Jbot's Automatic Navigation System Based on a Machine Learning Model

Yang Yan
Department Name
University of Oklahoma
Norman, Ok
email@email.com

Guanchong Huang
Department Name
University of Oklahoma
Norman, Ok
email@email.com

Shiyuan Liu
Computer Science
University of Oklahoma
Norman, Ok
Shiyuan.Liu-1@ou.edu

ABSTRACT

In this project, we worked on making a small robotic car, called Jbot, smarter at moving around on its own using machine learning. We tried two different ways to teach Jbot: one to avoid bumping into things and another to follow a path marked on the ground. We saw some good results, but we also found some problems like Jbot getting too hot. Going forward, we want to combine the two ways we taught Jbot to make it even smarter and fix the overheating problem.

1 Introduction

Our project focuses on developing a robotic car called "Jbot" that can autonomously navigate its environment using machine learning, a branch of computer science that enables machines to learn and adapt from past experiences. The main goal of Jbot is to operate independently without a lot of human intervention. To achieve this goal, we employed two different strategies. The first strategy is to teach Jbot how to avoid obstacles and thus prevent collisions. This is critical to its ability to navigate safely in any environment. The second strategy aims to teach Jbot to follow specific paths marked on the ground, which is critical for directional movement and task performance in controlled spaces. By integrating these methods, we aim to enhance Jbot's understanding of its surrounding environment and improve its decision-making process. Through trial and error and incremental learning, Jbot is expected to become more proficient at interpreting various scenarios and navigating with greater accuracy. This dual approach not only expands Jbot's operational capabilities but also lays the foundational framework for the more complex tasks Jbot may encounter in future deployments.

2 Methodology

2.1 Workflow

In this study, we developed an autonomous navigation system incorporating two main components (shown in Figure 1): road following and collision avoidance. The process began with image collection, where we captured and labeled a dataset using a gamepad and a live camera to gather diverse training samples. Using this dataset, we built and trained a ResNet18 deep learning model to enable effective navigation and ensure reliable performance in varied scenarios. After training, the model was rigorously tested in both simulated and real-world environments. The system successfully demonstrated its ability to follow roads and avoid obstacles, operating autonomously without human intervention, showcasing promising results.

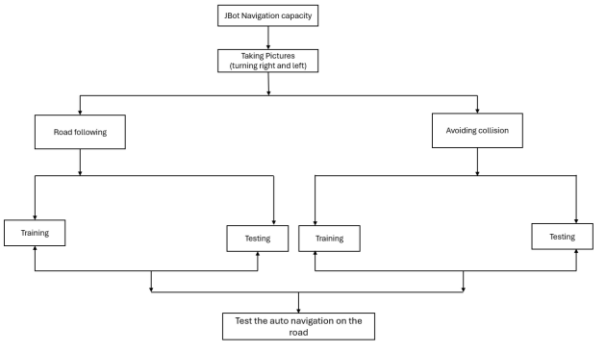

Figure 1: Workflow of the Study

2.2 Data Collection:

We began by teaching Jbot to navigate safely by avoiding obstacles, focusing specifically on executing right and left turns. This required identifying edges and corners, which are common obstacles in real-world environments. Using a gamepad, we manually controlled the car to capture images and labeled these images to create a robust dataset. For this purpose, we collected approximately 100 images of right and left turns, depicting various corners and edges encountered during navigation. These images provided diverse scenarios essential for training the system to recognize and respond to obstacles effectively (as illustrated in Figure 2 and Figure 3).

Figure 2: Edge Detection Photo

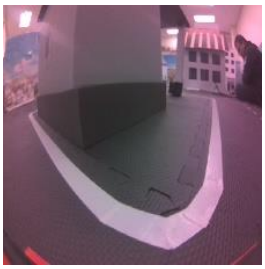

Figure 3: Corner Detection Photo.

Following the development of the collision avoidance system, we shifted our focus to teaching Jbot how to follow a prescribed path, specifically a circular path, without deviating from the road. Using the same gamepad, we manually controlled the car to capture images while labeling them for training purposes (shown in Figure 4). We collected up to 100 images, highlighting key features of the road, such as boundaries and inner and outer circle sections marked by solid lines. These images were meticulously organized in a separate "road following" folder, distinct from the collision avoidance dataset. This systematic approach helped Jbot accurately learn road boundaries, significantly improving its ability to navigate predefined routes with consistency and precision.

