AI-Driven Drone Navigation and Obstacle Avoidance in Webots Using Neural Networks for Real-Time Environmental Analysis

Koti Leela Sai Praneeth Reddy[1], Suhas S Bhat[2], Mukil LD[3], Nippun Kumaar AA[4]
Dept. of Computer Science and Engineering, Amrita School of Computing, Bengaluru,
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21102@bl.students.amrita.edu, bl.en.u4cse21200@bl.students.amrita.edu,
bl.en.u4cse21109@bl.students.amrita.edu, aa_nippunkumaar@blr.amrita.edu

***Abstract*—Obstacle avoidance is a very important aspect of autonomous drone navigation and calls for strong and efficient methods that would guarantee the safe and reliable operation of the drones. This paper presents a state-of-the-art Obstacle Avoidance System for drones using Convolutional Neural Networks. In this scheme, the proposed system will employ a CNN-based model to process real-time sensory data in order to accurately detect and classify obstacles in a drone's path of flight. Using the high spatial awareness and pattern recognition capability of CNNs, the system can determine collisions that will probably occur and correspondingly generate proper avoidance maneuvers. Experimental results show that the CNN-based approach in diverse and dynamic environments proves efficient, bringing about marked improvement in accuracy regarding the detection of obstacles and the reaction time in comparison with the previous traditional method. The proposed system implementation details, the process of training, and the performance evaluation are discussed in this paper, underlining the potential to enhance the safety and autonomy of drone operations.**

***Index Terms*—Drone navigation, obstacle avoidance, Convolutional Neural Networks (CNN), autonomous systems, real-time processing, sensory data, collision detection, avoidance maneuvers, spatial awareness, pattern recognition..**

I. PROBLEM STATEMENT

Autonomous navigation is a critical capability for drones, particularly in environments where human intervention is limited or impractical. Traditional obstacle avoidance methods, which often depend on predefined rules and sensor fusion techniques, have fundamental limitations on adaptability and generalization to new and dynamic environments. Such limitations can considerably affect the performance and safety of the drone, particularly in complex and unprepared environments.

In this context, this project aims to develop a machine learning-based intelligent obstacle avoidance system for autonomous drones using Convolutional Neural Networks. The project will leverage the Webots simulation platform for creating realistic and controlled environments for data collection, model training, and performance evaluation.

The project aims at improving the performance of autonomous drones, making them more effective and efficient in tracking routes in difficult environments. Deep learning for obstacle avoidance will, therefore, be a large stride forward in intelligent robotic systems with a wide range of applications across industries and use cases.

II. INTRODUCTION

A. Background

During recent times, autonomous robotics has made great strides due to innovations in machine learning and artificial intelligence. Autonomous drones have attracted much attention for use in surveillance, search and rescue, agriculture, and delivery. The capabilities of these autonomous drones to navigate complex environments without human intervention are a critical part of their functionality.

Obstacle avoidance is an intrinsic competence of autonomous drones to operate in dynamic and unpredictable environments. Traditional methods for obstacle avoidance frequently depend upon rule-based algorithms and sensor fusion techniques. Though these are effective in some situations, they are limited by the fact that they might not be able to generalize to new and unseen environments.

B. Motivation

The integration of machine learning, specifically deep learning, presents a bright solution for the deficiencies in traditional obstacle avoidance techniques. Deep learning models, specifically Convolutional Neural Networks, are models developed mainly for image processing tasks and can learn from visual inputs, recognize patterns, and make decisions from such recognition. With CNNs, drones could be made to be better understood and respond to the real world in a more adaptive and intelligent form.

Webots is a professional mobile robot simulation software that allows for the development and testing of autonomous robotic systems. It gives a realistic simulation environment for the safe development and evaluation of complex algorithms before deployment into real-world scenarios. This simulation environment is of utmost importance in the making of robust machine learning models, as it allows for the collection of vast amounts of data and experimentation without the risks associated with real-world testing.

