Department of Electrical and Computer Engineering
North South University

Directed Research

Intelligent Robot Vehicle with Real-Time Object Classification and Autonomous Navigation

NAME	ID
JOYANTA BHATTACHARJEE	1821535042

Faculty Advisor:

Md. Shahriar Hussain

Senior Lecturer

ECE Department

Spring, 2023

LETTER OF TRANSMITTAL

February, 2025

To

Dr. Mohammad Abdul Matin,

Chairman,

Department of Electrical and Computer Engineering,

North South University, Dhaka.

Subject: Submission of Directed Research (498R) Final Report on “**Intelligent Robot Vehicle with Real-Time Object Classification and Autonomous Navigation.**”

Dear Sir,

With due respect, I would like to submit my Directed Research (498R) Final Report on “**Intelligent Robot Vehicle with Real-Time Object Classification and Autonomous Navigation.**” as a part of my BSc program. The report deals with **Real-Time Object Classification and Autonomous Navigation**. This project was extremely valuable to me as it provided hands-on experience in both hardware and software fields, allowing me to gain practical expertise in IoT and autonomous systems. I strived to meet all the necessary dimensions required for this report to the best of my abilities.

I will be highly obliged if you kindly receive this report and provide your valuable judgment.

It would be my immense pleasure if you find this report useful and informative to have an apparent perspective on the issue.

Sincerely Yours,

....

Joyanta Bhattacharjee

ECE Department

North South University, Bangladesh

APPROVAL

Joyanta Bhattacharjee (ID # 1821535042) from Electrical and Computer Engineering Department of North South University, have worked on the Directed Research Project titled "**Intelligent Robot Vehicle with Real-Time Object Classification and Autonomous Navigation.**" under the supervision of **Md. Shahriar Hussain** partial fulfillment of the requirement for the degree of Bachelors of Science in Engineering and has been accepted as satisfactory.

Supervisor's Signature

……………………………………

Md. Shahriar Hussain

Senior Lecturer

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

Chairman's Signature

……………………………………

Dr. Mohammad Abdul Matin

Professor & Chair

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

DECLARATION

This is to declare that this project/directed research is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. All project related information will remain confidential and shall not be disclosed without the formal consent of the project supervisor. Relevant previous works presented in this report have been properly acknowledged and cited. The plagiarism policy, as stated by the supervisor, has been maintained.

Students' names & Signatures

Joyanta Bhattacharjee

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

ACKNOWLEDGEMENTS

First and foremost, acclaims and gratitude to God, the Almighty, for His showers of blessings all through my research work to effectively finish the research.

I would like to express my heartfelt gratitude towards this project and research supervisor, Md. Shahriar Hussain, Senior Lecturer, Department of Electrical and Computer Engineering, North South University, Bangladesh, for his invaluable support, precise guidance and advice pertaining to the experiments, research and theoretical studies carried out during the course of the current project and also in the preparation of the current report.

Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly.

ABSTRACT

Intelligent Robot Vehicle with Real-Time Object Classification and Autonomous Navigation

Intelligent robotic systems that can sense and interact with their surroundings are increasingly important in assistive technologies and automation. The project presents a novel IoT-based Intelligent Robot Vehicle that can classify and move towards a target object in line with a verbal instruction. The system is rooted in a tailored Convolutional Neural Network (CNN) on ResNet-50, which is pre-trained on the COCO database, that is used to categorize four pre-specified objects: a bottle, a cup, a book, and a phone. The robot can navigate towards a single target in a field with multiple objects placed in its path. The robot listens to a verbal instruction, takes a picture of the target field, applies its CNN-based classification algorithm, and finds out which target is in line with its instruction. After classification, it finds a optimal path with A* Algorithm and moves towards its target with obstacle detection via an ultrasonic sensor. The central processor is an ESP32 dev board that executes classification output and move instructions via an L298N driver. In a combination of classification with a deep learning approach, a verbal input, as well as autonomous navigation, the project presents a viable methodology in smart robotics with a prospect towards IoT-based automation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

1.1 Background and Motivation

With the advent of automation and smart systems, autonomous robots are increasingly important in assistive technologies as well as industrial sectors. The biggest robotics challenge is making machines not just understand objects correctly but also move towards them on its own, particularly in real-world scenarios with multiple objects. The methods traditionally followed are not highly adaptive, which is why solutions with deep learning are becoming vital in enhancing object detection as well as making decisions [1].

This project is a robot car that is IoT-based with a target sense with a verbal command. The system detects a bottle, a cup, a book, and a phone with a ResNet-50-based CNN that is specifically trained on a COCO dataset. The robot takes a snapshot as soon as a verbal command is transmitted, sends it through a CNN model, and detects which route is optimal with A* Algorithm. The robot is equipped with a dev module from ESP32 with an L298N driver with ultrasonic sensor-based detection.

The motivation behind implementing this project is its application in assistive robotics, smart automation, as well as human robot interface [2]. The combination of autonomous navigation, voice recognition, and deep learning in this system makes robot intellect more efficient in executing activities in real-world scenarios.

1.2 Purpose and Goal of the Project

The primary goal of this project is to design an IoT-based Intelligent Robot Vehicle that can categorize objects and move towards a target object on its own with a voice command. The project brings about a combination of autonomous navigation with deep learning-based object classification in order to facilitate a more interactive robot with its surroundings [3].

The key activity in this project is implementing a specialized ResNet-50-based CNN model for classification in real time. The robot is trained on a COCO database in a way that it is able to categorize four pre-specified objects - a bottle, a cup, a book, and a phone. The robot captures a snapshot on a verbal prompt, sends it through a model, and locates its optimal route with A* Algorithm towards a target with an ultrasonic sensor [4].

A novel feature is seamless voice recognition, classification with deep learning, and autonomous mobility in an IoT platform [5]. The application is developed with the ESP32 dev board and driver L298N, with which advanced activities like detection of objects and navigation in a live situation can be achieved. The overall vision is making IoT-based robotics a possibility in operational fields like assistive devices, autonomous delivery, and human robot co-working [6].

1.3 Organization of the Report

This report is structured to provide a clear understanding of the project's objectives, methodology, and results. The organization of the report is as follows:

- **Introduction**
  This chapter is on the background, motivation, purpose and goal of the project. The chapter is a generic discussion regarding IoT-based Intelligent Robot Vehicle, its subsystems, and why the project is crucial in autonomous robotics as well as assistive technologies.
- **Literature Review**
  This chapter presents available research and technologies in robotics in IoT, autonomous navigation, and object classification. The chapter also presents application in models in deep learning that is ResNet-50 in image classification as well as application in autonomous systems via A* Algorithm in route planning.
- **Methodology**
  This section explains robot design in both hardware (ESP32, ultrasonic sensor, driver L298N), as well as software design. The design of ResNet-50-based CNN is also detailed, in addition with voice-based modules as well as global system flow both in terms of classification of objects as well as navigation.
- **Experimental Setup and Implementation**
  This chapter also outlines experimental configuration, training procedure on dataset

COCO in case of ResNet-50, module incorporation in terms of both software as well as hardware, as well as procedure in terms of autonomous travel as well as classification in real-time.

