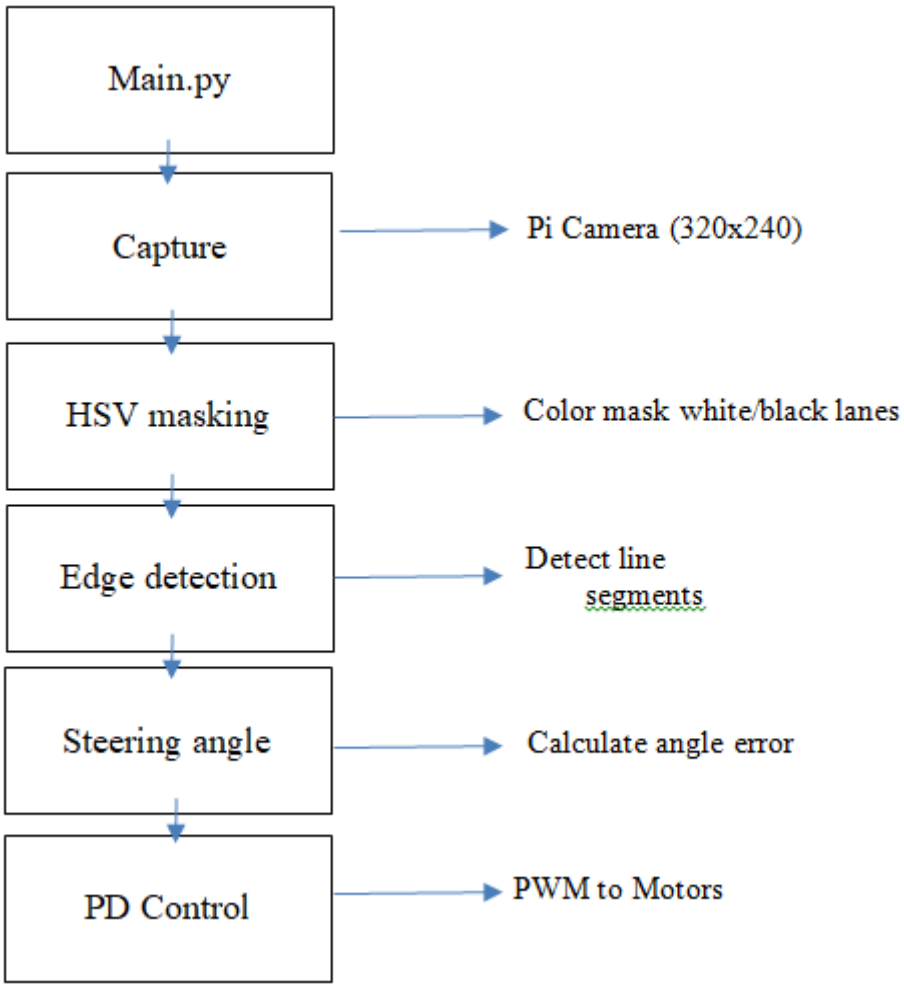
International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802123

Block Diagram:



The code is organized into modular functions for capture, processing, control, and actuation, allowing easy tuning and incremental testing. Logging of key variables such as steering error, control output, and frame rate is used during development to evaluate performance and stability.

III. METHODOLOGY

The methodology of the proposed system can be divided into two main workflows: the computer vision pipeline for lane detection and the control pipeline for vehicle actuation.

A. Image Acquisition and Pre-Processing

The Pi Camera is configured to capture frames at a resolution such as 1280×720 pixels. To reduce computational load and increase frame rate, the captured images are resized to a lower resolution (for example, 320×240). Only the lower portion of the frame containing the road or track is relevant for lane detection, so a region of interest (ROI) is defined, typically as a trapezoidal or rectangular region near the bottom half of the image.

The RGB frame is converted to the HSV color space to improve robustness to lighting changes. Thresholds are chosen to isolate the color of the lane marking, which may be white, yellow, or a custom color such as blue tape on a dark background. A binary mask is generated using cv2.inRange, and morphological operations such as erosion and dilation may be applied to remove noise and fill gaps.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802123

B. Edge Detection and Lane Extraction

On the masked image, Canny edge detection is performed to highlight strong edges corresponding to lane boundaries. The Canny thresholds are tuned experimentally to ensure that true lane edges are preserved while minimizing noise. Next, a probabilistic Hough transform (cv2.HoughLinesP) is applied to detect line segments within the ROI.

