IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Road Following And Collision Avoidance Using Jetbot

[1]Srinithi K, [2]Deeksha K, [3]Janarthanan N, [4]Vinivarshini R G,[5]Hari Priya R

[1]UG Student, [2]UG Student, [3]UG Student, [4]UG Student,[5]Assistant Professor

[1]Department of Electronics and Communication Engineering, [2] Department of Electronics and Communication Engineering, [3]Department of Information Technology, [4] Department of Electronics and Communication Engineering,[5] Department of Computer Technology.

[12345]Bannari Amman Institute of Technology, Sathyamangalam, India

Abstract: The development of a road-following and collision-avoidance system uses the Jetson Nano-powered JetBot. JetBot is an open-source robot platform designed for learning and experimentation in robotics and AI, equipped with various sensors and cameras to facilitate autonomous navigation. This project focuses on developing a robust system for autonomous vehicles that can effectively follow roads and avoid collisions. The system integrates road detection with real-time obstacle avoidance using advanced machine learning and sensor technologies such as computer vision, LiDAR, and ultrasonic sensors. A convolutional neural network (CNN) identifies lane markings and road boundaries for accurate road following, while reinforcement learning and path planning algorithms ensure proactive collision avoidance. Simulations and real-world tests demonstrate significant improvements in navigation reliability across diverse environments, advancing the safety and efficiency of autonomous driving technologies. This project aims to develop an integrated system for autonomous vehicles focusing on road following and collision avoidance. Finally, the outcome of this project is to Integrate the Road Following and Collision Avoidance models to allow Jetbot to autonomously work with an average speed while simultaneously avoiding collisions with obstacles. Furthermore, provide smooth system-to-system communication for effective operation and real-time decision-making. Use thorough testing procedures, including worst-case situations, to find any possible flaws. Use cutting-edge machine learning strategies to improve system performance over time. Create a cooperative atmosphere that promotes creativity and information exchange. Establishing a feedback loop with end users will also assist in quickly identifying and resolving any real-world issues. The Jetbot will handle the needs of dynamic and complicated settings by concentrating on these factors, which will lead to increased autonomy and safety..

Index Terms **- Road following, Collision Avoidance, Jetbot, Sensors, Convolutional Neural Network (CNN), Robot, Reinforcement learning.**

I. Introduction

The project on road following and collision avoidance using JetBot focuses on leveraging AI and sensor technologies to navigate autonomous vehicles efficiently. JetBot, a small and affordable AI-powered robot, is equipped with cameras and sensors to detect and follow predefined paths while avoiding obstacles. By integrating computer vision and machine learning techniques, the JetBot processes real-time data to make intelligent decisions, ensuring smooth and safe navigation. This application is particularly useful in autonomous vehicles, robotics, and other smart systems where obstacle avoidance and path-following are critical for operational safety and efficiency. The project also demonstrates the potential of deep learning for real-world problem-solving in robotics.

Since recent decades, digital image processing, image analysis and machine vision have been sharply developed, and they have become a very important part of artificial intelligence and the interface between human and machine grounded theory and applied technology. The core of the system lies in its dual capabilities: road detection and real-time obstacle avoidance. A convolutional neural network (CNN) is employed to accurately identify lane markings and road boundaries, enabling the vehicle to follow the road reliably. This deep learning approach ensures that the system can adapt to varying road conditions and maintain lane discipline even in challenging scenarios such as poor visibility, sharp curves, or worn-out lane markings. Training the models for road following and collision avoidance in JetBot involves optimizing neural networks, such as convolutional neural networks (CNNs), to accurately detect roads and obstacles from real-time sensor and camera data. Careful tuning is required to ensure that the JetBot can follow paths smoothly and avoid collisions while adapting to varying environmental conditions. Performance evaluation includes assessing the accuracy of road detection, obstacle recognition, and the robot's ability to adjust its trajectory in real-time. Challenges in this process include ensuring high-quality training data, handling diverse road conditions, and maintaining precise obstacle detection.

II. Literature Review

Preston Leigh and Yuan Xing (2022) [1] proposed the idea of working on JetBot without assistance from a human; the JetBot hopes to use the fully trained CNN to drive across a fictitious road. The CNNs utilized in this study were imported from PyTorch and trained using a Reinforcement-Learning approach. To reduce the result's margin of error, the pre-trained models are imported. This ideal CNN combination is determined by taking a "for each" training style. This process shows the use of RestNet18 with Adam optimizer works well in Jetbot performance and operation.