- **Results and Discussion**
  This section is a demonstration of experimental results, i.e., robot as well as model performances in terms of accuracy as well as robot performances in approaching target object. The results are interpreted, as also challenges met or limitations experienced in its application.
- **Conclusion and Future Work**
  The final section concludes with a brief overview of its key outputs, its impact, as well as with proposals on expansion in the future, i.e., extension on its classification feature on objects, enhancing its navigation, or addition with more sensors.

Chapter 2

Research Literature Review

2.1 Existing Research and Limitations

Recent advancements in autonomous robot applications in object recognition have led to various trends in deep learning in these aspects. Many researchers have researched various ways to obtain better efficiency and performance in these aspects. As an example, an object recognition system was created by Zhang et al. [1] through deep learning by restructuring YOLOv4 in autonomous robot recognition in real-world scenarios. Researchers have carried out testing in COCO dataset with an mAP value of 43.5% to validate better performance in recognition in real-world scenarios. Wang et al. [7] used transfer methods in indoor object recognition by integrating transfer methods with MobileNetV2 to achieve an accuracy value of 91.3% in Indoor Scene Recognition dataset.

In subsequent work, an independent system was developed by Li et al. [8] that utilizes Dijkstra's algorithm to achieve optimum pathfinding in crowded environments. This system was computer programmed and implemented in robot systems to achieve an effectiveness rating of 87% in navigating dynamic crowded environments. Additionally, in work by Kim et al. [9], an investigation was carried out in combining LiDAR technology with an RGB-D sensor in robot intelligence to achieve better spatial awareness while optimizing performance in object classification.

Despite these trends, boundaries remain to apply to work in modern times. Following key results have been established through an extensive search through the literature:

Insufficient Diversity in Dataset: Numerous projects fall back to conventional datasets, e.g., COCO or ImageNet, that fail to reflect indoor-related intricacies and detect entities in miniature sizes.

Comparative Evaluation: Current modern work is mostly only concerned with comparing independent systems of deep learning in vacuum and not comparative evaluation among systems. This is a hindrance to in-depth knowledge about most efficient systems used in automated object classification in real-world problems.

Deployment of Embedded Systems: Though most computer vision projects perform impressively in providing appreciable accurateness in solving problems in object recognition and navigating through problems, fewer have been deployed in working environments in practice. Embedded systems' in-built computational limitations in boards like Jetson Nano or ESP32 pose certain limitations to working efficiently in these deep learning models.

The limitations in this paper call for embracing an extensive methodology that utilizes diverse datasets to test diverse advanced algorithms in deep learning and optimize the use of embedded systems in order to realize practice effectiveness. This paper addresses these limitations by designing an autonomous robot to utilize YOLOv7 in object recognition, utilize A* algorithm in path finding, and employ ESP32 in embedded management in order to realize efficient path tracking and classification in dynamic environments.

Chapter 3

Methodology

3.1 Workflow

The intelligent robot car system initially processes an audio command by the user through an algorithm recognizing verbal statements to translate verbal statements to messages in text. On recognizing the command, an active camera is powered to capture live frames around the scene. The frames get processed by a custom trained ResNet-50 to classify objects to detect and confirm presence of target objects instructed by the user. On valid classification, an output is generated to confirm the object and proceeds to path planning.

Using a path-finding algorithm such as A* algorithm, the system calculates an efficient path to sensed objects considering factors in the environment. Motion-control commands are subsequently executed to maneuver the robot in an efficient manner. Such commands are sent to the motor driver, driving the motion of robot wheels to gain smooth motion. Additionally, an ultrasonic sensor continuously monitors surroundings to detect motion obstruction. As such, the system is in an ability to employ methods to avoid collisions. If an obstruction is detected, the system dynamically adjusts path to avoid collisions in order to gain secure efficient motion. This systematic process covers real-time object recognition, autonomous guidance, and adjustable obstruction avoidance making robot vehicle to be efficient in dynamic environments.

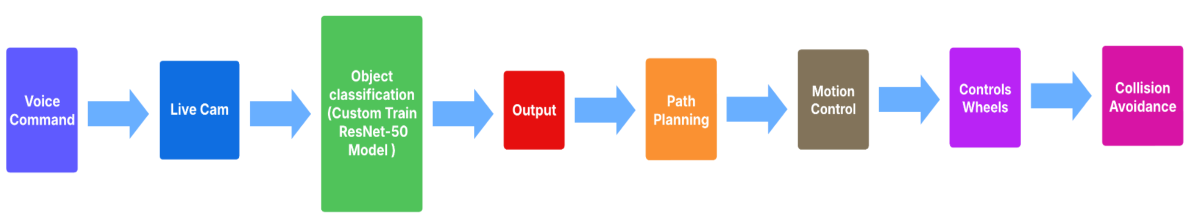


Figure 1. Overall Workflow

3.2 System Design

The intelligent robot has both hardware and software in synchronization. Software is used to take in through an utterance library to gain speech recognition to convert speech to text. This is employed to execute the classification model, an in-house trained ResNet-50 object recognition model. The classification model is employed to respond by giving feedback through pyttsx3 library by converting text to speech.

For real-time guidance, live video is streamed by an ESP32 cam to a web server. Processed information is later employed by an ESP32 development board to power motion. Obstacles and impacts in between are sensed by an HC-SR04 ultrasonic sensor employed by the board. Additionally, an L298N motor drive is employed to power gear motors to aid in smooth motion. All these units operate efficiently in this system to lead the robot to move in an area by user command.