III. Related Works

Devos et al.[1] addressed the development of autonomous drones for adaptive obstacle avoidance in real-world environments. The research focuses on implementing real-time obstacle detection and avoidance algorithms in dynamic scenarios, demonstrating practical applications of these technologies in autonomous systems. Woods and La[2] explored dynamic target tracking and obstacle avoidance using drones, emphasizing advanced visual computing techniques to enhance the drone's navigation capabilities in complex environments. Liang et al.[3] proposed an autonomous aerial obstacle avoidance system using LiDAR sensor fusion. Their study, published in PLOS ONE, integrates multiple LiDAR sensors to improve detection accuracy and avoidance efficiency.

Park and Cho[4] investigated collision avoidance for hexacopter UAVs based on LiDAR data in dynamic environments. Published in Remote Sensing, their research highlights the importance of LiDAR in providing reliable data for real-time obstacle avoidance. Ramasamy et al.[5] introduced a LIDAR obstacle warning and avoidance system for UAVs, detailed in Aerospace Science and Technology, showcasing the system's ability to enhance UAV safety through advanced sensing technology. Santos et al.[6] presented a UAV obstacle avoidance system using an RGB-D sensor at the International Conference on Unmanned Aircraft Systems (ICUAS), demonstrating the integration of depth-sensing technology for improved obstacle detection and navigation.

Ricardo et al.[7] discussed a low-cost, real-time obstacle avoidance system for mobile robots. Their paper, presented at the IEEE Annual Computing and Communication Workshop and Conference (CCWC), explored cost-effective solutions for real-time navigation. Aldao et al.[8] developed an obstacle avoidance algorithm for UAVs to navigate in dynamic building environments, published in Drones. Their research addresses the complexities of indoor navigation, providing robust solutions for UAV path planning.

Robinson and Jang[9] worked on developing a 2D LiDAR collision avoidance algorithm for quadcopters, focusing on enhancing the precision and reliability of obstacle detection mechanisms. Gyenes et al.[10] investigated the use of genetic algorithms for real-time obstacle avoidance for LiDAR-equipped mobile robots, published in Sensors. Their study evaluated the feasibility of genetic algorithms in optimizing obstacle avoidance strategies. Tu and Juang[11] presented UAV path planning and obstacle avoidance based on reinforcement learning in 3D environments, published in Actuators, highlighting the potential of reinforcement learning in improving UAV navigation.

Dong et al.[12] explored control methods for autonomous flight avoidance barriers of UAVs in confined environments, published in Sensors, providing comprehensive solutions for navigating and avoiding obstacles in restricted spaces. Sannapaneni et al.[13] discussed a learning from demonstration algorithm for a cloth folding manipulator, presented at the International Conference on Advances in Computing, Communications and Informatics (ICACCI), illustrating the application of learning algorithms in robotic manipulation tasks. Kumaar and TSB[14] introduced an intelligent lighting system using wireless sensor networks, presented in an arXiv preprint, emphasizing energy efficiency and automation in smart environments.

Kumaar and Sudarshan[15] presented mobile robot programming by demonstration at the International Conference on Emerging Trends in Engineering Technology, detailing methods for teaching robots tasks through human demonstration, showcasing advancements in human-robot interaction and autonomous task learning.

IV. System Overview

A. Webots Simulation Environment

In robotics research, simulation tools are important in the design and testing of algorithms before the actual testing in reality. Webots is a package developed by Cyberbotics Ltd., offering an integrated robotic simulation package that conforms to the IEEE Robotics and Automation Society standards for modeling robots. Researchers can use it to create, program, and test behavior in realistic 3D scenes with near-physics simulations. Webots has various models of robots, sensors, and programming languages, which helps develop new control algorithms for different scenarios.

Besides adhering to RAS standards, the best feature about Webots is its modular design and ease of use. The software is equipped with comprehensive libraries that enable researchers to develop arbitrary objects and environments for the simulation of certain tasks or real-world settings. Moreover, Webots provides some built-in mechanisms for data logging and analysis that assist a researcher in the efficient measurement and estimation of the performance of their control algorithms for robots in a simulated environment. All of these features combined make Webots a valuable asset for any researcher in any field of robotics.

