Design and Implementation of a Voice-Controlled Vehicle with Arduino

Project Report Submitted to

Department of Physics, School of Science and Humanities,

Shiv Nadar University Chennai

In partial fulfilment of the requirements for the award of the

Degree of **Bachelor of Technology in Artificial Intelligence and Data Science**

BY

Dheeran Sankaran (24011101027)

Johan Emmanuel (24011101043)

Joshua Edwin (24011101044)

K. Aditya (24011101045)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF ENGINEERING

SHIV NADAR UNIVERSITY CHENNAI

NOVEMBER – 2024

SELF ATTESTATION

This is certifying that, we have personally worked on the project entitled **“Design and Implementation of a Voice-Controlled Vehicle with Arduino**”. A case study and data mentioned in this report was obtained during genuine work done and collected by us.

Any other data and information in this report, which has been collected from outside agency, has been duly acknowledged.

STUDENTS

Dheeran Sankaran (24011101027)

Johan Emmanuel (24011101043)

Joshua Edwin (24011101044)

K. Aditya (24011101045)

Date: 25/11/2024

Place: Shiv Nadar University Chennai

TABLE OF CONTENT

FIGURE OF CONTENT

Symbol Explanation

$f(x)$: The periodic function being represented

a_n: The Fourier coefficient for the cosine term

b_n: The Fourier coefficient for the sine term

n: The harmonic index

π: Constant equal to 3.1415

L: Period of the function

$X(t,f)$: The result of the Short Time Fourier Transform

$x[n]$: The input discrete-time signal

$w[n-t]$: The windowing function applied to the signal

t: The time variable

f: The frequency variable

n: The discrete time index

j: Unit for Imaginary Numbers

$|X(t,f)|$: The magnitude of the STFT

$Re(X)$: The real part of the result

$Im(X)$: The imaginary part of the result

1. Objective

The objective of this project is to design and implement a voice-controlled car using an Arduino Uno. The car responds to voice commands such as "go," "left," "right," "stop," and others, which are processed by a trained deep learning model.

2. Components Used

- **Hardware:**
    1. **Arduino Uno**: Microcontroller board to control the car's functions.
    2. **HC-05 Bluetooth Module**: For wireless communication between the car and the control system.(version 2)
    3. **Motor Driver (L298N)**: To control the 4 DC motors driving the car.
    4. **4 BO Motors and Wheels**: Provide mobility to the car.
    5. **Power Supply**: Batteries to power the motors and Arduino.
    6. **Chassi**: Provides a base to the car.
    7. **Jumper Wires**: Used for connections
- **Software:**
    1. TensorFlow(2.18.0) for model training and prediction.
    2. Python(3.11) for voice processing and sending Bluetooth commands.
    3. Arduino IDE for programming the Arduino Uno.

3. Formulas

1. Fourier Series:

$$.f(x) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right)$$

2. Short-Time Fourier Transform:

$$. X(t, f) = \sum x[n] \cdot w[n - t] \cdot e^{-j2\pi f n}$$

3.Magnitude of Spectrogram:

. $| X(t, f) | = \sqrt{(Re(X)^2 + Im(X)^2)}$

4. Working Mechanism

The project combines machine learning, signal processing, and hardware integration to achieve real-time voice control. Below are the key components:

4.1 Machine Learning Model

- **Model Purpose**: The model processes real-time audio and classifies it into predefined commands like "left," "right," "go," "stop," etc.
- **Data Preparation**:
  1. A dataset of spoken words (e.g., "yes," "no," "go") is used.
  2. The audio samples are trimmed to one second and are sampled to 16KHz to prepare it for the next step.
  3. Each audio sample is converted into a **spectrogram** using Short-Time Fourier Transform (STFT), which provides a visual representation of frequency components over time.
- **Model Architecture**:
  1. **Convolutional Neural Network (CNN)**: Processes the spectrogram as image-like data.
  2. Layers include convolutional, pooling, and dense layers with softmax output for classification.
- **Prediction Process**:
  1. Real-time audio input is recorded through a microphone.
  2. The audio is pre-processed then converted into a spectrogram.
  3. The spectrogram is fed into the trained model, which outputs a series of probabilities for each command.
  4. The command is mapped to a code for the car.

4.2 Arduino System

- **Bluetooth Communication**:
    1. The HC-05 module receives commands sent via Python on a PC.
    2. The received commands are decoded by the Arduino Uno.
- **Motor Control**:
    1. The motor driver (L298N) receives signals from the Arduino.
    2. The driver controls the speed and direction of the 4 DC motors based on the commands to so the corresponding action.