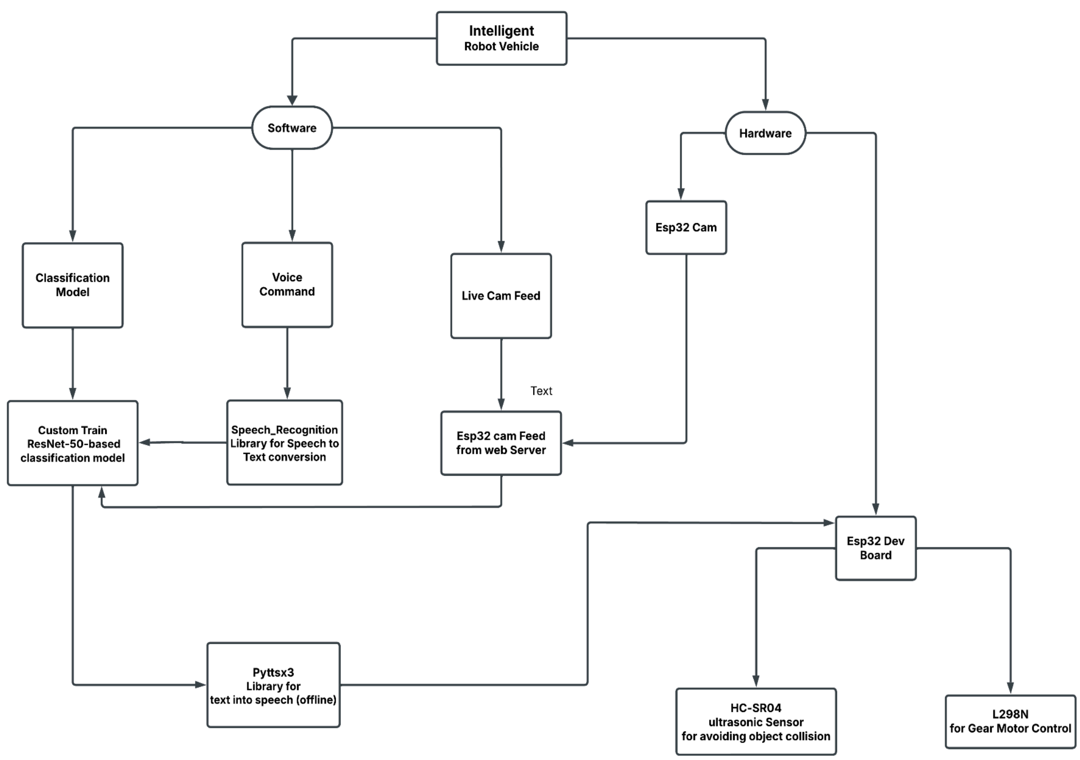


Figure 2. Initial design of the proposed solution

3.3 Tools (Software & Hardware)

3.3.1 Software Tools

This project is a combination of Software and Hardware. Below is a list of all the software tools I have used for this project :

TABLE 1. SOFTWARE TOOLS TABLE

Tool	Functions	Other similar Tools (if any)	Why selected this tool
Python	All the hardware, software and ML code is written in Python Language.	R, Swift, Matlab etc.	Easy to code and Rich ecosystem of libraries.
ResNet-50 (Custom Trained)	For classifying between 4 objects (Bottle, Cup, Book, Phone)	ResNet-18, ResNet-34, ResNet-101	Efficient architecture, balances accuracy and computational efficiency, which make it suitable for real-time object classification with fewer parameters.
OpenCV	Image Processing and Computer Vision Task	Pillow, scikit-image	Robust and optimized tools for real-time image processing and object detection
NumPy	For mathematical operations and data manipulation	SciPy	Fast numerical computations, efficient array handling and matrix operations
Speech_Recognition Library	Convert User Voice to Text	DeepSpeech	Simple API and supports multiple language
Pyttsx3	Convert Text to Speech for Voice Feedback	gTTS (Online), Festival TTS (offline)	offline and support multiple voices and

			customizable speech rate
Arduino IDE	For coding Esp32 Dev board and Esp32 Cam	PlatformIO, VS Code with Arduino Extension	Easy to use and have a vast community support
ESP32 Libraries (MicroPython)	Controls the Esp32 Dev board for hardware Interaction	n/a	To control L298N and handle the requests from ML
Requests Library	Sends and receive requests from the web server for the movement of Gear Motor	http.client, aiohttp	Simple and efficient in handling the HTTP requests and API communications
A* Algorithm	For path planning towards the classified Object	Dijkstra algorithm	Efficient in finding the optimal path
Jupyter Notebook	1. Loading the ResNet-50 Model for object classification 2. Loading the Esp32 cam feed for live video 3. Working as a brain of the project 4. Getting the Voice Command 5. Working as a bridge between software and hardware 6. Sending commands to Esp32 to control the L298N motor driver for controlling the Gear Motors 7. Showing output	Google Colab, Kaggle Kernels, and Microsoft Azure Notebooks	For its local environment setup
Google Colab	For the training of the ResNet-50 Model	Kaggle Kernels, Jupyter Notebooks, and Microsoft Azure Notebooks	For it’s free access to GPU/TPU, that’s why loads very fast and takes less time for training

3.3.2 Hardware Components

3.3.2.1 Esp32 Development Board

For my project I used the Esp32 Dev Board for building connections with my software and also used it to control the L298N motor Driver. The ESP32 is a low-cost, low-power system on a chip series of microcontrollers with Wi-Fi and Bluetooth capabilities and a highly integrated structure powered by a dual-core Tensilica Xtensa LX6 microprocessor [10] .

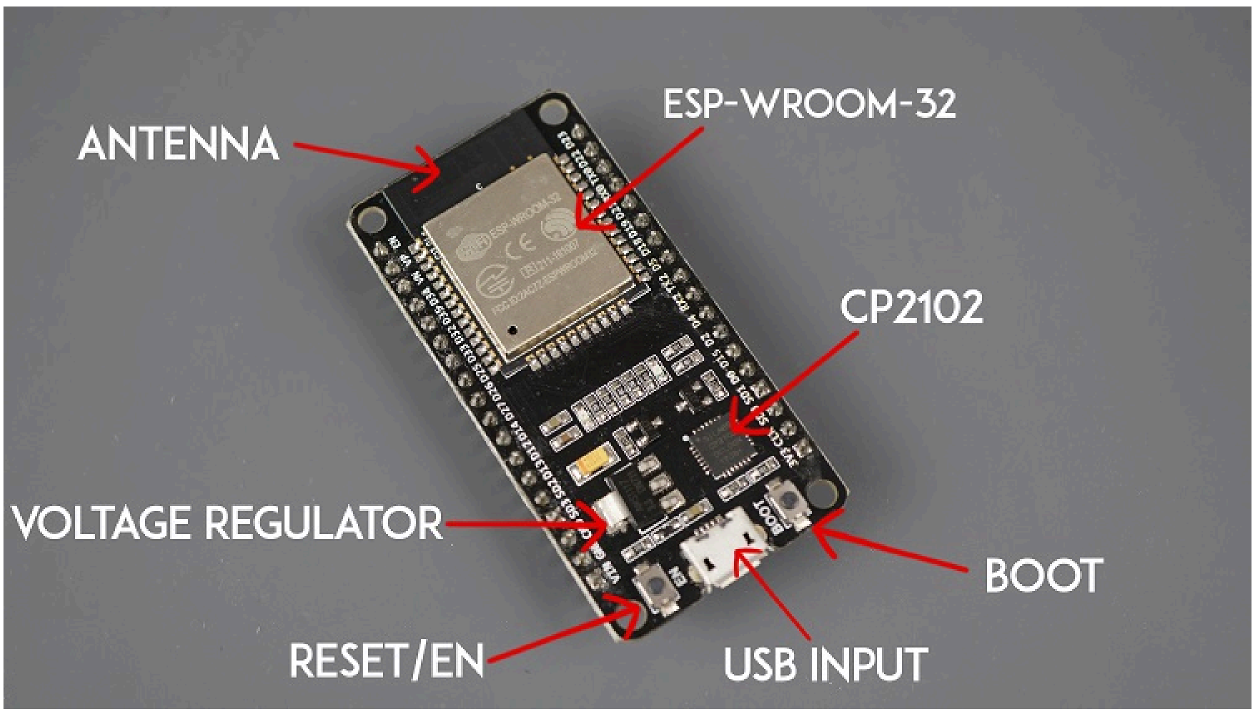


Figure 3. Esp32 Development Board

3.3.2.2 Esp32-CAM

I used an Esp32 cam for the live feed to classify the objects. Esp32-CAM is a low-cost microcontroller which comes with an integrated camera, Wi-Fi and Bluetooth connectivity which makes it efficient for computer vision and IoT devices. It has an OV2640 camera sensor which enables real-time image processing and wireless transmission [11].