Minyoung Christina Lee et al (2020) [2] a modified deep learning-based object identification model is used in the JetBot tracking function implementation. For a workable operation in the suggested framework involving mobile robots, a transfer learning algorithm with a pretrained mobile model is present. By combining these cutting-edge methods, the robot is guaranteed to function well in real-world situations and exhibit enhanced tracking and recognition abilities, making it a useful instrument for applications needing human contact and autonomous navigation.

Shinji Kawakura et al (2020) [3] the centerpiece of the system is JetBot, a robot in the shape of a car equipped with an NVIDIA AI focused board. JetBot is able to locate and avoid a variety of items. They created a system that can accurately identify barriers using deep learning-based techniques (techs), avoid obstacles on its own, and automatically distribute small products to manual agri-workers and managers. The system consists of designing, tuning, data collection, deep learning execution, indoor experiments, and assessment, highlighting the practical applications of this innovative technology in agriculture which was implemented on real-time objects.

Aarav Nigam et al (2024) [4] presents a useful implementation of the Reinforcement Learning Algorithm for challenges involving visual path following on real robots. By applying Deep Q learning to JetBot, a nonholonomic wheeled mobile robot,the study evaluates normalization methods including Batch Normalization, Layer Normalization, Min-Max Normalization, and Z-score Normalization. It presents a new method that uses normalization techniques to improve generalization and speed up training. Results are tested and compared to the basic Deep Q learning model.

Cheng Chang et al [5] as they use an innovative vision-based architecture to assess the geometric characteristics and determine the path, incorporating edge computing and machine learning. The use of UGVs is becoming more common for indoor and outdoor applications. However, GPS can be unreliable in some scenarios due to signal issues. In addition, the goal is to do a scenario analysis in order to identify the optimal performance setting. The prototype exhibits the proper qualities and performance, based on the experimental results.

Hai-Wu Lee et al (2017) [6] introduces a safer Bluetooth-controlled robot, operable via a smartphone, featuring collision avoidance through ultrasonic sensors. The system emulates human driving by maintaining a safe following distance, ensuring the bot can halt before any impact. This system design prioritizes safety and operational efficiency, making it a valuable tool for applications requiring precise navigation and interaction within various environments.

Teck Chew Ng et al (2004) [7] presents a vehicle following system with the obstacle prevention by combining a novel tracking and detection algorithm with a path-planner for autonomous navigation. This approach enhances the road following performance and ensures safety, highlighting the system's robustness and practical applicability in challenging environments.

III. Design

The objective of the "Road Following and Collision Avoidance using JetBot" project is to integrate the best road following and obstacle prevention models to allow JetBot to navigate the track at its own speed while simultaneously avoiding collisions with obstacles. Through the integration of road following and collision prevention in a Jetbot gives the ability to independently follow a predetermined route or track. The Jetbot will be guided to stay on the intended course by the road following model and also gives the capacity to recognize impediments in real time. In order to keep the Jetbot from running into obstacles, the collision avoidance model will initiate an action (such as stopping). Additionally, making the Jetbot realize both features at the same time. The Jetbot must be able to follow both the path and steer clear of any unforeseen impediments that may arise in its path.

Artificial Intelligence plays a crucial role in the development of autonomous bots and vehicles, significantly enhancing their capabilities in tasks such as road following, collision prevention and pothole detection. Collision avoidance is another part where AI proves indispensable by using reinforcement learning algorithms, the JetBot can learn from its environment and make real-time decisions to navigate around obstacles,improving its performance through the interaction with surroundings,prioritizing safety while optimizing the path. Moreover, AI facilitates sensor fusion,integrating data from cameras, LiDAR, and ultrasonic sensors to create a comprehensive situational awareness. This multi-sensor approach enhances the effectiveness and reliability of the JetBot's Navigation system.Overall AI's role in making this project is fundamental in achieving autonomous navigation and demonstrates the transformative potential of AI in developing intelligent, autonomous systems capable of handling real-time challenges.