B. Other Software Requirements

Python: Ensure Python is installed on your system, preferably version 3. x. TensorFlow: Install TensorFlow, an open-source deep learning library, which provides high-level APIs for building and training neural networks. Keras: Install Keras, a high-level neural networks API, which serves as the frontend for TensorFlow and simplifies the process of building deep learning models. NumPy: Install NumPy, a fundamental package for scientific computing with Python, which provides support for numerical operations and data manipulation. Pandas: Install Pandas, a powerful data analysis and manipulation library, which is used for handling structured data such as datasets and data frames. Matplotlib: Install Matplotlib, a plotting library for creating visualizations and graphs, which is used for displaying images and analyzing model performance. Pre-trained Models:

Download pre-trained convolutional neural network (CNN) models such as VGG16, ResNet, Densnet, and Xception. These models are trained on large datasets and can extract meaningful features from images, which are essential for image captioning tasks. Pre-trained models can be obtained from the Keras applications or TensorFlow Hub repositories.

V. Proposed Methodology

A. Data Collection

1) Environment Setup: To get going with this obstacle-avoiding drone, a simulation environment was set up in Webots, a professional robot simulation software. this study created a Webots project and set up a simulation world that this study are going to litter with a large number of different types of obstacles to resemble a real navigation scenario for the drone. this study then placed the pre-made model of the drone into the simulation environment. This setup actually provided the base for capturing the needed data to train this machine learning model.

2) Controller Development: To collect the data, this study have developed a specialized controller script for data collection, called ”datacollector.py”. It allowed the drone to move inside the simulation environment in user manual control while taking images from the drone camera. For every captured image, its corresponding command of the direction, such as forward, backward, left, and right, was stored. The images have been arranged in categorized directories according to these commands, hence making them easily labeled for the subsequent training process.

3) Execution: The simulation is run with the data collection controller, guiding the drone manually to collect a heterogeneous dataset. The dataset included images representing various cases of navigation, which will equip the model to handle all possible obstacle configurations and motion commands. This was an important step to have a robust training dataset representing navigation challenges of the real world.

B. Data Preparation

1) Image Processing: After the data collection process, the captured images were preprocessed. Every image was converted into grayscale in order to decrease computational complexity while keeping the essential features. After that, the images were resized to a standard size of 50x50 pixels, making the input uniform to the neural network. The resizing ensures the model receives input in a standard format that is crucial for accurate training and prediction.

2) Label Encoding: Concurrently with image processing, this study encoded the directional commands as categorical values. This is the process of mapping each navigation command to a unique integer, thus allowing the model to differentiate among various movement instructions during training. This encoding step is important in that it makes the model learn the relationship between the features of images and corresponding navigation instructions.

3) Data Storage: this study stored the pre-processed images and their corresponding labels as numpy arrays, namely DroneData.npy and DroneTarget.npy, to ensure that these data can be easily loaded and manipulated for a later stage of training. Moreover, using a numpy array optimized memory use and computational efficiency, necessary for large datasets.

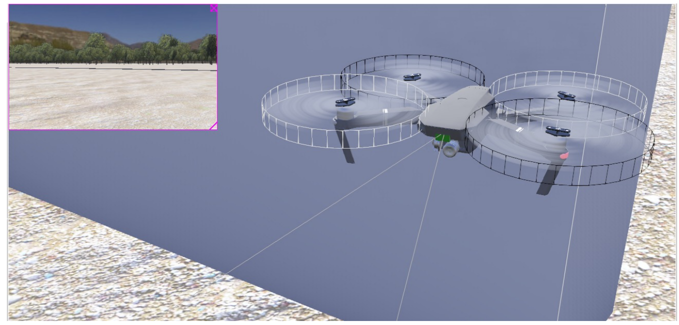

Fig. 1. Drone with camera used to collect dataset

C. Model Architecture

For the machine learning model, thus paper developed a Convolutional Neural Network (CNN) to process the input images and predict navigation commands. The architecture of the CNN used multiple convolutional layers for feature extraction and was followed by fully connected layers to classify the data. This architecture leverages the spatial hierarchies in the images in an attempt to solve visual recognition problems, like obstacle avoidance.