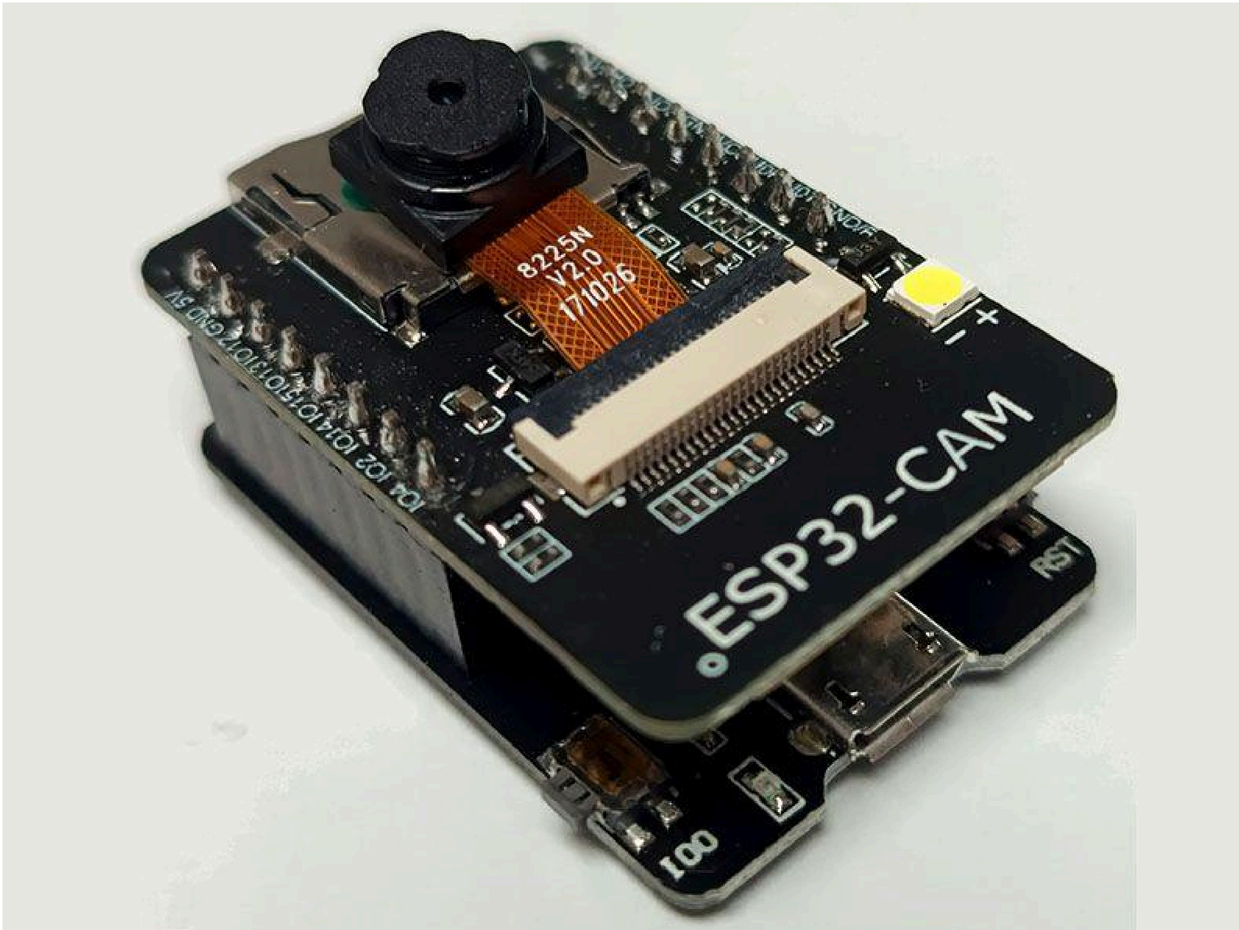


Figure 4. Esp32-CAM

3.3.2.3 L298N DC Motor Driver

L298N is a dual H-Bridge motor driver that can control DC motors and stepper motors using microcontrollers like Ardunio and Esp32. It supports bi-directional motor control with independent speed regulation via PWM signals. With a built-in 5V regulator, it can power low-power microcontrollers [12]. With the help of L298N I controlled my 2 DC motor for vehicle movement.