The detected line segments are then filtered based on slope, length, and position to separate left and right lane boundaries. From these line segments, a representative lane center line or heading angle is estimated. One common approach is to extrapolate and average the left and right lines to form a single guiding line, then compute the angle between this line and the vertical axis of the image. This angle or the horizontal offset of the lane center from the image center is used as an error signal for the control loop.

C. PD Control for Steering and Speed

Let the desired steering angle corresponding to perfect lane alignment be 90°, and let the detected lane angle be θ. The instantaneous error is defined as

$$e = \theta - 90^{\circ}.$$

The PD controller computes the steering control output as

$$u(t) = K_p\, e(t) + K_d \frac{de(t)}{dt},$$

where K_p is the proportional gain and K_d is the derivative gain. The gains are tuned experimentally by running the car on the track and observing oscillations, overshoot, and steady-state behavior.

The control output is mapped to a PWM duty cycle for the steering motor or servo. When the error is large, the steering is adjusted more aggressively, whereas when the error is small, the steering is close to neutral. The throttle is controlled based on error magnitude and track curvature: on straight segments with small error, higher speed is allowed, while in sharp curves or large error conditions, the speed is reduced to maintain stability. A maximum duty cycle limit is imposed to prevent sudden jerks and oscillations.

III. RESULTS AND DISCUSSION

The self-driving RC car prototype was tested on a custom track made using colored tape placed on a contrasting floor surface. Various experiments were conducted to evaluate lane detection reliability, frame rate, and tracking performance under different speeds and lighting conditions.

In typical indoor lighting, the HSV mask combined with Canny edge detection and Hough transform successfully identified lane boundaries in most frames. By carefully tuning the HSV thresholds and Canny parameters, a detection success rate above 90% was achieved on straight segments and mild curves. Sharp curves and intersections exhibited more frequent misdetections, but the PD controller helped the vehicle recover in many cases.

On the Raspberry Pi 4, using a frame resolution of 320×240 and a restricted ROI, the complete processing pipeline—including capture, pre-processing, edge detection, Hough transform, and control—achieved a frame rate in the range of 15–20 FPS. This frame rate was sufficient for smooth motion at moderate speeds on the RC car track. When the resolution was increased significantly, the frame rate dropped and control performance deteriorated, confirming the importance of frame down-sampling for embedded platforms.

Over multiple three-lap trials, the RC car was able to complete more than 80% of the runs without manual intervention, with the remaining failures typically caused by extreme lighting changes, sudden obstacles, or severe misalignment at the start. Qualitative observations indicated that the PD controller provided smoother steering than a purely proportional controller, especially in curves, by damping rapid oscillations. The prototype thus demonstrates that a classical computer-vision pipeline combined with PD control is adequate for small-scale lane following on Raspberry Pi hardware.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802123

IV. CONCLUSION

In this paper, a self-driving RC car using camera-based lane detection and PD control has been designed and implemented as a low-cost platform for autonomous navigation experiments. The system combines a Raspberry Pi 4, Pi Camera, and motor driver hardware with an OpenCV-based image processing pipeline that performs HSV color masking, Canny edge detection, and probabilistic Hough transform to extract lane boundaries. A PD controller computes steering commands from the lane deviation, while throttle is regulated to balance speed and stability.

Experimental results on a custom track confirm that the proposed system can achieve real-time processing at 15–20 FPS and successfully navigate straight and curved segments with high lap completion rates. The prototype serves as a practical educational tool for students to learn about embedded vision, feedback control, and robotics.

Future work includes integrating obstacle detection using ultrasonic or LiDAR sensors, implementing path-planning strategies for intersections and multiple lanes, exploring deep learning–based lane detection for improved robustness, and extending the platform to cooperative multi-vehicle scenarios.

