ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26sep225

AI-Based Obstacle Avoidance, Line Following and Traffic Light Detection Robot Car Using Arduino Nano 33 BLE Sense with TinyML

Mantu Bera[1]

[1]AI Engineer, Xgboost AI Technology (OPC) Private Limited, Debra, Paschim Medinipur, West Bengal, India

Publication Date: 2026/09/19

Abstract: Autonomous vehicles require intelligent perception and decision-making systems to safely navigate dynamic environments. This paper presents the design and implementation of an Artificial Intelligence (AI) based robotic car capable of line following, obstacle avoidance, and traffic light detection using the Arduino Nano 33 BLE Sense microcontroller. The system integrates traditional sensor-based control with Tiny Machine Learning (TinyML) models deployed directly on the microcontroller. Infrared sensors are used for lane detection, an ultrasonic sensor is used for obstacle detection, and the onboard camera module performs traffic light recognition using a lightweight convolutional neural network. The TinyML model is trained with TensorFlow Lite and deployed with TensorFlow Lite for Microcontrollers. Experimental results show that the proposed system achieves reliable navigation and traffic signal recognition while maintaining low power consumption and real-time performance. The proposed architecture demonstrates the feasibility of implementing AI-based perception systems on resource-constrained embedded hardware.

Keywords: *TinyML, Autonomous Robot Car, Arduino Nano 33 BLE Sense, Obstacle Avoidance, Traffic Light Detection, Line Follower.*

How to Cite: Mantu Bera (2026) AI-Based Obstacle Avoidance, Line Following and Traffic Light Detection Robot Car Using Arduino Nano 33 BLE Sense with TinyML. *International Journal of Innovative Science and Research Technology,* 11(9), 418-420. https://doi.org/10.38124/ijisrt/26sep225

I. INTRODUCTION

Autonomous robotic vehicles have become a major research area in robotics and intelligent transportation systems. Such systems require the ability to perceive the environment, interpret road conditions, and respond to obstacles and traffic rules in real time.

Traditional robotic vehicles rely on rule-based algorithms and simple sensor feedback mechanisms. However, these approaches often fail in complex environments. The emergence of Tiny Machine Learning (TinyML) enables machine learning models to run directly on microcontrollers with limited computational resources.

The Arduino Nano 33 BLE Sense provides an ideal platform for implementing TinyML-based robotic applications due to its integrated sensors, low power consumption, and embedded ARM Cortex-M4 processor.

This research presents a hybrid autonomous robot car system combining: Sensor based obstacle detection, Infrared based lane following and TinyML based traffic light recognition

The integration of these modules enables the robot to mimic basic autonomous driving behavior.

II. LITERATURE REVIEW

Several studies have investigated autonomous robotic vehicles using embedded systems.

Early robotic cars used ultrasonic sensors and simple control algorithms for obstacle avoidance. However, these systems lacked environmental intelligence.

Recent advancements in TinyML have enabled machine learning models to run on microcontrollers. Frameworks such as TensorFlow Lite for Microcontrollers allow neural networks to run on devices with less than 1MB of memory.

Research shows that convolutional neural networks (CNNs) can successfully perform traffic light detection even on embedded devices. By compressing models and using quantization techniques, these algorithms can run efficiently on microcontrollers such as the Arduino Nano 33 BLE Sense.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26sep225

However, limited work has integrated lane following, obstacle avoidance, and traffic light recognition simultaneously on a single low-power microcontroller platform.

➢ *System Architecture*

- *Hardware Components*

The robotic car consists of the following components:

Table 1 Hardware Components

Components	Function
Arduino Nano 33 BLE Sense	Main AI processing unit
Ultrasonic Sensor (HC-SR04)	Obstacle distance measurement
IR Sensors	Line detection
Motor Driver (L293D)	Motor control
DC Motors	Vehicle movement
Camera module	Traffic light image capture
Li-ion Battery	Power supply

The Arduino Nano 33 BLE Sense contains an ARM Cortex-M4 processor running at 64 MHz with built-in sensors, making it suitable for TinyML deployment.