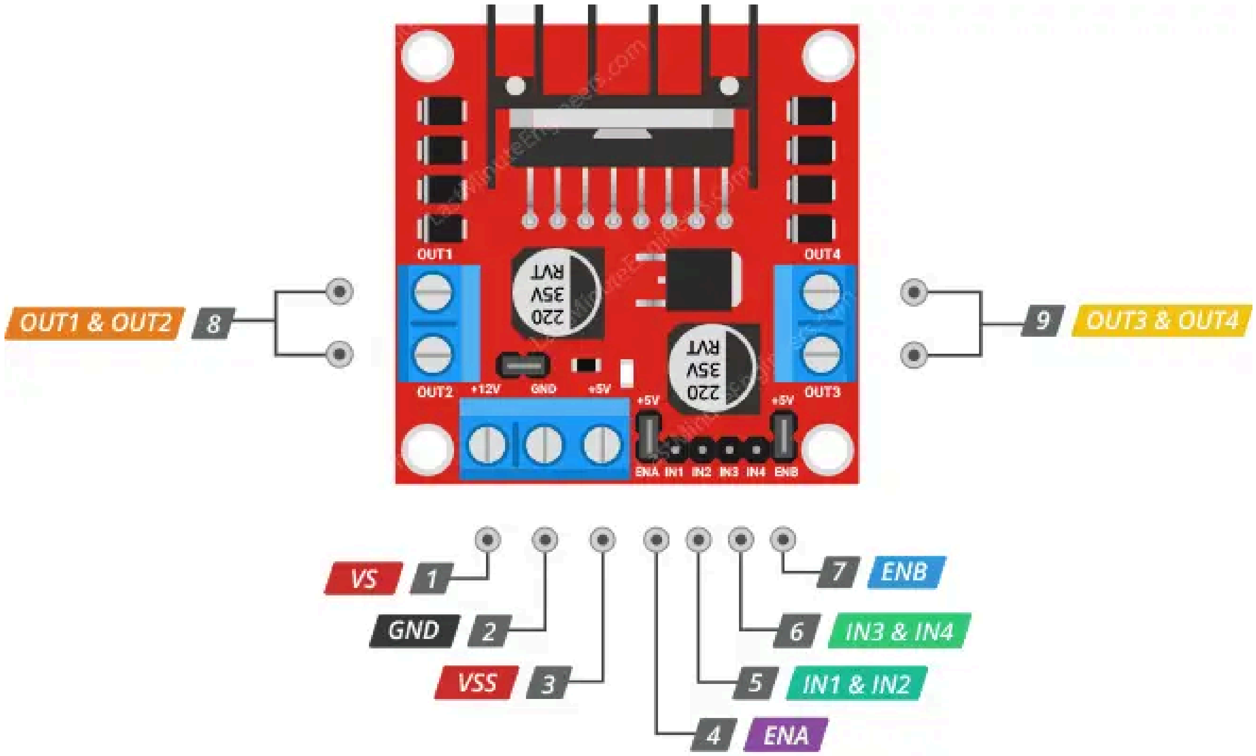


Figure 5. L298N DC Motor Driver

3.3.2.4 HC-SR04 Ultrasonic Sensor

HC-SR04 is an ultrasonic sensor which is mainly used for its precise measurement by emitting high-frequency sound waves and detecting their reflection. It can measure from 2cm to 400cm [13]. I also used this to avoid obstacle collisions.

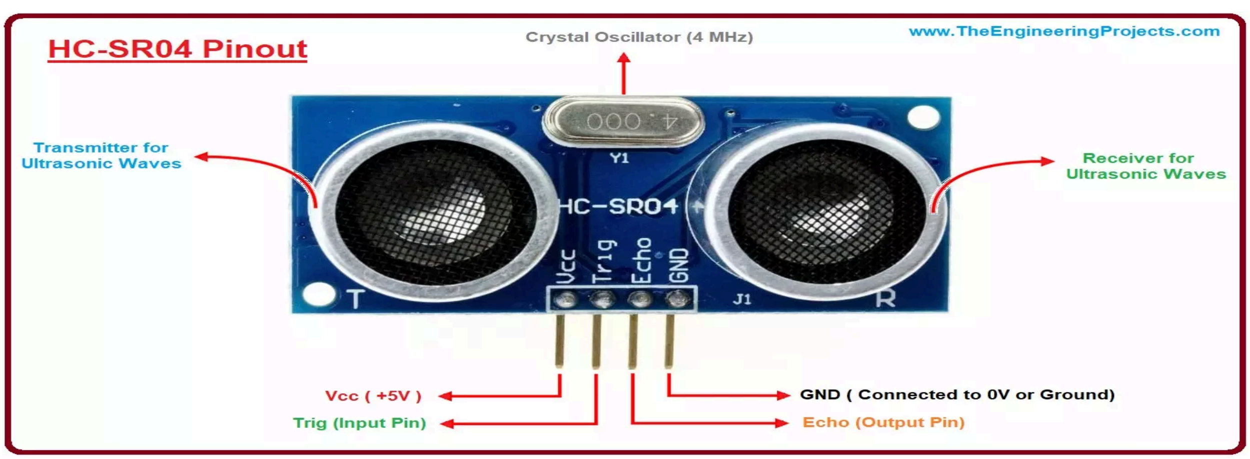


Figure 6. HC-SR04 Ultrasonic Sensor

3.3.2.5 DC Gear Motors & Wheels

For my project I used 2 gear motors and 2 Rubber wheels and 1 Caster Wheels with 360 Degree Moving Brack.

Figure 7. DC Gear Motor

Figure 8. Rubber Wheel

Figure 9. 360 Caster Wheel

3.3.2.6 Rest Hardware Components

In the project-

1. To manually turn on and off the vehicle I used a switch.
2. I used some male to female, female to female and male to male jumper wires for building connections between all the hardware components.
3. I used a power bank to power the Esp32 Dev board and Esp32-CAM.
4. I used 3 3.7V Rec-B Battery for powering the L298N.
5. For the body I used PVC board
6. Used Double Sided tape, normal tape and Super Glue to add the components.
7. I used 1 2S and 1 1S battery holders and built the connection between them and made a 3S battery holder for my project.
8. I used 1 Breadboard to build the connection with my Esp32 and HC-SR04.

3.4 Model Training For Object Classifying

To enable my IoT-based smart Robot Vehicle to detect and move towards a target object, I have trained a Convolutional Neural Network (CNN) upon ResNet-50. The network is pre-trained upon the database of COCO and later trained upon a custom database for four given objects as a cup, book, phone, and a bottle.

3.4.1 Dataset and Preprocessing

My dataset consisted of:

- **Cell Phones:** 4,087 training images, 176 test images
- **Bottles:** 7,725 training images, 355 test images
- **Cups:** 6,083 training images, 246 test images
- **Books:** 3,754 training images, 171 test images

```
Train images for cell phone: 4087
Test images for cell phone: 176
Train images for bottle: 7725
Test images for bottle: 355
Train images for cup: 6083
Test images for cup: 246
Train images for book: 3754
Test images for book: 171
```

Figure 10. Custom Dataset

The images are enriched and enriched in rotation form, scaling, and normalization for improved generalization in the model. The training and testing are divided in a ratio of 80-20.

3.4.2 Training Process

I fine-tuned network ResNet-50 at a learning rate of 0.001 and Adam optimizer and loss based on cross-entropy. Training lasted for 20 epochs and performance metrics are recorded.

3.5 Hardware setup

First I built the body of my vehicle using the PVC board. Then using the Arduino IDE I uploaded the code to my Esp32-CAM and Esp32 Dev Board. Then I used female to female jumper wires to build connection with my L298N and Esp32 Dev board. I used 2 gear motor for my project. So I placed one gear motor with the output 1 of the L298N and placed another one with output 2 of the L298N. Then to power the L298N I used 3 3.7V battery and placed the positive with a switch and used another wire with the switch to build the connection. I used a switch so that I can manually turn on/off the vehicle. I put the wire into the L298N VS (+12V) port and put the negative wire into the ground port. From the ground port I used another male to female wire to build the connection with the Esp32 Dev board. I used the female part with the Esp32 Dev board ground pin. To control the both motor speed I used 2 female to female wires with the ENA and ENB and put it in the Esp32 Dev pins. Then to control the gear to go forward left right etc. I placed the wires into IN1, IN2, IN3 and IN4 and put it in the Esp32 Dev pins. I left the VSS of L298N because I'm powering my Esp32 Dev with the power bank. With the gear motors I used 2 rubber wheels and in the front I used the 3rd wheel which is a Caster Wheel with 360 Degree Moving Brack. To build the connection with the HC-SR04 ultrasonic sensor I used a bread board and some male to female jumper wires. I placed the HC-SR04 so that it can detect obstacles in front of the vehicle. Then I used a small part of the PVC board and used glue to add the Esp32-CAM in front of the Vehicle for the object classifying purpose. I used a power bank and 2 USB cables to power the Esp32-CAM and Esp32 Dev board.