The NVIDIA JetBot is a specialized platform designed for autonomous vehicle making tasks, specifically for road following and collision avoidance making it easy to integrate with AI. With an NVIDIA Jetson Nano, it uses strong AI and computer vision technologies to interpret data and make decisions in real time. This makes it possible to precisely identify obstructions and lane markers, guaranteeing safe and easy navigation. The JetBot is a more dependable and efficient option for these applications since a standard bot does not have the processing capacity and specific algorithms needed for such difficult jobs. The core advantage of the JetBot lies in its integration of the NVIDIA Jetson Nano, a powerful AI computing device optimized for real-time processing of complex data. The JetBot's AI algorithms fuse data from these sensors to create a detailed map of the environment, enabling the navigation system. The Nvidia JetBot's combination of advanced hardware and software capabilities makes it an ideal choice for developing and testing autonomous navigation systems.

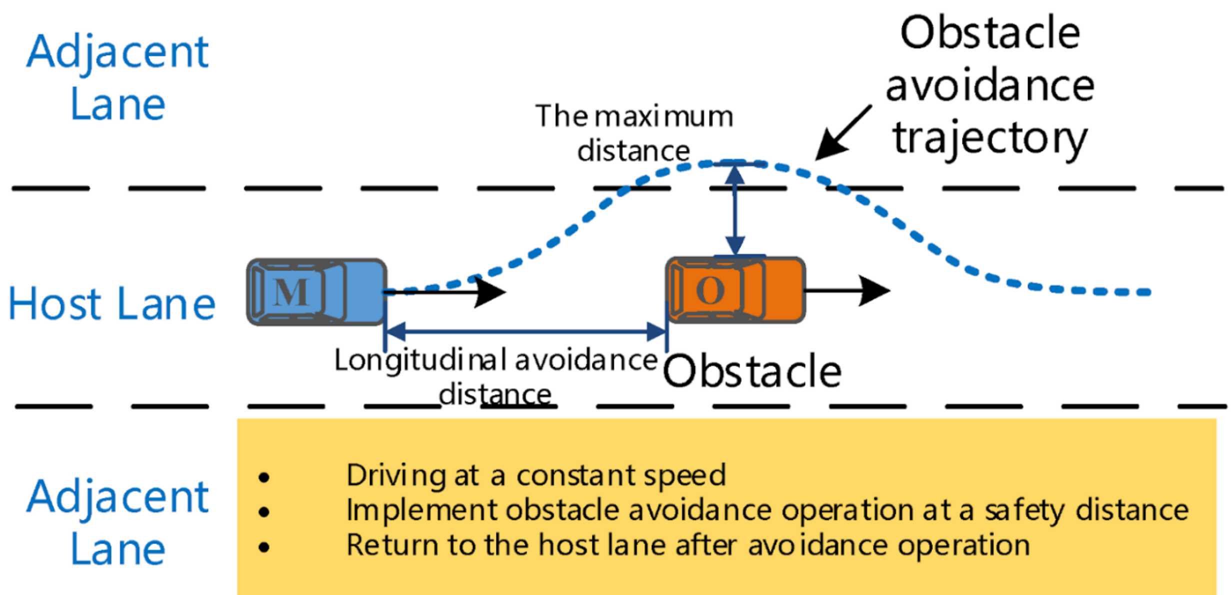


Fig.1: The working process of the JetBot.

Deep learning has become a pivot technology in road following and collision avoidance for autonomous driving systems. Using CNNs and RNNs, DL models can process vast amounts of sensor data such as camera feeds, LiDAR and ultrasonic sensors to make real-time decisions. CNNs are particularly adept at recognizing road features such as road markings, lane markings, obstacles, and traffic signs, helping vehicles stay within lanes and navigate complex road environments. In collision avoidance, DL models analyze dynamic data to predict and track the movement of other vehicles, pedestrians, and objects, enabling timely maneuvers to avoid collisions.

A family of deep learning models called convolutional neural networks (CNNs) was created especially for handling organized, grid-like input, like pictures. CNNs can automatically learn spatial hierarchies of features from input pictures, which makes them especially well-suited for image identification and classification applications. The convolutional layer, the central part of a CNN, detects certain characteristics like edges, textures, or colors by using tiny, learnable filters (also called kernels) that move over the input picture. In deeper layers, these characteristics grow more abstract and complex. In this JetBot project, Convolutional Neural Networks (CNNs) are crucial for road following and collision avoidance. They process visual data to detect lane markings and road boundaries, enabling the JetBot to follow paths accurately. CNNs also identify obstacles, helping the JetBot navigate safely and avoid collisions in real-time.