4.3 Circuit Diagram

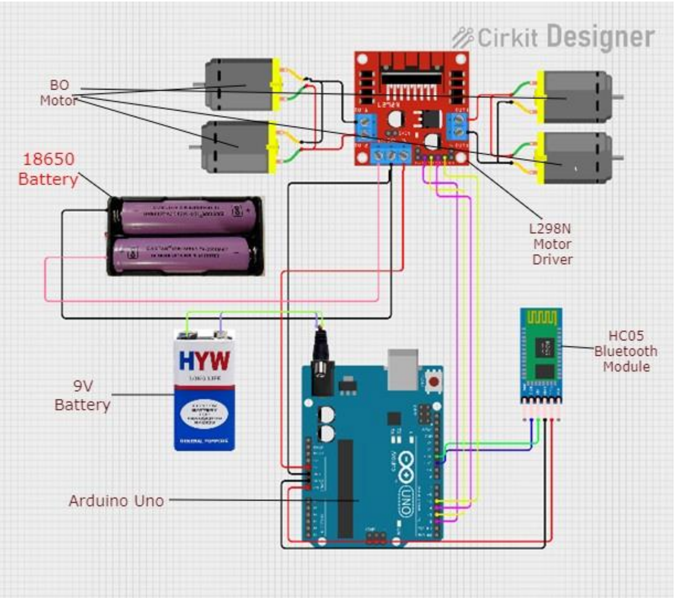

Fig. 1. Circuit Diagram of Setup

5. Code Overview

Python Code:

1. **Recording Audio**:
    - Audio is recorded using the PyAudio library.
    - A silence threshold ensures commands are processed only when spoken.

2. **Model Prediction**:
    - Pre-processed audio is passed to the TensorFlow model for prediction.
    - Command Output are mapped to motor control instructions.
3. **Bluetooth Communication**:
    - Commands are sent via the HC-05 module to the Arduino Uno.

Arduino Code:

4. The Arduino listens for incoming commands via the HC-05 module.
5. Corresponding motor driver signals are sent to control motor speed and direction.

6. Challenges

1. **Noise in Audio Input**: The microphone may pick up background noise, leading to misclassifications.
    - Solution: Implemented a silence threshold and additional data preprocessing.
2. **Real-Time Processing**: Ensuring low latency for real-time control.
    - Solution: Use of more advanced Bluetooth modules to improve range and speed.
3. **Power Management**: Ensuring motors and Arduino are adequately powered.
    - Solution: Used separate power supplies for motors and Arduino.

7. Results

1. The car successfully responds to voice commands like "up", "stop", "left", "right", “no” and "down."
2. The real-time processing pipeline ensures a responsive and interactive user experience.

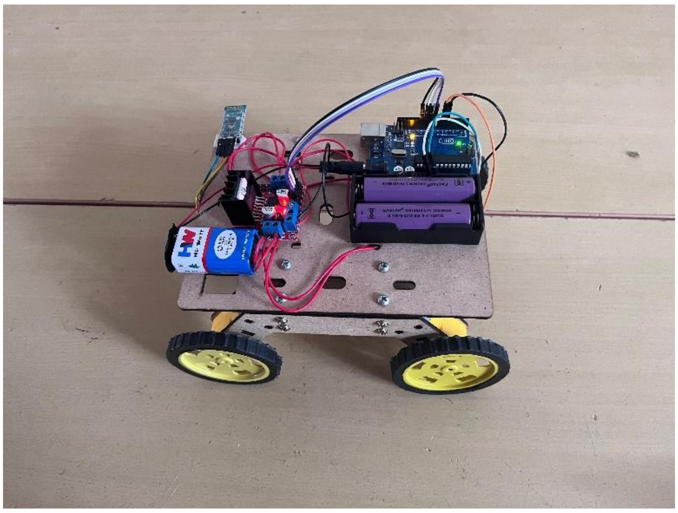

Fig.2. Image of the Assembled setup

8. Future Enhancements

1. **Noise Robustness**: Use advanced preprocessing or a more complex model to handle noisy environments.
2. **Use more advanced Bluetooth**: The use of better Bluetooth modules will allow is to both increase the speed and range of communication.
3. **Obstacle Detection**: Add ultrasonic sensors for obstacle detection and autonomous navigation.
4. **Ease of Use**: Can have microphone be on either the car or operate it from a mobile app for greater convenience.

9. Scope

1. The base of the project can be extended to other voice-controlled robotics applications.

2. The concept can be utilised to conduct humanitarian and military missions in hostile environments.

3. It can be further developed more automated and easy driving.

10. Conclusion

This project demonstrates the integration of machine learning and embedded systems for real-time voice-controlled robotics. The system provides a practical and interactive way to control an Arduino car using voice commands, highlighting the potential for voice-based interfaces in robotics and automation.

11. References

1) Tensorflow - **https://www.tensorflow.org**

2) Arduino - **https://www.arduino.cc**

3) PyAudio - **https://pypi.org/project/PyAudio/**

4) HC-05 Datasheet - **https://components101.com/sites/default/files/component**

5) Pyserial documentation - **https://pyserial.readthedocs.io/en/latest/**