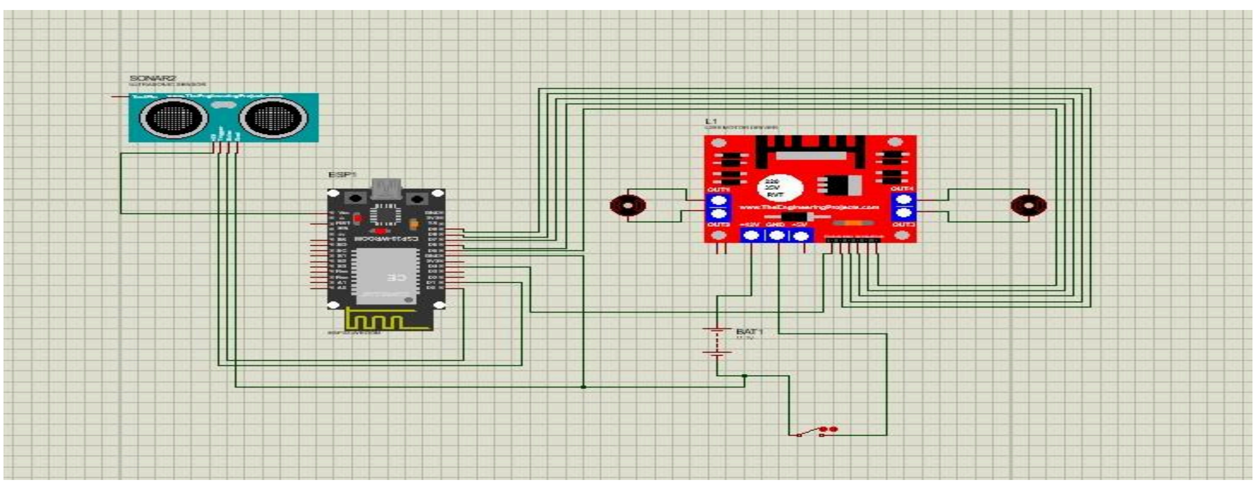


Figure 11. Connection diagram using Proteus

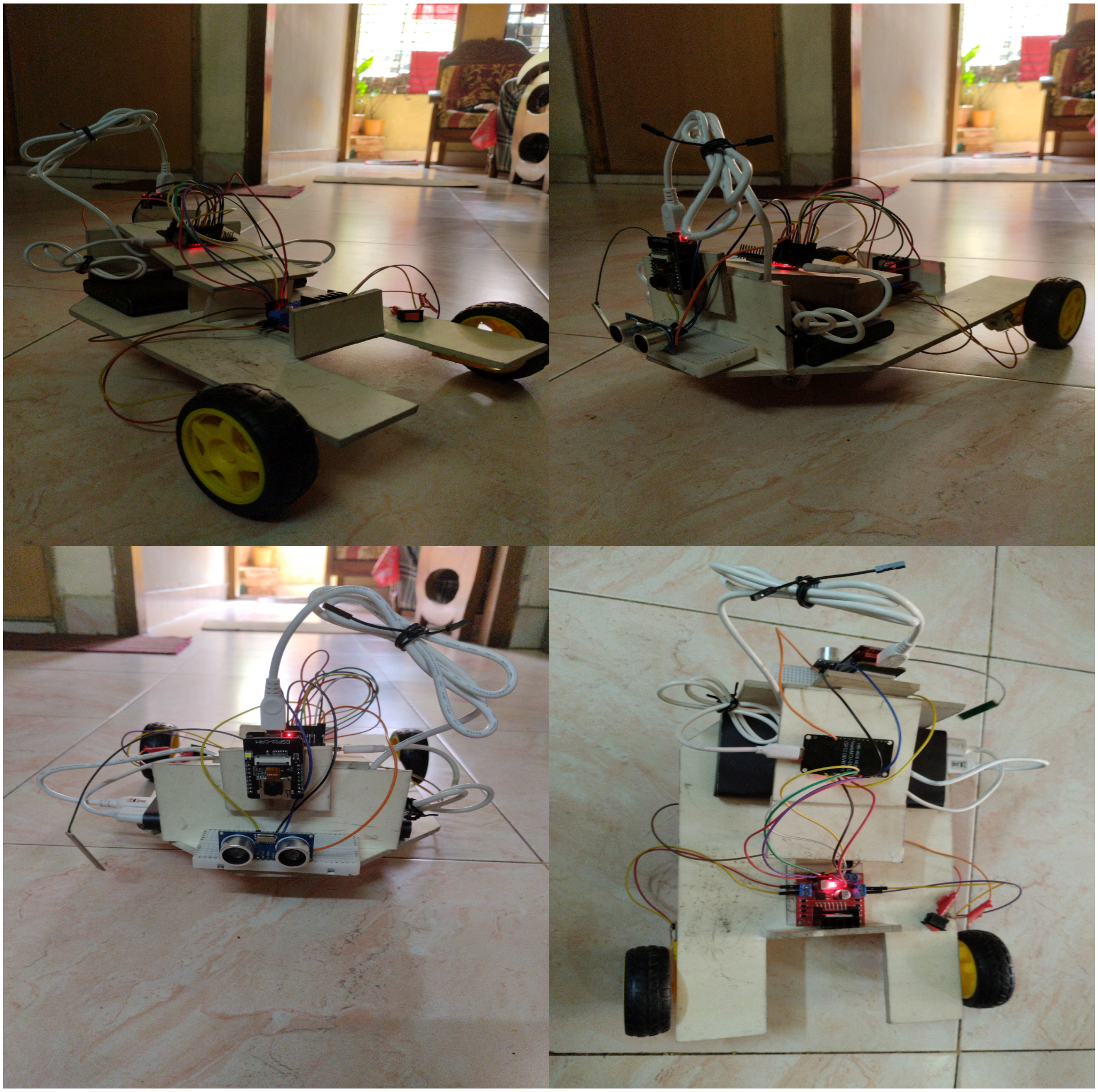


Figure 12. Hardware Setup

Chapter 4

Results

4.1 Model Test Result

4.1.1 Graph 1 - Training Loss & Accuracy Over Epochs

- **Loss Reduction:** The training loss commenced at beyond 1.2 and fell continuously throughout epochs, suggesting convergent behavior. The loss plummeted after about 7–8 epochs and reached a lower level at a relatively steady state after some fluctuation.
- **Accuracy Improvement:** The training accuracy began below 50% but improved rapidly and rose beyond 80% at as early as 5 epochs and rose to above 95% at epochs 7–8. The remaining epochs had good accuracy, indicating good learning in the model.