The combination of LiDAR, ultrasonic sensors, and cameras in JetBot provides a comprehensive solution for navigation and obstacle avoidance. LiDAR i.e., Light Detection and Ranging sensors are excellent for precise distance measurements by emitting laser beams and calculating the reflection time. This helps JetBot map its surroundings and detect obstacles with high accuracy, even at longer ranges. LiDAR sensors are vital in the JetBot's road-following and collision avoidance tasks due to their ability to create precise 3D maps of the environment. These sensors perform well in various lighting conditions and offer a longer range compared to other sensors.Real-time images of the surroundings are captured by cameras, which are essential for visual perception. By recognizing lane markers and visualizing obstructions, they allow CNNs to follow roads. Rich contextual information from cameras helps JetBot negotiate challenging areas. Together, these sensors ensure that JetBot can navigate effectively, avoiding obstacles and following paths accurately in diverse conditions.

In the development of autonomous navigation for JetBot, various algorithms are employed to achieve road following and collision avoidance. These include Convolutional Neural Networks (CNNs) and Sensor Fusion for lane detection, reinforcement learning for obstacle avoidance, and sensor fusion for enhanced situational awareness, collectively ensuring safe and efficient navigation in diverse environments.

The development of JetBot is a versatile process that brings AI and robotics together. JetBot is an open-source educational robot based on the NVIDIA Jetson Nano, developed by NVIDIA. It is designed to provide hands-on experience with AI applications. The JetBot kit includes a chassis, wheels, motors, camera and various sensors, making it easy to assemble and program. The JetBot platform supports multiple sensors and neural networks, enabling functionalities like object recognition, collision detection and path following. This platform is compatible with popular AI frameworks such as Tensorflow and PyTorch, allowing for a wide range of AI applications.

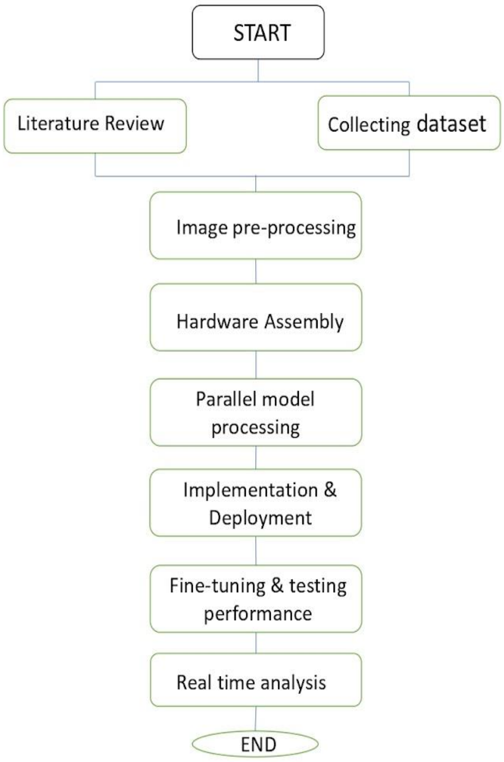


Fig.2: The Process flow of design modules.

Parallel model processing in JetBot enables simultaneous handling of road-following and collision-avoidance tasks, enhancing the bot's autonomous navigation capabilities. The Jetson Nano, with its CUDA-enabled GPU, allows for parallel processing, which can efficiently manage multiple neural network models in real-time. When two models are set up in parallel, one concentrates on road following by examining the route and directing the JetBot's course, while the second model keeps an eye out for obstructions to avoid crashes. These models operate in parallel, processing road features and obstacle data at the same time by utilizing the Jetson Nano's GPU cores. This dual-task processing enables the bot to make prompt decisions without delay, which is crucial for establishing a seamless and secure navigation experience.

Both models receive real-time data from the camera. While the collision-avoidance model recognises surrounding barriers and generates stop or avoid orders, the road-following model recognises lane or path edges and outputs steering commands. When an impediment is recognised, a decision-making layer prioritizes collision-avoidance commands by integrating the outputs of both models. The JetBot can respond to complicated settings by balancing directional accuracy and safe navigation by running models in parallel. In dynamic environments, this configuration optimizes JetBot's reactivity and efficiency.

