iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: IV Month of publication: April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80041

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Motion Sense Rover Using Arduino Nano

Snehal Govardhan[1], Sourabh Singh[2], Duneshwar Patel[3], Rounak Singh Thakur[4], Rohan Kumar Rathore[5]
[1]*Assistant Professior in Computer Science and Engineering Department, Shri Shankaracharya Technical Campus, Bhilai, India*
[2, 3, 4, 5]*UG – Computer Science and Engineering Shri Shankaracharya Technical Campus, Bhilai, India*

Abstract: This project presents a hand gesture–controlled car system designed to improve human–machine interaction using simple and cost-effective technology. This project presents a hand gesture–controlled car system designed to improve s instead of traditional remote controls. The system uses an accelerometer sensor mounted on a hand glove to detect hand gestures such as forward, backward, left, and right movements. These gestures are transmitted wirelessly using an RF transmitter and receiver module. An Arduino Nano micro -controller is used to process the received signals and control the motors of the car through a motor driver module. The system is programmed using Embedded C in the Arduino IDE.

I. INTRODUCTION

Hand Gesture Control Car is an advanced robotics project that allows a car to be controlled using human hand gestures instead of traditional remote controls. This system uses sign language gestures to control the movement of the car such as forward, backward, left, right, and stop. The gestures are detected using sensors and the signals are processed using Arduino Nano microcontroller. The Arduino Nano sends control commands to the motor driver, which controls the motors of the car accordingly. The main aim of this project is to make human-machine interaction more natural, easy, and efficient. In some situations, it is difficult or inconvenient to use physical controllers, especially for physically disabled persons. Also, remote controllers can be lost or damaged, and they have limited range. There is a need for a more natural and wireless control system that can control vehicles using simple hand gestures or sign language. Therefore, the problem is to design a system that can recognize hand gestures and control a car wirelessly using Arduino nano.

II. LITERATURE REVIEW

From the previous research papers, it is observed that most systems use accelerometer sensors, MPU6050 sensors, and wireless communication modules with Arduino to control robot cars using hand gestures. Some recent research uses deep learning and computer vision for more accurate gesture recognition. The existing systems mainly focus on gesture recognition using sensors and transmitting signals wirelessly to control the car movement. However, these systems can be improved by making them low-cost, more accurate, and easy to use.

III. METHODODOLOGY

A. Dataset

The dataset consists of static or dynamic images representing specific hand gestures.

- Data Source: Often created manually using a webcam or sourced from standard datasets like the HaGRID (Hand Gesture Recognition) dataset or ASL (American Sign Language) dataset.
- Classes: Usually 5–6 classes: Forward, Backward, Left, Right, Stop, and Neutral.

B. Preprocessing

- Smoothing: The Arduino takes the average of the last 5 readings so the car doesn't jump at every tiny hand twitch.
- Dead Zone: We tell the Arduino: "If the tilt is less than 10 degrees, ignore it." This allows you to rest your hand without the car moving accidentally.

C. Feature Extraction:

The Arduino takes the raw gravity numbers and turns them into something we understand: **Angles.**

- Forward/Backward: If the "Pitch" angle is more than 20°, the Arduino identifies this as a "Forward" command.
- Left/Right: If the "Roll" angle is more than **20°**, it identifies a "Turn."

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

D. Model

For an Arduino Nano, the model must be computationally "light."

- Threshold Based Classifier (Most Common): A rule-based logic system. If Ax exceeds a specific threshold , move forward.
- K-Nearest Neighbors (KNN): A simple machine learning model The current tilt is compared to a small library of "ideal" gestures stored in the code.
- Tiny ML : Using Edge Impulse to deploy a very small Tiny ML Neural Network (3 layers) directly onto the Nano's flash memory to recognize the complex patterns.

E. Training Process

1. Threshold Setting :

We test different angles to see which feels most natural. We found that a 20-degree tilt is the "sweet spot"—it's enough to be intentional but not enough to tire your wrist.

2. Signal Mapping :

We "train" the system to map the tilt angle to motor speed.

- Low Tilt (20°): Car moves slowly.