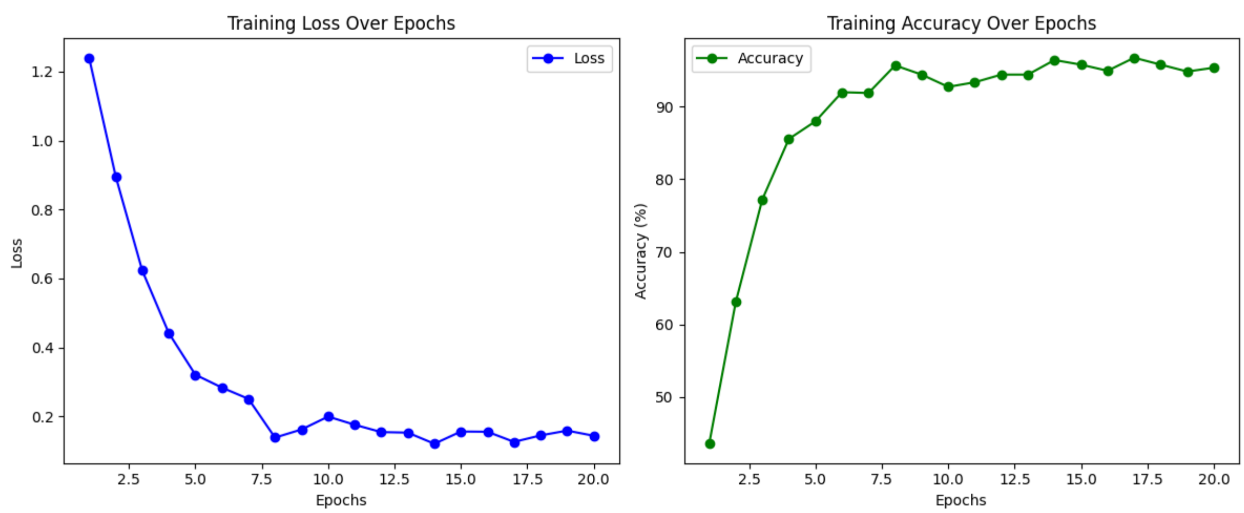


Figure 13. Training Loss & Accuracy Over Epochs

4.1.2 Graph 2 - Test Accuracy Over Batches

- The test accuracy fluctuated over different batches, initially reaching 100%, but then dropped due to batch variations.
- The accuracy level settled and improved in general but some variations suggest generalizable problems.

- The unstable batch performance may be indicative of perhaps the model being slightly overfitting or being confronted with tough samples in certain batches.

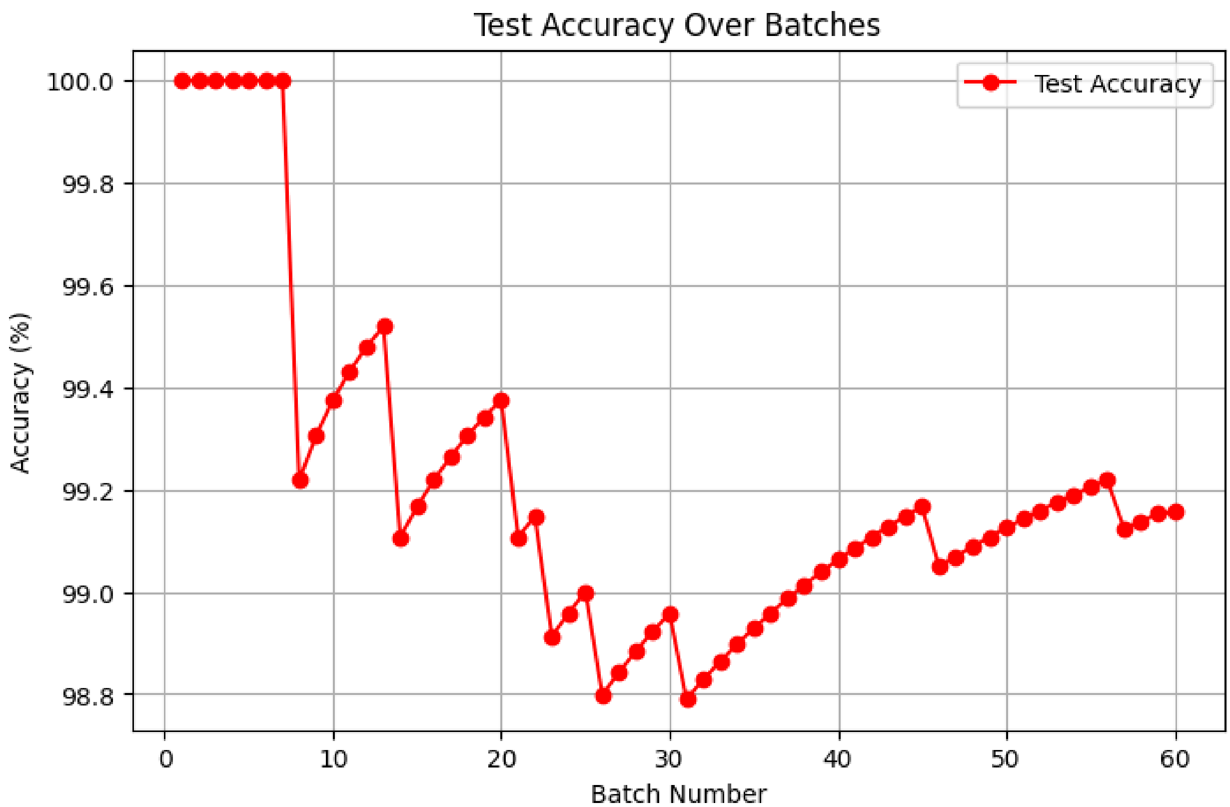


Figure 14. Test Accuracy Over Batches

4.1.3 Model Test Result On Pictures

After fine tuning and training the ResNet-50 model on a custom dataset I run the model on different types of images to check the accuracy and testing on the model. And everytime the model has shown good results.