From model training to real-world testing, there are several crucial processes involved in implementing and deploying road-following and collision-avoidance models on the JetBot. The JetBot is first manually driven through a variety of road conditions in order to gather a dataset. Images are then labeled with the proper directions (left, right, and straight) for road following and clear or blocked routes for collision avoidance. Two distinct machine learning models – one for collision detection and the other for road following – are then trained using this labeled dataset. The models are optimized for Jetson Nano deployment after training. Reducing the model size and translating it to TensorRT format are two possible optimisations that make the Nano run the models more quickly and effectively. Both models are loaded onto the JetBot after they have been optimized, and they operate in parallel to process live camera data in real time.

Fig.3: JetBot Hardware and Final Bot.

For safe, responsive navigation, the JetBot's road-following and collision-avoidance system requires real-time analysis. The JetBot continuously records live video from its camera during real-time operation, which is then fed into the road-following and collision-avoidance models that are operating simultaneously. Real-time processing necessitates models that are efficient and optimized to handle video frames with the least amount of latency, so the Jetson Nano's GPU processes both models simultaneously for speedy analysis.

IV. Discussion

For real-time autonomous navigation, this research shows how well TensorRT-optimized deep learning models work together. The dual-purpose capacity of the JetBot to avoid collisions and follow roads demonstrates how model optimization and GPU acceleration allow it to navigate complicated situations with swift, responsive responses. The simultaneous operation of both models was made possible by TensorRT's latency reduction, which is crucial for real-time obstacle identification and avoidance. Although the collision-avoidance model's accuracy was good, its robustness in a variety of settings might be further improved by adding a wider range of obstacles to the training dataset. All things considered, this study provides a strong basis for upcoming developments in autonomous robotics using effective, responsive deep learning algorithms.

