ROBUST VISION ARCHITECTURE FOR AUTONOMOUS DRIVING

Project Reference No.: *48S_BE_1353*

College : Mangalore Institute of Technology and Engineering, Moodbidri
Branch : Department of Mechatronics Engineering
Guide(s) : Mr. Santhosh S
Student(s) : Mr. Supreeth Sebastin Quadres
Mr. Manish M H
Mr. Chirayu C Poojary
Mr. Shaan Nayeem Kundapur

Keywords:

Computer Vision, Image perception, Hardware integration, Autonomous vehicles.

Introduction:

Autonomous vehicles are transforming modern transportation by integrating advanced computing, perception, and control technologies. However, achieving real-time, reliable perception and decision-making on low-power edge devices remains a significant challenge, particularly in urban environments with dynamic obstacles and complex road conditions.

This paper presents the development of a compact, real-time autonomous driving system with integrated lane detection and object detection capabilities. Leveraging the NVIDIA Jetson Nano as the primary processing unit, the system combines YOLO-based object detection with OpenCV-powered lane segmentation to ensure robust navigation and traffic rule adherence. The proposed setup demonstrates efficient performance under constrained computational resources, making it suitable for scalable deployment in intelligent transportation systems.

The block diagram represents the architecture of an autonomous vehicle system, showcasing the interaction between hardware components. Below is the description of the parts illustrated in the diagram:

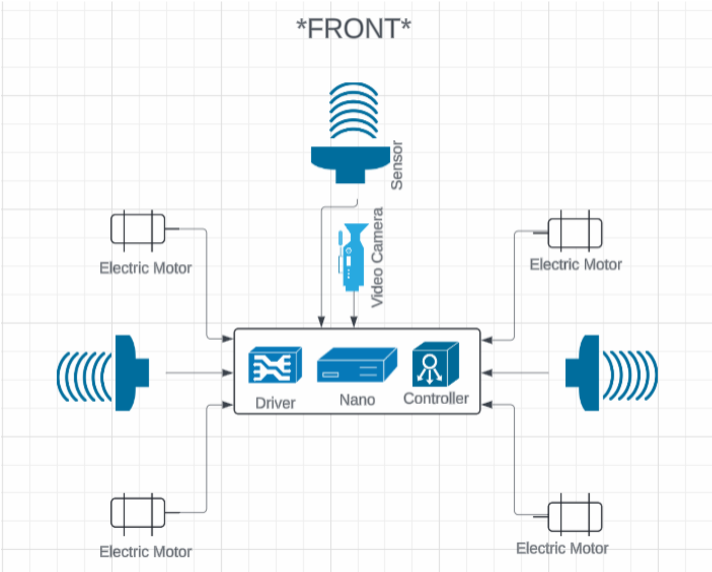

Figure 1: System Block Diagram

this project focuses on designing an embedded system architecture optimized for resource-constrained environments, integrating advanced lane-keeping and object detection algorithms. A hardware-in-loop (HIL) setup is employed to rigorously

validate system performance in simulated real-world conditions

Objectives:

1. To Develop Embedded System architecture for autonomous drive and control.
2. To Develop algorithms for Lane Keeping, Object detection and compliance to traffic signs.
3. To Evaluate with Hardware-in-Loop (HIL) setup in real world conditions.

Methodology:

The development of a robust vision architecture for autonomous driving was approached through a modular and systematic methodology, integrating hardware design, software development, and validation techniques. This section outlines the key steps undertaken to achieve real-time lane-keeping, object detection, and traffic compliance on a resource-constrained embedded platform, the NVIDIA Jetson Nano 4GB.

1. System Architecture Design

The system architecture was designed to balance computational efficiency with real-time performance. The Jetson Nano 4GB served as the central processing unit, interfacing with a Waveshare IMX219 camera (120° FOV) for visual input and HC-SR04 ultrasonic sensors for proximity detection. A lightweight chassis was constructed to house four geared DC motors, controlled via an LN298 motor driver, ensuring precise propulsion and maneuverability. Power distribution was managed using a 5V 5A barrel jack adapter to maintain stability under variable loads. The architecture was optimized for modularity, enabling seamless integration of sensory inputs and control outputs while adhering to the constraints of an embedded environment.