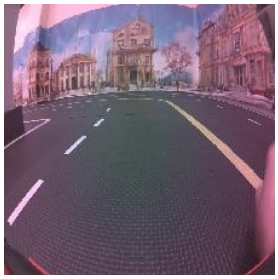

Figure 4: Road Following photo

To enhance control and flexibility during the testing phase, we integrated a handheld remote control into our setup (Figure 5). The device is designed to display the Jbot's location and distance to its next destination, allowing us to precisely guide its movement. Through the remote control, we can adjust the Jbot's direction based on real-time data, making it very effective for dynamic testing. Using this remote, we can also take new photos to continuously improve Jbot's learning data, ensuring that it learns to navigate with increasing accuracy. This interactive approach not only improves the quality of our tests, it also significantly facilitates the training process, allowing Jbot to adapt to its surroundings more effectively.

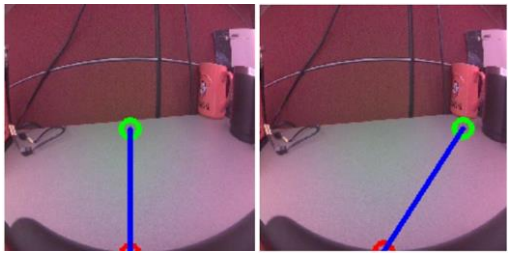

Figure 5: Remote Control: Original State [left] and Right Turn [right]

2.3 Building ResNet18 model in Collision Avoidance and road Following

Since the model is same for the two systems, we will show the general workflow.

2.3.1 Connection the Gamepad to Label Images

We intend to connect the left and right vertical axes of the gamepad using the ‘dlink’ function. This function offers an advantage over the traditional `link` function by enabling a transformation to be applied between the input source and the output target. This capability is particularly valuable for adapting the gamepad's input to generate accurate labeled data. By utilizing ‘dlink’, we aim to create a more efficient and precise workflow for integrating gamepad controls into the image labeling process, ultimately improving the quality and reliability of the data used to train our navigation system.

2.3.2 Define a ML model and training the model

ResNet18, short for Residual Network with 18 layers, is a popular deep learning model introduced by Microsoft Research. It is part of the ResNet family, which revolutionized deep neural networks by addressing the vanishing gradient problem commonly faced in training deep models. ResNet18 employs residual learning through skip connections, allowing the model to learn identity mappings more effectively. These connections enable information to bypass one or more layers, facilitating the training of deeper networks without performance degradation.

In our study, ResNet18 was utilized as the backbone for training the navigation system, leveraging its ability to extract high-level features from images, such as edges, corners, and road boundaries. This model is especially suited for tasks requiring robust feature extraction and classification, making it ideal for applications like obstacle avoidance and road following. Its relatively lightweight architecture and proven accuracy ensured efficient training and testing within our project. In addition, 70% of data was used in training and rest of them used as a testing dataset.

After several times testing, we made modifications to the parameters in the ML model for collision_avoidance to enhance its decision-making capabilities, shown in Figure 6. Notably, we introduced right-turn functionality for Jbot, which greatly expands its navigation options and intelligence. This is a significant improvement because previously Jbot was limited to going straight

or turning left, limiting its ability to effectively navigate complex paths.

```
#     blocked_slider.value = prob_blocked

    if lprob_blocked > 0.5:
        robot.right(speed_slider.value)
    elif rprob_blocked>0.5:
        robot.left(speed_slider.value)
    else:
        robot.forward(speed_slider.value)
```

Figure 6: Collision avoidance code we modified

To further optimize Jbot's automatic navigation performance, we tuned four specific parameters: speed gain, steering gain, steering kd, and steering bias, showing in Figure 7. These adjustments are critical to optimizing Jbot's responsiveness and accuracy. The speed gain is set to 0.26 to ensure that the Jbot moves at optimal speed, balancing fast navigation without compromising stability. Steering gain is set to 0.07 for smoother, more accurate turns, reducing the risk of overshooting or not being properly aligned with the path.

Additionally, we kept the steering kd and steering bias at 0.00, which after many tests proved to provide the best balance for our current navigation challenges. The changes were tested extensively in real-world scenarios, with Jbot demonstrating its improved ability to navigate around a house without crossing any solid lines. The combination of these customized parameter settings and code enhancements significantly improves Jbot's adaptability and accuracy in navigating real-world environments.