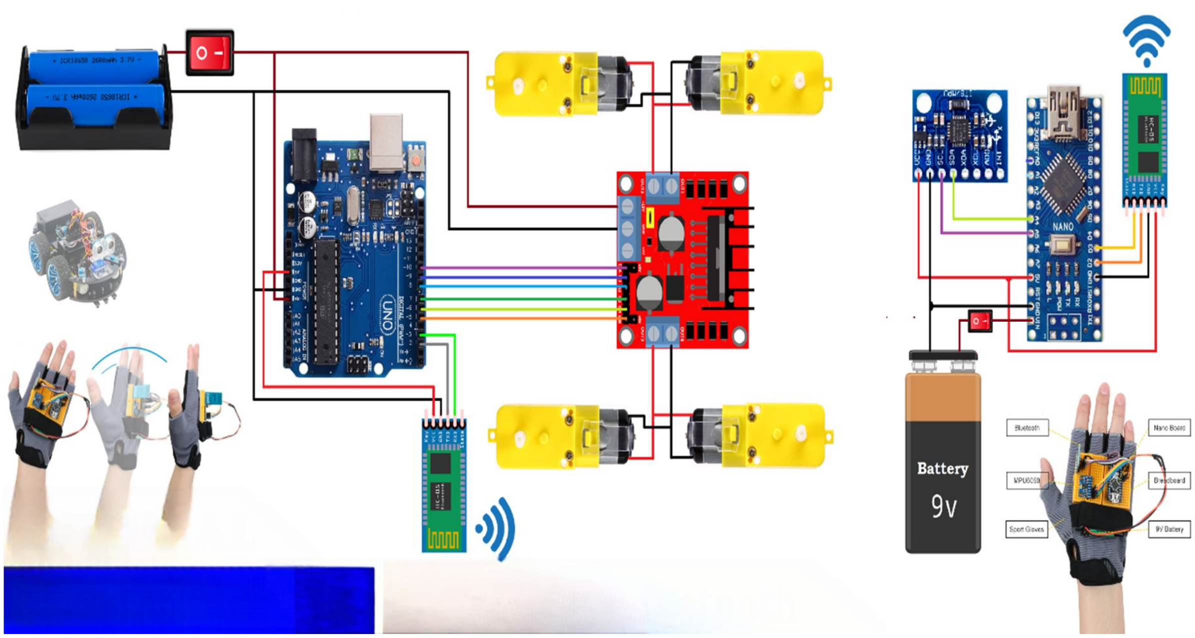
High Tilt (45°): Car moves at full speed.

3. Testing :

The system is tested by performing 100 gestures. If the car moves forward when you tilt forward 95 times out of 100, the system has 95% Accuracy.

IV. SYSTEM ARCHITECTURE

1) Hand Gesture: User moves hand to give direction commands (forward, backward, left, right, stop).
2) MPU6050 Sensor: Detects hand motion (tilt and movement) using accelerometer and gyroscope.
3) Arduino Nano (Transmitter): Processes sensor data and converts it into control signals.
4) Wireless Module (RF / Bluetooth): Transmits signals from hand unit to the car wirelessly
5) Receiver Module: Receives the transmitted signals from the transmitter.
6) Arduino Nano (Receiver): Reads received signals and decides motor actions.
7) Motor Driver (L293D / L298N): Controls speed and direction of DC motors.
8) DC Motors (Car Movement): Move the car based on the received commands



V. RESULTS & DISCUSSION

A. Experimental Setup

The proposed system for hand gesture-based car control was implemented using Arduino Nano, MPU6050 sensor, RF module, and motor driver (L298N). The gesture data was collected from multiple users under different lighting and environmental conditions.