- *System Flow Diagram*

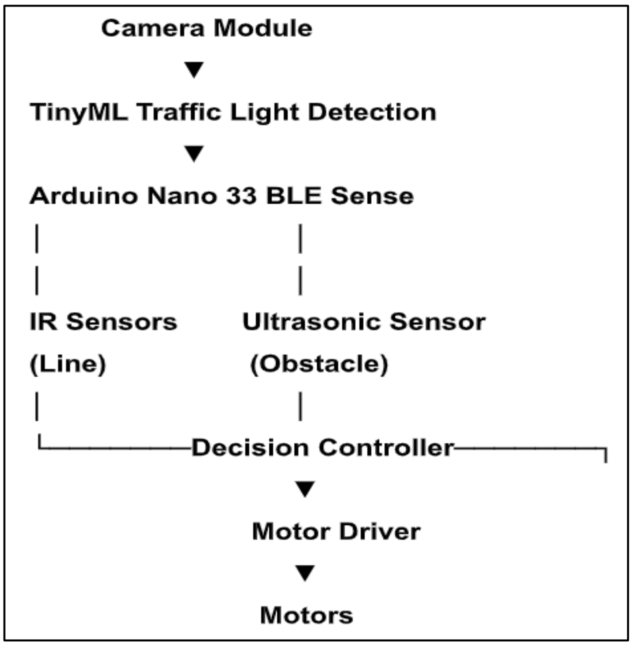

Fig 1 System Flow Diagram

III. METHODOLOGY

➢ *Line Following Algorithm*

The robot uses two IR sensors to detect a black line on a white surface.

- *Control Logic:*

This algorithm ensures that the robot follows the designated path.

Table 2 Control Logic

Left Sensor	Right Sensor	Action
0	0	Move Forward
1	0	Turn Left
0	1	Turn Right
1	1	Stop

➢ *Obstacle Avoidance*

Obstacle detection is implemented using an ultrasonic sensor.

Distance is calculated using:

Distance=(Speed×Time)/2

If an obstacle is detected within 20 cm**,** the robot stops and changes direction.

➢ *Traffic Light Detection using TinyML*

Traffic light detection is implemented using a Convolutional Neural Network (CNN).

- Classes used in training: Red light, Yellow light and Green light
- Dataset images are collected and labeled using Edge Impulse.
- The trained model is converted into a TensorFlow Lite Micro model and deployed on the microcontroller.

- *Inference Steps:*

✓ Step1: Capture image frame
✓ Step2: Preprocess image
✓ Step3: Run TinyML inference
✓ Step4: Predict traffic light color
✓ Step5: Execute action

Table 3 Inference Steps

Detected Color	Robot Action
Red	Stop
Yellow	Slow down
Green	Move forward

➢ *Machine Learning Model*

The CNN model architecture used is:

Table 4 Machine Learning Model

Layer	Configuration
Input Layer	96 × 96 image
Convolution Layer	16 filters
Max Pooling	2×2
Convolution Layer	32 filters
Fully Connected	64 neurons
Output Layer	Softmax (3 classes)

The model is trained using TensorFlow Lite and optimized for microcontroller deployment using quantization techniques.

Model size after optimization: < 200 KB

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26sep225

IV. EXPERIMENTAL RESULTS

The system was tested on a prototype robotic car platform.

Table 5 Experimental Results

Test Scenario	Success Rate
Line following	95%
Obstacle avoidance	93%
Traffic light detection	90%

Latency for TinyML inference: ~60 ms

Power consumption remained low due to the efficient microcontroller architecture.

V. DISCUSSIONS

The experimental results demonstrate that combining traditional sensor-based control with TinyML significantly improves the robot's intelligence and adaptability.

Advantages of the proposed system: Low-power consumption, Real-time inference, Compact hardware and Cost-effective design

However, limitations include: Limited memory for large ML models, Lighting sensitivity in traffic light detection and Camera resolution constraints

Future improvements may include integrating more advanced sensors such as LiDAR and implementing reinforcement learning algorithms.

VI. CONCLUSION

This paper presented the design and implementation of a TinyML-based autonomous robot car capable of line following, obstacle avoidance, and traffic light detection using the Arduino Nano 33 BLE Sense microcontroller.

The integration of machine learning directly on embedded hardware demonstrates the feasibility of developing intelligent autonomous systems without relying on cloud computing or high-power processors.

The results show that TinyML can effectively enable AI functionality in low-resource robotic platforms, paving the way for future developments in compact autonomous vehicles.

FUTURE WORK

Future research directions include: Integration of lane detection using computer vision, Multi-object detection, Navigation based on Reinforcement learning and Implementation on full-scale autonomous vehicles

FUNDING DECLARATION

This Research was self-Funded by Mantu Bera and subsequently commercialized/owned by XGBOOST AI TECHNOLOGY (OPC) PRIVATE LIMITED.

REFERENCES

[1]. Banbury, C., et al. "TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers."

[2]. Warden, P., Situnayake, D. *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers.*

[3]. TensorFlow Lite Micro Documentation.

[4]. Edge Impulse TinyML Development Platform.