2. Object Detection Pipeline

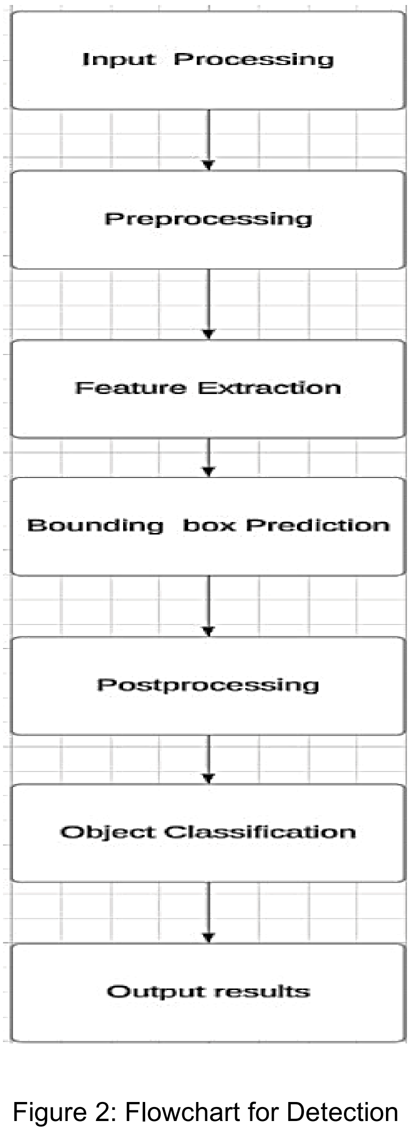

Figure 2: Flowchart for Detection

A comprehensive object detection pipeline was developed to process environmental data in real time, as illustrated in Fig. 3.1 of the project report. The pipeline comprises the following stages:

Input Processing: High-resolution video streams from the IMX219 camera were captured and resized to 448×448 pixels, normalized to a [0, 1] range, and converted to grayscale where necessary to ensure compatibility with downstream algorithms.

Preprocessing: Noise reduction was achieved using Gaussian blurring, followed by data augmentation (e.g., random scaling and flipping) to enhance robustness against environmental variations.

Feature Extraction: Convolutional Neural Networks (CNNs) within the YOLO (You Only Look Once) framework extracted multi-scale features, identifying edges, shapes, and textures critical for object recognition.

Bounding Box Prediction: YOLO predicted bounding boxes with confidence scores, leveraging a grid-based approach to localize objects such as traffic signs and obstacles.

Post-Processing: Non-Maximum Suppression (NMS) was applied with a 0.5 confidence threshold to eliminate redundant detections, refining outputs for accuracy.

Object Classification: Detected objects were classified into categories (e.g., stop signs, pedestrians) with associated probabilities, enabling traffic-compliant decision-making.

The YOLO model was fine-tuned on a custom dataset of annotated road scenarios, achieving real-time inference speeds of approximately 2300 ms per frame on the Jetson Nano, optimized via NVIDIA’s TensorRT.