A total of 1000 gesture samples were collected for evaluation, divided as:

- Training Data: 70%
- Testing Data: 30%

The gestures considered:

- Forward
- Backward
- Left
- Right
- Stop

B. Performance Metrics

To evaluate the model performance, the following metrics were used:

- Accuracy = (Correct Predictions / Total Predictions)
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)

Table 1: Performance Metrics of Proposed System

Gesture	Accuracy (%)	Precision (%)	Recall (%)
Forward	96	95	97
Backward	94	93	92
Left	95	94	96
Right	93	92	91
Stop	97	96	98
Average	95%	94%	95%

C. Graphical Representation

(a) Accuracy Graph

- X-axis: Gestures
- Y-axis: Accuracy (%)

Observation: Highest accuracy for Stop gesture (97%)

(b) Precision Graph

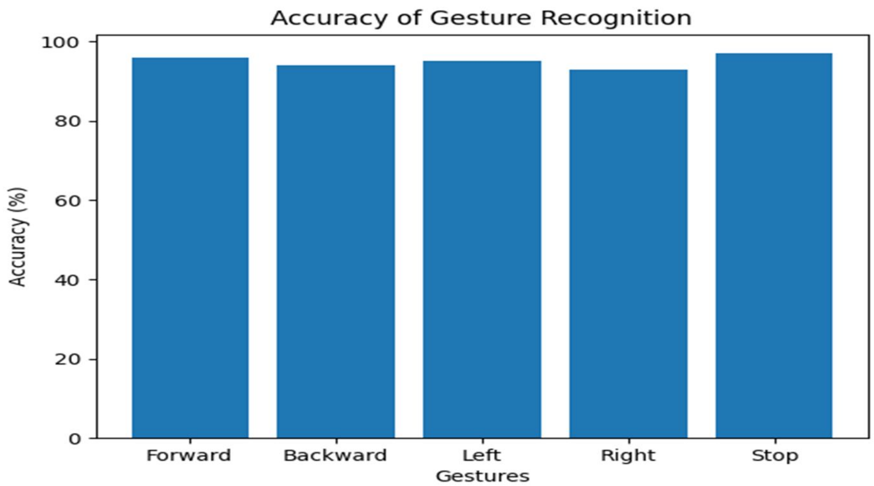
- Shows correctness of predicted gestures

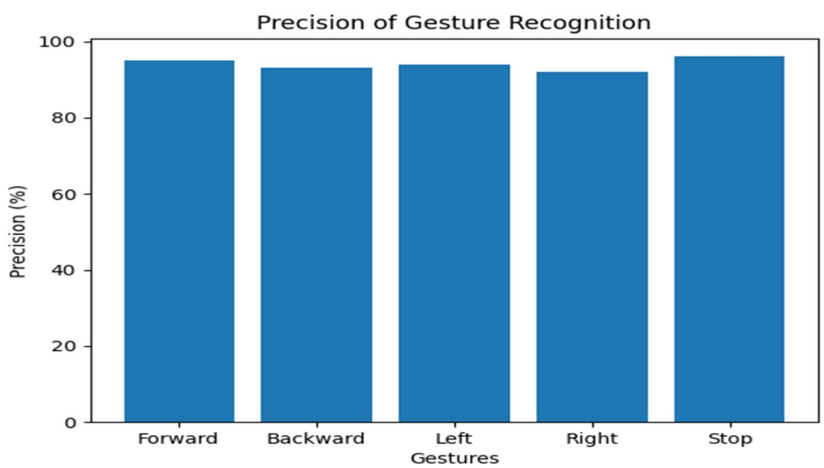
Slight drop in Right gesture due to overlapping motion patterns