V. Results

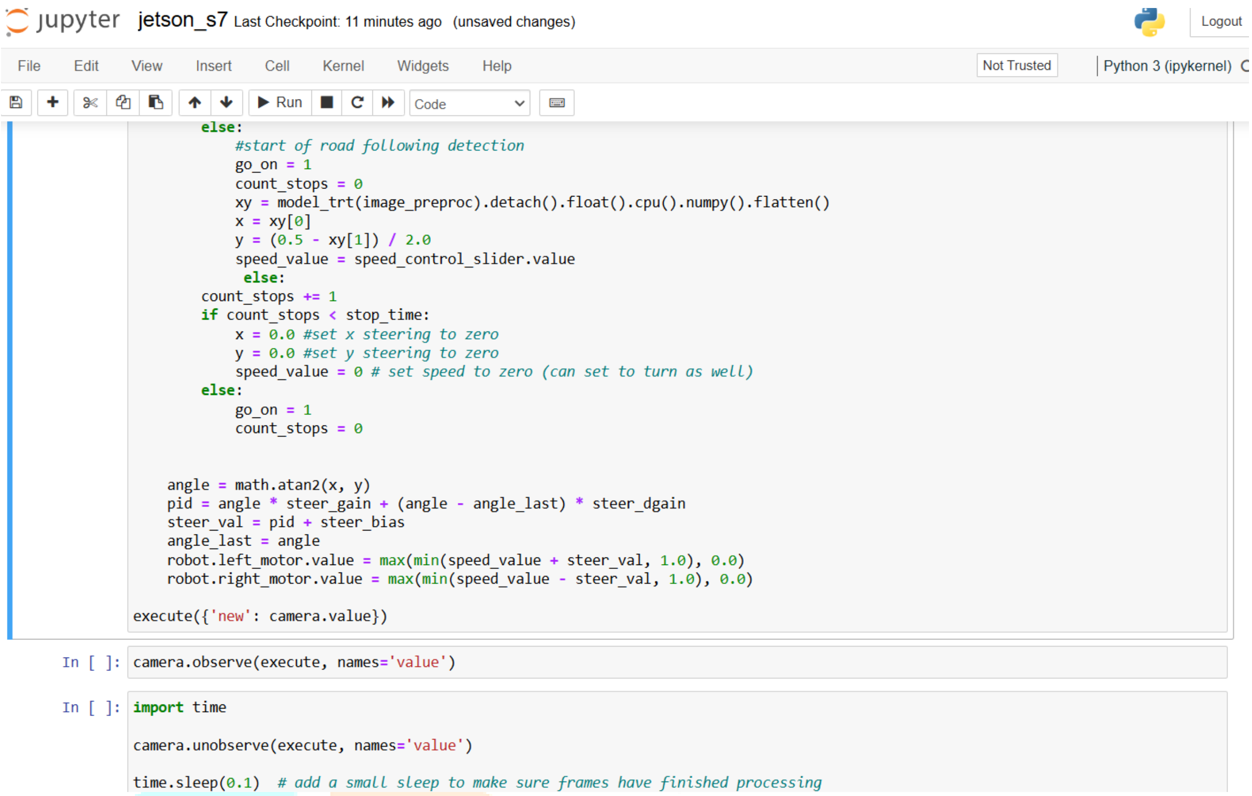
Real-time road following and collision avoidance capabilities were successfully combined in the JetBot project to produce an effective autonomous navigation system. By combining TensorRT-optimized models for both tasks, the JetBot was able to recognize and avoid obstacles with ease while navigating its path. Because of the fine steering control provided by the road-following model, the JetBot was able to maintain track alignment and smoothly transition between straight and curved sections. Rapid processing made possible by running this model on the GPU was essential for preserving constant path alignment under varied circumstances. In the meantime, the JetBot was able to take evasive action in real time thanks to the collision-avoidance model, which correctly distinguished between clear and obstructed areas of the track and detected impediments like cars or objects. Both models were able to function simultaneously thanks to the TensorRT improvement, offering fast, reliable inferences that ensured stable and responsive autonomous navigation. This integration showcased a robust, real-time solution capable of handling dynamic environments effectively.

```
In [1]: import torch
        device = torch.device('cuda')

In [ ]: import torch
        from torch2trt import TRTModule

        model_trt = TRTModule()
        model_trt.load_state_dict(torch.load('best_steering_model_xy_trt.pth')) # well trained road following model

        model_trt_collision = TRTModule()
        model_trt_collision.load_state_dict(torch.load('best_model_trt.pth')) # well trained collision avoidance model

In [ ]: import torchvision.transforms as transforms
        import torch.nn.functional as F
        import cv2
        import PIL.Image
        import numpy as np

        mean = torch.Tensor([0.485, 0.456, 0.406]).cuda().half()
        std = torch.Tensor([0.229, 0.224, 0.225]).cuda().half()

        def preprocess(image):
            image = PIL.Image.fromarray(image)
            image = transforms.functional.to_tensor(image).to(device).half()
            image.sub_(mean[:, None, None]).div_(std[:, None, None])
            return image[None, ...]

In [ ]: from IPython.display import display
        import ipywidgets
        import traitlets
        from jetbot import Camera, bgr8_to_jpeg
```

Fig.3: Code implemented for the JetBot (1/2)

The optimisation of model performance for real-time processing on the Jetson Nano's constrained resources was one of the many difficulties this JetBot project faced. In order to achieve low latency while maintaining road-following and collision-avoidance model accuracy, extensive model tuning and size reduction were needed. Additional challenges were presented by environmental variability, which frequently necessitated further data collection and retraining to increase model resilience. Examples of these variables included variations in lighting, surface textures, and unforeseen impediments. Managing the concurrent processing of two models presented another difficulty, occasionally resulting in latency problems that impacted the JetBot's responsiveness. To overcome these obstacles and achieve steady, autonomous navigation, iterative testing and fine-tuning were necessary.