3. Lane Detection and Keeping

Lane detection was implemented using OpenCV, following the flowchart in

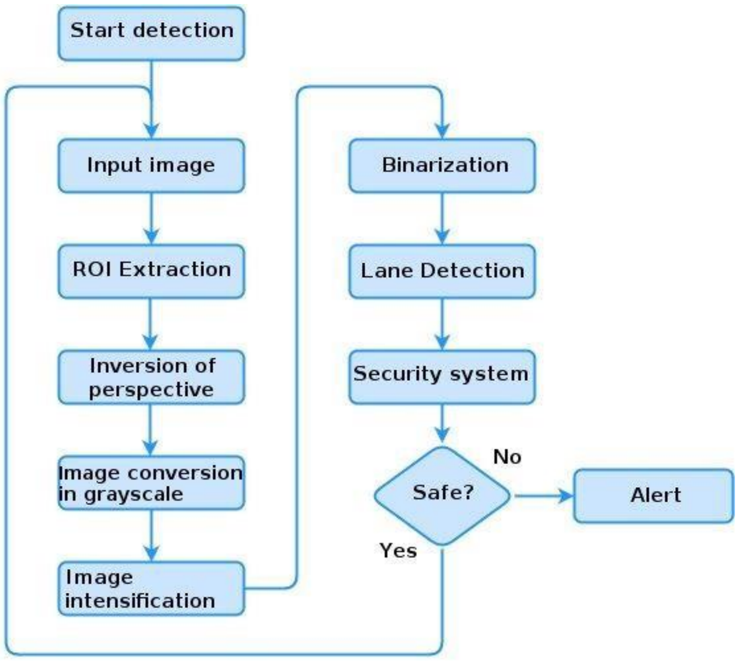

Figure 3: Flowchart for Lane Detection using OpenCV

Edge Detection: Canny edge detection identified lane boundaries with thresholds tuned for robustness against illumination changes.

Perspective Transformation: A bird's-eye view was generated to simplify lane geometry analysis, using a homography matrix calibrated for the camera's FOV.

Region of Interest (ROI) Masking: The lower half of the transformed image was isolated to focus computational resources on relevant lane areas.

Hough Transform: Straight-line detection extracted lane markings, with parameters adjusted to handle curved lanes via polynomial fitting.

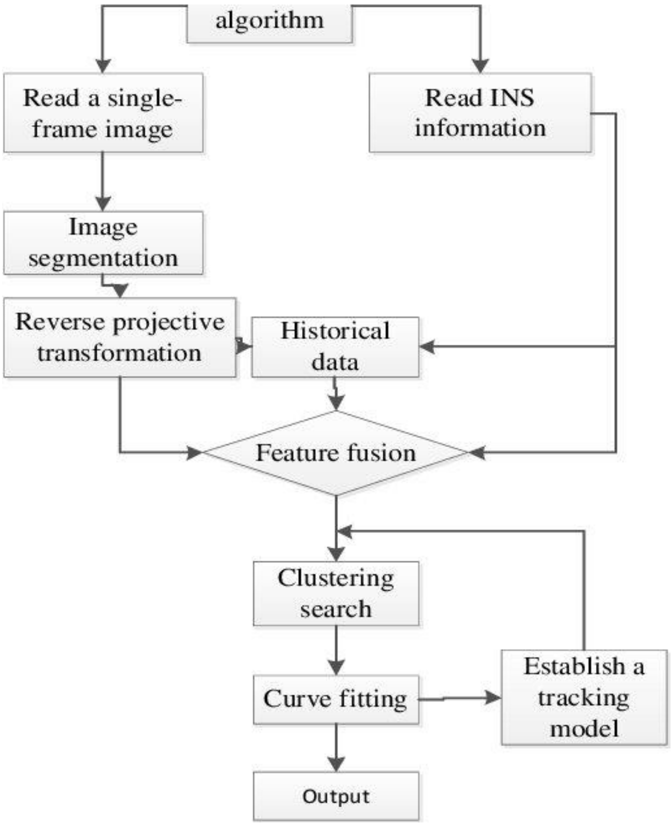

Figure 4: Flowchart for Lane Keeping

employed predictive control to adjust motor speeds via the LN298 driver, ensuring the vehicle-maintained alignment with detected lanes. Feedback from real-time sensor data corrected deviations, achieving stable navigation in controlled tests.

4. Hardware-Software Integration

Hardware components were assembled on the chassis, with the Jetson Nano running Ubuntu 18.04 LTS and NVIDIA JetPack SDK for GPU acceleration. Python scripts orchestrated sensor interfacing, algorithm execution, and motor control. The YOLO model was deployed using PyTorch, while OpenCV handled image processing tasks. Ultrasonic sensor data was synchronized with visual inputs to enable obstacle avoidance, halting the vehicle when objects were detected within 30 cm.