Figure 15. Prediction using ResNet-50 Model

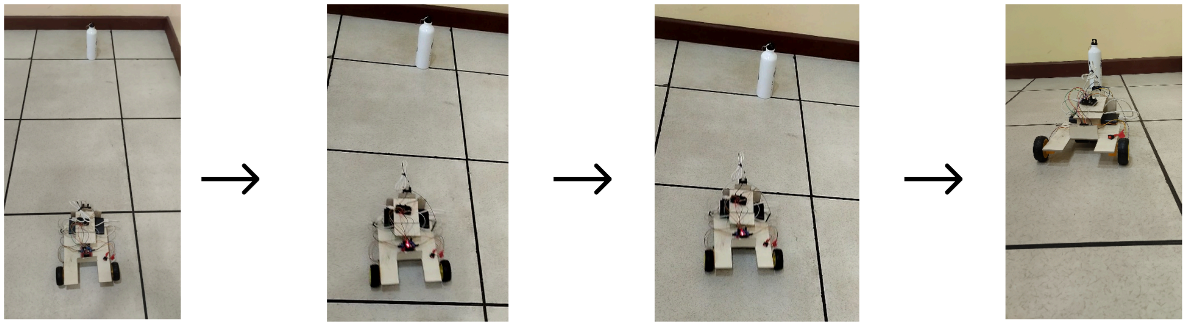
4.2 Live Project Result

In the final part of my project I used a jupyter notebook to locally load the model and use it to classify between 4 objects (Bottle, Cup , Book and Cell Phone). In this part first I load the custom trained ResNet-50 model in my jupyter notebook environment. Then I give a voice command using my headphone and then the Speach_Recognation converts the voice into text. and then I load the url of the Esp32-CAM. Then the model uses the live camera feed of Esp32-CAM to classify between the four objects. When the object is classified the the jupyter notebook code instruct the Esp32 via wifi using the request command to go towards the object using the A* algorithm. And when the vehicle is near the object it uses the HC-SR04 ultrasonic sensor to stop the vehicle to avoid the collision.

```
Listening for command...
You said: find bottle
Path to follow: [(0, 1), (0, 2), (0, 3), (0, 4), (1, 4), (2, 4), (3, 4), (4, 4)]
Moving to: (0, 1)
Command sent successfully. Response: Command received: move_forward
Moving to: (0, 2)
Command sent successfully. Response: Command received: move_forward
Moving to: (0, 3)
Command sent successfully. Response: Command received: move_forward
Moving to: (0, 4)
Command sent successfully. Response: Command received: move_forward
Moving to: (1, 4)
Command sent successfully. Response: Command received: move_forward
Moving to: (2, 4)
Command sent successfully. Response: Command received: move_forward
Moving to: (3, 4)
Command sent successfully. Response: Command received: move_forward
Moving to: (4, 4)
Command sent successfully. Response: Command received: move_forward
Object found: Bottle
Distance to object: 3.28 cm
Object is close enough. Stopping the vehicle.
Command sent successfully. Response: Command received: stop
```

Figure 16. Final Result of the Project

Figure 17. Instructed Object is being found



Initial Stage Second Stage Third Stage Final Stage

Figure 18. Going forward with the A* algorithm path planning towards the object

Chapter 6

Conclusions

8.1 Summary

This project develops an IoT-based smart Robot Vehicle classifying and moving towards a target object as instructed verbally. It applies a CNN on ResNet-50 trained on COCO and classifies four different objects (bottle, cup, book, and phone) and classifies them as instructed. It computes the optimal path based on A* Algorithm and self-navigates avoiding obstacles based on an ultrasonic sensor. An ESP32 dev board receives commands and controls movement based on an L298N motor driver. It applies deep learning, speech recognition, and self-navigating features, thus making it a smart robot and application for automated solutions based on IoT.

8.2 Limitations

1. **Hardware Constraints :** ESP32 computational limitations, L298N inefficiencies and inefficiencies in ultrasonic sensor in dealing with some obstacles.
2. **Real-Time Processing :** ESP32 and ResNet-50 are slow and can't perform speech recognition at some instances.
3. **Environmental Factors :** The lighting or noise in certain surroundings may affect voice commands and classification.

8.3 Future Improvement

1. **Enhanced Object Recognition :** Upgrade to advanced architectures (i.e., MobileNet or ResNet-50) for better and efficient classification.
2. **Improved Path Finding:** Include RRT* or D* Lite for real-time efficient movement handling in moving obstacles [14].
3. **Better Speech Recognition :** Implement noise-robust models and multiple-language capability [15].
4. **Power Efficiency:** Raise energy consumption for a longer time and add solar charging [16].

5. **Edge Computing:** Processing minimized in cloud or at edge for better response time [17].
6. **Reinforcement Learning:** Make the robot improve and adapt behavior based on learning in real-time from the surrounding environment [18].
7. **Multi-Robot Coordination:** Enable concurrent execution of multiple tasks across multiple robots [19].
8. **Mobile App Integration:** Create a real-time monitoring and control application with a friendlier user interface [20].

Bibliography

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770-778.

[2] A. D. Sharma, P. A. S. Pimentel, and N. S. R. S. J. Gama, "A review of assistive robotics," *IEEE Access*, vol. 7, pp. 138935-138946, 2019.

[3] A. Y. Ng, "Machine learning for robotics," *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 100-105.

[4] Y. Liu, Y. Wang, Z. Wen and X. Yuan, "Automatic driving path planning based on A-Star algorithm," 2022 7th International Conference on Intelligent Informatics and Biomedical Science (ICIIBMS), Nara, Japan, 2022, pp. 19-21, doi: 10.1109/ICIIBMS55689.2022.9971587.

[5] M. H. Khan, "IoT-based smart systems for automation," *IEEE Access*, vol. 8, pp. 3456-3465, 2020.

[6] S. Lee, J. H. Lim, and Y. H. Kim, "Assistive robotics for human-robot collaboration," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 3002-3010, 2019.

[7] H. Wang, C. Li, and Y. Zhao, "CNN-based indoor object recognition with transfer learning," *IEEE Access*, vol. 10, pp. 56789-56798, 2023.

[8] X. Li, P. Chen, and R. Sun, "Autonomous navigation using Dijkstra's algorithm in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 112-119, 2024.

[9] S. Kim, J. Park, and T. Lee, "Sensor fusion of LiDAR and RGB-D cameras for intelligent robot navigation," *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3045-3056, 2023.

[10] A. Maier, A. Sharp and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things," 2017 Internet Technologies and Applications (ITA), Wrexham, UK, 2017, pp. 143-148, doi: 10.1109/ITECHA.2017.8101926.

[11] N. Kumar, K. Kumar and A. Kumar, "Application of Internet of Things in Image Processing," 2022 IEEE Delhi Section Conference (DELCON), New Delhi, India, 2022, pp. 1-5, doi: 10.1109/DELCON54057.2022.9753308.

[12] Last Minute ENGINEERS https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/

[13] THE ENGINEERING PROJECTS https://www.theengineeringprojects.com/2018/10/introduction-to-hc-sr04-ultrasonic-sensor.html

[14] L. J. Guibas and S. M. LaValle, "Real-Time Search for Optimal Paths in Robotics," *IEEE Trans. Robotics*, vol. 14, no. 1, pp. 7-22, Feb. 1998, doi: 10.1109/70.661311.

[15] Y. Xu, J. Xue, and W. Yang, "Robust Speech Recognition Based on Deep Neural Networks," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 730-741, Apr. 2013, doi: 10.1109/TASL.2012.2216093.

[16] A. B. A. E. Salama, S. M. Shatnawi, and A. M. J. Al-Qutayri, "Energy-Efficient Mobile Robot Navigation Systems: A Review," *IEEE Access*, vol. 8, pp. 140789-140800, 2020, doi: 10.1109/ACCESS.2020.3013571.

[17] N. A. S. Al-Khateeb, A. S. Abd El-Latif, and M. A. El-Dosuky, "Edge Computing for IoT Systems: A Survey and Future Directions," *IEEE Access*, vol. 8, pp. 114700-114716, 2020, doi: 10.1109/ACCESS.2020.3002610.

[18] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, 2018.

[19] R. A. L. S. Oliveira, L. C. S. Magalhães, and S. A. R. Ribeiro, "Multi-Robot Task Allocation: A Review," *IEEE Access*, vol. 7, pp. 116436-116449, 2019, doi: 10.1109/ACCESS.2019.2934101.

[20] J. Lee, J. Choi, and S. Lee, "Development of a Mobile App for Robot Control and Monitoring," *IEEE Trans. on Industrial Electronics*, vol. 65, no. 5, pp. 4401-4410, May 2018, doi: 10.1109/TIE.2017.2753725.