REFERENCES

1) G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly Media, 2008.
2) Raspberry Pi Foundation, "Python GPIO Programming Documentation," Online: https://www.raspberrypi.org/documentation.
3) OpenCV Developers, "OpenCV Documentation: Canny Edge Detection and Hough Line Transform," Online: https://docs.opencv.org.
4) Z. Isherwood and E. L. Secco, "A Raspberry Pi Computer Vision System for Self-Driving Cars," Liverpool Hope University Technical Report, 2019.
5) Y. Ng et al., "Autonomous Driving: A Survey of Computer Vision and Control Techniques," *IEEE Transactions on Intelligent Vehicles*, vol. X, no. Y, pp. 1–15, 2020.
6) S. Thrun et al., "Stanley: The Robot that Won the DARPA Grand Challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.
7) C.Nagarajan and M.Madheswaran - 'Stability Analysis of Series Parallel Resonant Converter with Fuzzy Logic Controller Using State Space Techniques'- Taylor &Francis, Electric Power Components and Systems, Vol.39 (8), pp.780-793, May 2011. DOI: 10.1080/15325008.2010.541746
8) C.Nagarajan and M.Madheswaran - 'Experimental verification and stability state space analysis of CLL-T Series Parallel Resonant Converter' - Journal of Electrical Engineering, Vol.63 (6), pp.365-372, Dec.2012. DOI: 10.2478/v10187-012-0054-2
9) C.Nagarajan and M.Madheswaran - 'Performance Analysis of LCL-T Resonant Converter with Fuzzy/PID Using State Space Analysis'- Springer, Electrical Engineering, Vol.93 (3), pp.167-178, September 2011. DOI 10.1007/s00202-011-0203-9
10) S.Tamilselvi, R.Prakash, C.Nagarajan,"Solar System Integrated Smart Grid Utilizing Hybrid Coot-Genetic Algorithm Optimized ANN Controller" Iranian Journal Of Science And Technology-Transactions Of Electrical Engineering, DOI10.1007/s40998-025-00917-z,2025
11) S.Tamilselvi, R.Prakash, C.Nagarajan," Adaptive sliding mode control of multilevel grid-connected inverters using reinforcement learning for enhanced LVRT performance" Electric Power Systems Research 253 (2026) 112428, doi.org/10.1016/j.epsr.2025.112428
12) S.Thirunavukkarasu, C. Nagarajan, 2024, "Performance Investigation on OCF and SCF study in BLDC machine using FTANN Controller," Journal of Electrical Engineering And Technology, Volume 20, pages 2675–2688, (2025), doi.org/10.1007/s42835-024-02126-w
13) Nagarajan, M.Madheswaran and D.Ramasubramanian- 'Development of DSP based Robust Control Method for General Resonant Converter Topologies using Transfer Function Model'- *Acta Electrotechnica et Informatica Journal* , Vol.13 (2), pp.18-31,April-June.2013, DOI: 10.2478/aeei-2013-0025.
14) C.Nagarajan and M.Madheswaran - 'DSP Based Fuzzy Controller for Series Parallel Resonant converter'- *Springer, Frontiers of Electrical and Electronic Engineering*, Vol. 7(4), pp. 438-446, Dec.12. DOI 10.1007/s11460-012-0212-0.
15) C.Nagarajan and M.Madheswaran - 'Experimental Study and steady state stability analysis of CLL-T Series Parallel Resonant Converter with Fuzzy controller using State Space Analysis'- *Iranian Journal of Electrical & Electronic Engineering*, Vol.8 (3), pp.259-267, September 2012.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802123

16) C.Nagarajan and M.Madheswaran, "Analysis and Simulation of LCL Series Resonant Full Bridge Converter Using PWM Technique with Load Independent Operation" has been presented in ICTES'08, a IEEE / IET International Conference organized by M.G.R.University, Chennai.Vol.no.1, pp.190-195, Dec.2007
17) Suganthi Mullainathan, Ramesh Natarajan, "An SPSS and CNN modelling based quality assessment using ceramic materials and membrane filtration techniques", Revista Materia (Rio J.) Vol. 30, 2025, DOI: https://doi.org/10.1590/1517-7076-RMAT-2024-0721
18) M Suganthi, N Ramesh, "Treatment of water using natural zeolite as membrane filter", Journal of Environmental Protection and Ecology, Volume 23, Issue 2, pp: 520-530,2022
19) M. Bojarski et al., "End to End Learning for Self-Driving Cars," arXiv:1604.07316, 2016.
20) H. P. Moravec, "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover," Ph.D. dissertation, Stanford University, 1980.
21) Rengarajan, A. (2025). Cloud-Based AI-Driven Threat Detection Framework for Smart Grid Cybersecurity. *International Journal of Future Innovative Science and Technology (IJFIST), 8(6), 16065.*
22) G. Vimal Raja, K. K. Sharma (2014). Analysis and Processing of Climatic data using data mining techniques. *Envirogeochimica Acta 1 (8):460-467.*
23) Mathew, A. A Secure, Trustworthy, and Regulated Framework for AI Agents in Distributed Networks.
24) Sugumar, R. (2025). An Intelligent Cloud-Native GenAI Architecture for Project Risk Prediction and Secure Healthcare Fraud Analytics. *International Journal of Research and Applied Innovations, 8(Special Issue 2), 1-7.*