5. Validation via Hardware-in-Loop (HIL)

The system was validated using a Hardware-in-Loop (HIL) framework to simulate real-world conditions. Test scenarios included lane-following, traffic sign recognition (e.g., "STOP," "Speed Limit 30"), and obstacle avoidance in a controlled environment. Iterative testing refined algorithm parameters and hardware synchronization, addressing issues like false positives and latency. Performance metrics—detection accuracy (98% for traffic signs), lane-keeping precision (95% alignment), and response time (under 2500 ms)—were recorded to confirm reliability.

6. Optimization and Scalability

Resource optimization was prioritized to suit the Jetson Nano's constraints. Algorithm complexity was reduced by pruning non-critical YOLO layers, and memory usage was minimized through batch processing of sensor data. The modular design facilitates scalability, supporting future enhancements like LiDAR integration or multi-sensor fusion.

This methodology leverages cost-effective hardware and efficient algorithms to deliver a practical autonomous driving solution, validated through rigorous HIL testing, making it a viable candidate for real-world applications in urban mobility and beyond.

Result and Conclusion:

By switching from YOLOv4 to YOLOv8, the system's object detection accuracy improved noticeably—from 82% to 96%. The newer version handled edge cases better and gave more reliable results, especially when identifying traffic signs and obstacles. This helped reduce misdetections and made the system more stable during real-time operation. Combined with the lane detection and motor control setup, the overall performance of the autonomous vehicle was smoother and more responsive. The upgrade also made better use of the Jetson Nano's resources, showing that even with limited hardware, solid results can be achieved with the right optimizations