1) Training Process: Split the preprocessed dataset into training and validation sets for evaluation regarding model performance during training. The training of the CNN model was conducted on the training set, whereby the hyperparameters were tuned for optimization of performance. Advanced techniques, such as data augmentation and dropout, were implemented to make the model more robust against overfitting. The process involved training the model iteratively by updating the weights based on the error between the predicted and actual label, minimizing the loss function.

2) Model Saving: After achieving satisfactory performance on the validation set, this study saved the created and trained model, along with its weights, to disk as dronecnnmodel.h5 and dronecnnmodelweights.h5. This ensures that the trained model is loaded for use in making real-time predictions at the time of simulation. Saving the model also allows its reuse and further fine-tuning, if needed.

D. Integration and Testing

1) Main Controller Development: After achieving satisfactory performance on the validation set, this study saved the created and trained model, along with its weights, to disk as dronecnnmodel.h5 and dronecnnmodelweights.h5. This ensures that the trained model is loaded for use in making real-time predictions at the time of simulation. Saving the model also allows its reuse and further fine-tuning, if needed.Fig.1 shows us the example of it.

2) Obstacle Avoidance Logic: Obstacle avoidance core logic was implemented in the main controller. The controller adapted the movement of the drone based on model predictions by controlling the speed of motors. This included the mapping

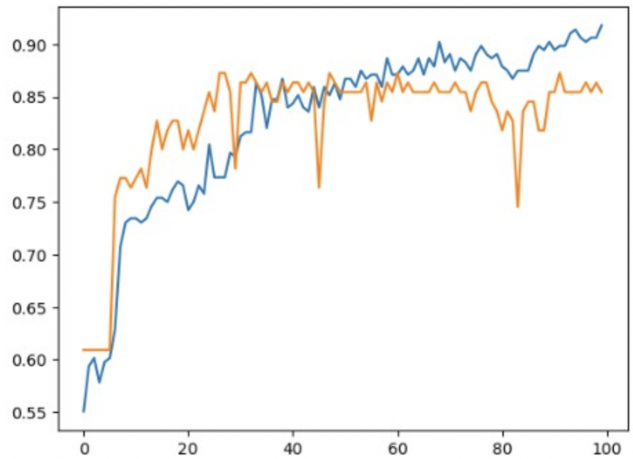

Fig. 2. CNN Model with acuuracy

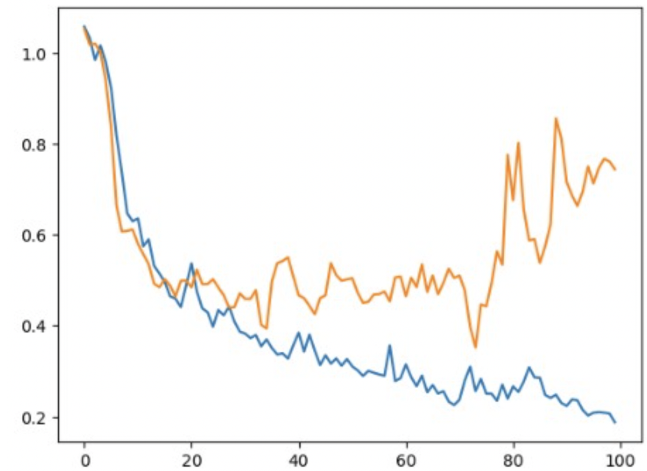

Fig. 3. model loss function

of these predicted navigation commands to concrete inputs of the motor so that the drone reacted correctly to the obstacle avoidance. The controller permanently monitored the drone's sensors and adjusted the navigation commands in real-time, thus being able to dynamically avoid obstacles.