Additional sensors, such LiDAR or infrared, could be integrated into this JetBot project in the future to increase obstacle recognition and navigation accuracy in low light. The JetBot may be able to identify intricate road patterns and lane markers more successfully if more sophisticated computer vision techniques are used, such as semantic segmentation. Further lowering latency and facilitating quicker real-time responses would be possible by using more effective neural networks, such as lightweight or quantised models. The JetBot's capabilities could be more broadly applied to a greater variety of navigation scenarios and applications by broadening the dataset to cover a variety of surroundings, such as outdoor terrains.




```
        else:
            #start of road following detection
            go_on = 1
            count_stops = 0
            xy = model_trt(image_preproc).detach().float().cpu().numpy().flatten()
            x = xy[0]
            y = (0.5 - xy[1]) / 2.0
            speed_value = speed_control_slider.value
             else:
        count_stops += 1
        if count_stops < stop_time:
            x = 0.0 #set x steering to zero
            y = 0.0 #set y steering to zero
            speed_value = 0 # set speed to zero (can set to turn as well)
        else:
            go_on = 1
            count_stops = 0


    angle = math.atan2(x, y)
    pid = angle * steer_gain + (angle - angle_last) * steer_dgain
    steer_val = pid + steer_bias
    angle_last = angle
    robot.left_motor.value = max(min(speed_value + steer_val, 1.0), 0.0)
    robot.right_motor.value = max(min(speed_value - steer_val, 1.0), 0.0)

execute({'new': camera.value})
```

In [ ]:
```
camera.observe(execute, names='value')
```

In [ ]:
```
import time

camera.unobserve(execute, names='value')

time.sleep(0.1)  # add a small sleep to make sure frames have finished processing
```

Fig.4: Code implemented for the JetBot (2/2)

VI. Conclusion

In conclusion, the JetBot's implementation of road-following and collision-avoidance capabilities shows how easily available robotics and AI may facilitate autonomous navigation. This research demonstrated the potential of edge AI on the NVIDIA Jetson Nano platform by training the JetBot to understand visual data, follow predetermined courses, and identify obstacles in real-time. The research demonstrates the iterative process of creating reliable, real-time navigation systems, from data gathering and model training to optimisation and deployment. By handling path detection and collision avoidance duties simultaneously, the dual-model technique allowed for safe and accurate management. Although it was difficult to optimize for real-time performance on a small, low-power device like the Jetson Nano, the experiment showed that it was possible to get the quick, dependable answers required for autonomous tasks.

In addition to highlighting the value of testing, fine-tuning, and real-time monitoring, this project laid the groundwork for future improvements like adding more sensors or sophisticated environmental awareness. All things considered, the JetBot project is a prime example of how artificial intelligence (AI) can teach and adjust to its environment, opening the door for more widespread uses in industrial, educational, and personal robotics.

References

[1] P. Leigh and Y. Xing, "Evaluation of Multiple Convolutional Neural Networks in Training the NVIDIA JetBot," 2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, NY, USA, 2022, pp. 0524-0530, doi: 10.1109/UEMCON54665.2022.9965727.

[2] M. C. Lee and M. Lee, "Deep Learning-based Target Following and Obstacle Avoidance Methods in Mobile Robots," 2022 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Yeosu, Korea, Republic of, 2022, pp. 1-4, doi: 10.1109/ICCE-Asia57006.2022.9954653.

[3] Kawakura, Shinji & Shibasaki, Ryosuke. (2020). Deep Learning-Based Self-Driving Car: JetBot with NVIDIA AI Board to Deliver Items at Agricultural Workplace with Object-Finding and Avoidance Functions. European Journal of Agriculture and Food Sciences. 2. 10.24018/ejfood.2020.2.3.45.

[4] Nigam, Aarav & Sanodiya, Rakesh & Joshi, Piyush & Subhangi, Subhangi. (2024). Generalized Visual Path Following on Jetbot Using Normalization with Reinforcement Learning. 247-252. 10.1109/ARSO60199.2024.10557750.

[5] C. -C. Chang, Y. -H. Juan, C. -L. Huang and H. -J. Chen, "Scenario Analysis for Road Following Using JetBot," 2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 2020, pp. 403-406, doi: 10.1109/ECICE50847.2020.9302017.

[6] H. -W. Lee and C. -Y. Wong, "The study of the anti-collision system of intelligent wheeled robot," 2017 International Conference on Applied System Innovation (ICASI), Sapporo, Japan, 2017, pp. 885-888, doi: 10.1109/ICASI.2017.7988577.

[7] Teck Chew Ng, J. Ibanez-Guzman, Jian Shen, Zhiming Gong, Han Wang and Chen Cheng, "Vehicle following with obstacle avoidance capabilities in natural environments," IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, New Orleans, LA, USA, 2004, pp. 4283-4288 Vol.5, doi: 10.1109/ROBOT.2004.1302391.