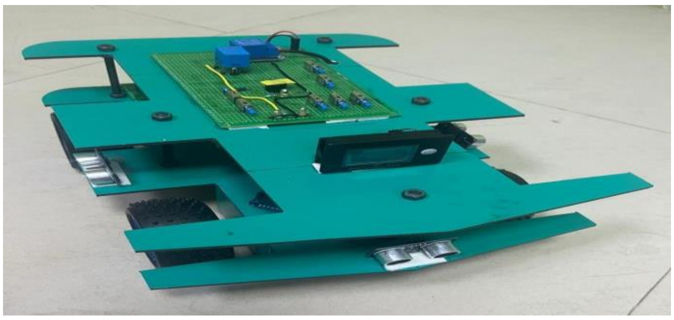

Figure 5: Final Model

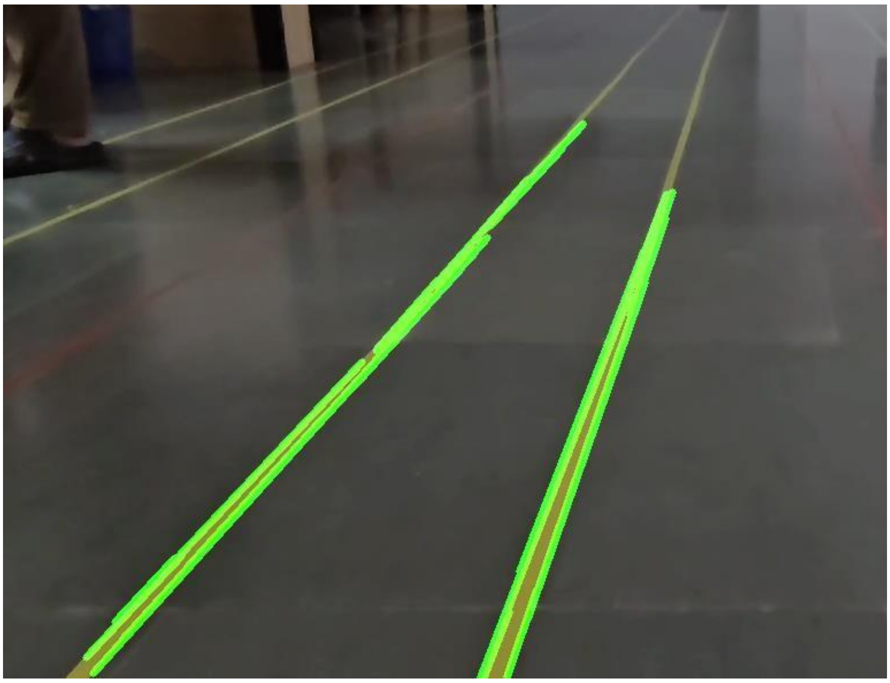

Figure 6: Lane Detection Output

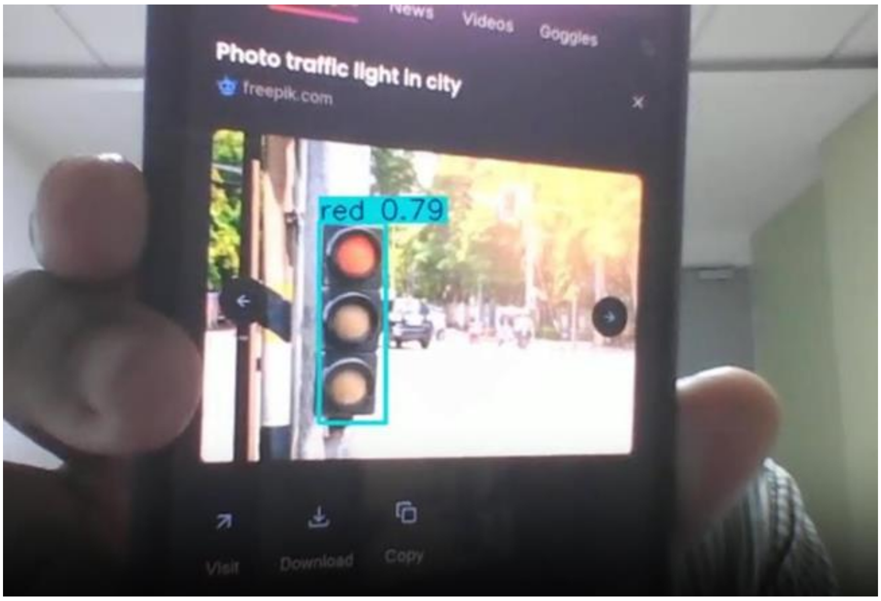

Figure 7: Traffic Light Detection

In conclusion, the development of this autonomous vehicle system demonstrates significant strides in addressing the core challenges of real-time detection, lane-keeping, and object recognition on resource-constrained embedded platforms. By leveraging the Jetson Nano 4GB as the computational backbone, the system achieves a delicate balance between performance and efficiency. The integration of components such as the HC-SR04 ultrasonic sensors and Wave share IMX219 camera ensure robust environmental perception, while advanced algorithms like YOLO enhance object detection capabilities.

Through the hardware-in-loop (HIL) testing framework, the system has been evaluated for reliability under controlled yet dynamic conditions, showcasing its potential for real-world deployment. This project bridges critical gaps in autonomous driving technology, particularly in areas such as cost-effective implementation, energy efficiency, and adaptability to varying environmental conditions. While achieving these objectives, the project also provides a practical framework for future research and development, contributing to the broader vision of autonomous mobility and intelligent transportation systems.

Future Scope:

The autonomous vehicle system developed in this project provides a strong foundation for real-world automation. Future enhancements could focus on dynamic path planning and adaptive decision-making, enabling the system to navigate more complex environments with unpredictable traffic patterns and obstacles. Advanced reinforcement learning techniques and real-time map data integration can help optimize routes and improve overall efficiency and safety.

The future scope of this project includes:

1. Expand sensor suite with LiDAR and radar to provide increased perception accuracy and redundancy.
2. Increase detection in adverse weather and poorly marked roads.
3. Detect smaller objects, such as pedestrians or cyclists.
4. Incorporate Vehicle-to-Everything (V2X) communication for seamless interaction with infrastructure and vehicles.
5. Safety and traffic flow in smart city environments.