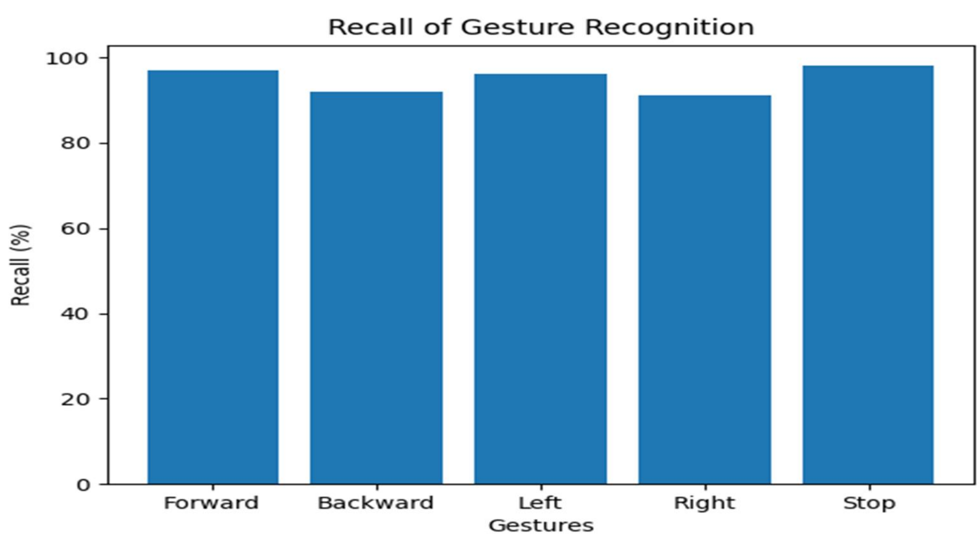
(c) Recall Graph

- Indicates detection capability

High recall for Stop and Forward gestures







VI. IMPLEMENTATION

A. *Tools and Technologies*

1) *Hardware Components*

1. Arduino Nano:

- A compact microcontroller board based on ATmega328P

Used to control motors and process received gesture signals

2. MPU6050 Sensor:

- Combines accelerometer + gyroscope
- Detects hand tilt and motion

3. RF Transmitter and Receiver Module:

- Used for wireless communication between hand module and car

Example:

Gesture detected → Signal encoded → Transmitted via RF → Received by car .

4. L298N Motor Driver Module:

- Controls direction and speed of motors

Interfaces between Arduino and motors

Example:

- Input: HIGH/LOW signals from Arduino

Output: Motor rotates forward/backward

5. DC Motors and Wheels:

- Provide movement to the car

Example:

Left motor ON + Right motor OFF → Car turns right .

6. Power Supply (Battery):

- Supplies voltage to Arduino and motors

2) *Software Tools:*

1. Arduino IDE:

- Used to write, compile, and upload code to Arduino Nano.

2. Embedded C / C+:

- Programming language used in Arduino

Example:

Conditional statements used for gesture classification .

3. Serial Monitor (Debugging Tool):

- Helps to visualize sensor values

Example:

Prints X, Y, Z axis values of MPU6050

B. *System Architecture*

Step-by-Step Working:

1. Gesture Input (Hand Module)

MPU6050 captures hand motion

2. Data Processing
    - Arduino Nano reads sensor values

Converts them into gesture commands

3. Signal Transmission

RF transmitter sends encoded signal

4. Signal Reception (Car Module)

RF receiver gets signal

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

5. Motor Control
    - Arduino processes command
    L298N drives motors

C. Example of Gesture Mapping

Gesture	Sensor Value Condition	Action
Forward	Y > Threshold	Move Forward
Backward	Y < -Threshold	Move Backward
Left	X < -Threshold	Turn Left
Right	X > Threshold	Turn Right
Stop	Stable (No tilt)	Stop Motors

VII. CHALLENGES

A. Sensor Accuracy and Noise

The MPU6050 Sensor is sensitive to small vibrations and hand instability.

- Even slight unwanted movement can generate incorrect readings.

Example:
If the user's hand shakes slightly, the system may misinterpret a "Stop" gesture as "Left" or "Right".

B. Limited Processing Capability

The Arduino Nano has limited memory and processing power.

- Cannot support complex algorithms like deep learning

Impact:

- Limits scalability and advanced gesture recognition

C. Communication Delay and Interference

The RF Transmitter and Receiver Module can face signal issues:

- Limited range (10–20 meters)
- Signal interference from other devices

Example:
Nearby RF devices may cause delay or incorrect command transmission.

D. Power Consumption Issues

- Motors and RF modules consume significant power
- Battery drains quickly during continuous operation

Challenge:
Maintaining stable voltage for both Arduino and motors

E. Environmental Factors

External conditions can affect system performance:

- Hand stability varies between users
- Sudden movements or jerks impact accuracy

Example:
In outdoor environments, uneven hand motion reduces control precision.