Figure 7: Tuning four specific key parameters for navigation

3 Results

3.1 Performance

Jbot does an excellent job of following designated paths and keeping its route within road markings. The successful implementation of a machine learning-based road tracking model is a key achievement. The Jbot is able to drive on a circular track, deftly avoiding any solid lines throughout the track. This achievement was achieved by using up to 150 photos to train the model (shown in Figure 8), which enhanced Jbot's ability to accurately identify and respond to road boundaries. Shifting Jbot's focus from obstacle avoidance to path following poses significant challenges as different types of training images and settings are required. Each model requires unique image processing and feature identification methods, making conversion a complex but rewarding endeavor.

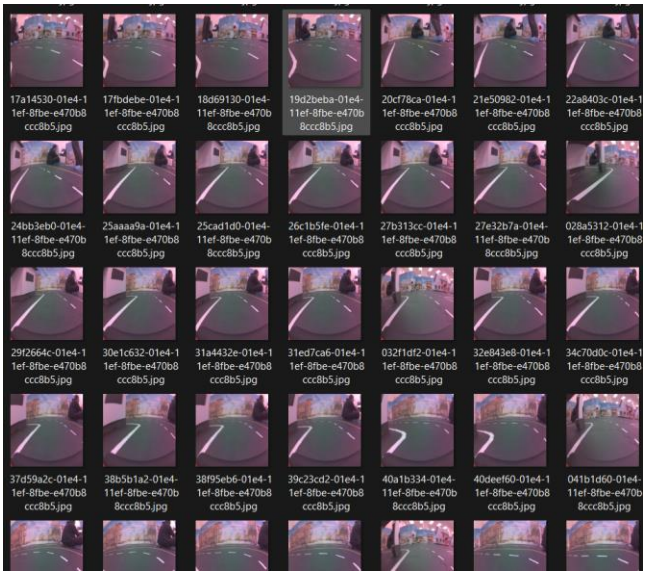

Figure 8: Training dataset

3.2 Challenges

One of the big challenges we encountered was managing the heat dissipation of the Jbot. After 3-5 mins, the Jbot is prone to overheating, which not only damages its internal components but also poses a risk to its overall functionality and longevity. Addressing this issue is critical to maintaining Jbot's operational effectiveness and security.

3.3 Experiments

Throughout the project, we conducted a lot of experiments, and every week we had the opportunity to go to the laboratory to conduct track experiments to improve Jbot's performance on the track. These tests are critical for adjusting various parameters such as speed gain and steering accuracy, ensuring that the Jbot can navigate smoothly without crossing any boundaries. By adjusting these settings, Jbot is able to maintain the correct speed and perform precise turns, significantly improving its adhesion to the trail. This rigorous testing phase is critical to optimizing Jbot's response to real-world driving conditions, thereby increasing its reliability and effectiveness in following marked paths.

4 Future Work

In the next phase of our project, it will be critical to address the thermal management issues that have arisen. Jbot faced overheating issues that severely limited its uptime and reliability. Developing more efficient cooling systems or modifying existing hardware to dissipate heat more efficiently will allow Jbot to operate for

extended periods of time without the risk of thermal damage. This enhancement is critical to the longevity and safety of the robot.

Additionally, we plan to improve Jbot's functionality by integrating collision avoidance and road following methods into a more powerful navigation system. This model integration will enable Jbot to switch seamlessly between navigating around collision and following paths, making it more useful and effective in a variety of environments. By combining these models, we aim to enhance the Jbot's decision-making capabilities, allowing it to more intelligently and autonomously assess and react to its surroundings.

5 Conclusion

This project was very inspiring and gave us an idea of the potential of using machine learning to enhance robot autonomy with Jbot. Throughout the process, we not only honed Jbot's ability to prevent collisions and identify paths, but also pointed out key areas for improvement, such as its tendency to overheat. Overheating has become a significant issue affecting the operational life and safety of Jbot, and it is important to address this issue. In the future, we can integrate the two learning modes obstacle avoidance and path following into one system. Improve Jbot's decision-making process to make it easier and more intelligent to handle unpredictable environments. By combining these models, the overall functionality of Jbot will be significantly enhanced and more complete.