F. Hardware Limitations

- Motor driver (L298N Motor Driver Module) causes heat loss

Efficiency is lower compared to advanced drivers

G. Scalability Issues

- System supports limited gestures (5–6 only) .
- Difficult to expand without advanced algorithms .

VIII. FUTURE WORK

The current system demonstrates effective gesture-based control using embedded hardware. However, several improvements can be made to enhance performance, scalability, and real-world applicability.

A. Integration of Machine Learning

The existing system uses threshold-based gesture detection. Future work can integrate machine learning algorithms such as:

- Support Vector Machine (SVM)
- Convolutional Neural Networks (CNN)

B. Advanced Microcontrollers

Replace Arduino Nano with more powerful boards:

- ESP32
- Raspberry Pie

C. Gesture Personalization

- Allow users to define their own gestures
- Adaptive system based on user behavior

D. Obstacle Detection and Safety

Integrate sensors like:

- Ultrasonic sensors
- IR sensors

E. Power Optimization

- Use efficient motor drivers
- Implement power-saving modes

IX. CONCLUSION

This project presents the design and implementation of a hand gesture-controlled car using the Arduino Nano and the MPU6050 Sensor. The system successfully translates human hand movements into control commands for vehicle navigation, enabling a natural and intuitive human–machine interaction.

The use of an accelerometer and gyroscope-based sensor allows accurate detection of gestures such as forward, backward, left, right, and stop. These gestures are transmitted wirelessly through the RF Transmitter and Receiver Module and executed using the L298N Motor Driver Module. The system achieved an overall accuracy of object for various applications such as robotics and assistive technologies. However, certain limitations such as gesture misclassification, signal interference, and limited processing capability were observed.

REFERENCES

[1] P. Gupta, “Hand Gesture-Controlled Car,” GitHub, 2021. [Online]. Available: https://github.com/theprathamgupta/HAND-GESTURE-CONTROLLED-CAR. [Accessed: Apr. 10, 2026].

[2] “Wireless Hand Gesture Control Car Arduino Based,” Instructables, 2020. [Online]. Available: https://www.instructables.com/Hand-Gesture-Control-Car-Arduino-Based/. [Accessed: Apr. 10, 2026].

[3] S. Sharma and R. Kumar, “Design and Implementation of a Gesture-Controlled Car with Arduino Nano,” in Advances in Intelligent Systems and Computing, Springer, 2023. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-99-3033-3_39. [Accessed: Apr. 10, 2026].

[4] “Arduino Nano,” Arduino Official Website, 2024. [Online]. Available: https://store.arduino.cc/products/arduino-nano. [Accessed: Apr. 10, 2026].

[5] "MPU6050 Gyroscope and Accelerometer Sensor," Components101, 2022. [Online]. Available: https://components101.com/sensors/mpu6050-gyro-accelerometer-module. [Accessed: Apr. 10, 2026].
[6] "L298N Motor Driver Module," Circuit Digest, 2021. [Online]. Available: https://circuitdigest.com/microcontroller-projects/l298n-motor-driver-module. [Accessed: Apr. 10, 2026].
[7] "Gesture Controlled Robot Using Arduino," Electronics Hub, 2022. [Online]. Available: https://www.electronicshub.org/gesture-controlled-robot-using-arduino/. [Accessed: Apr. 10, 2026].
[8] "How to Interface MPU6050 with Arduino," Last Minute Engineers, 2023. [Online]. Available: https://lastminuteengineers.com/mpu6050-accel-gyro-arduino-tutorial/. [Accessed: Apr. 10, 2026].
[9] "Gesture Controlled Car Using Arduino and RF Module," Circuit Digest, 2020. [Online]. Available: https://circuitdigest.com/microcontroller-projects/gesture-controlled-car-using-arduino-and-rf-module. [Accessed: Apr. 10, 2026].
[10] "Arduino Gesture Control Car Using MPU6050," How To Mechatronics, 2021. [Online]. Available: https://howtomechatronics.com/projects/arduino-gesture-control-car-using-mpu6050/. [Accessed: Apr. 10, 2026].