VI. RESULTS AND DISCUSSION

The tests conducted for our AI-driven drone in a simulated Webots environment provided promising results with respect to autonomous navigation and obstacle avoidance. Data collection is a very important stage, as we had a very important, diversified, and comprehensive dataset of images, capturing situations and different directions of movement. Such a dataset formed the basis for training our machine learning model. Images were captured while moving the drone to the front, back, left, and right, creating a robust dataset that mirrored different navigational challenges.

In the data preparation stage, we resized, converted images to grayscale, and labeled them. This has given us consistency and enables the neural network to process the data with efficiency. It is a very critical step in standardizing the input data and is hence feasible by a convolutional neural network, to be trained. Then we split the labeled dataset into training and validation data, so it can learn and be evaluated on unseen data, which is really important to determine its generization capability.

Our model training was done on a CNN, which is very applicable for image recognition tasks. Thus, the training process iteratively tunes the model parameters to make the minimum prediction error. It learned what specific visual patterns from the images to associate with which movements. After several epochs, the model's accuracy improved, meaning it had the capability to predict the right movement direction based on visual inputs. The hyperparameters were tuned in the training phase, and techniques like data augmentation were applied in order to enhance the model's performance and prevent overfitting.This can be seen in Fig.2.

The trained model is integrated into a drone's control system within the Webots simulation environment during the final deployment phase. The drone captures images from its onboard camera continuously and feeds them into the trained model. The model will process these images in real-time, predicting the proper movement commands to navigate the environment and avoid obstacles. Drone movements are smooth and responsive to any conditions that may arise. Real-time predictions shown in this scenario demonstrate the ability of the model to generalize from training data to new, unseen environments.

The results can be seen from Fig.3 that the loss value decreases after 80 epochs this enlightened us to stop the epohchs at 100.

the results indicate that the decision making skills of the drone is 85.4 percent accurate.

In general, the results illustrated the use of a CNN for autonomous drone navigation. A drone navigated through the simulated environment, avoiding obstacles and making decisions based on visual input alone. This work showed the possibility of using computer vision and machine learning techniques together to enable autonomous behavior in drones. Future work can include expanding the dataset with more diverse scenarios, refining the model architecture, and testing the system in more complex and dynamic environments to enhance the navigation capabilities of the drone further.

VII. CONCLUSION

In this project, this study presented a comprehensive methodology for developing an obstacle-avoiding drone using machine learning techniques within the Webots simulation environment. this approach includes different stages: from data collection to preprocessing and to model training, integration, and testing of how to create a robust autonomous system for navigation.

With systematic data collection and preprocessing, this study had a very good and diverse set of drone navigation images with their respective direction commands. this study used the capabilities of Webots to simulate realistic navigation scenarios with obstacles; this ensured the representativeness and relevance of the dataset to real-world problems.

The core of this methodology was the design and training of a Convolutional Neural Network (CNN) model for the processing of input images to predict the navigation commands. this study made fine adjustments to the architecture of the model and hyperparameters, aided by techniques like data augmentation and dropout regularization, to achieve optimum performance and generalization.

The model, after proper training, was integrated into the Webots simulation environment by developing a main controller script. This enabled real-time processing of camera feed from the drone and the use of the trained CNN model to make decisions for navigation. The drone, through the dynamic adjustment of motor speeds based on model predictions, was able to successfully navigate through the environment with high accuracy in avoiding obstacles.

This results demonstrate the effectiveness of the developed system in autonomously navigating through complex environments while avoiding obstacles. Successful execution of the simulation proves the efficacy of this approach in leveraging machine learning techniques for drone navigation tasks.

VIII. FUTURE SCOPE

Further, in the enhancement of the obstacle avoidance capabilities of the drone, future research may be focused on advanced techniques in sensor fusion. While this current implementation uses data from the camera mostly, a fusion of this with data coming from other sensors, such as LIDAR or ultrasonic sensors, provides further depth perception and environment awareness. With the fusion of this multi-sensor data, the drone would be able to build an even better understanding of its surroundings for more accurate and reliable navigation in challenging environments. Another way to improve the drone's perception and estimation of obstacles includes adding depth estimation algorithms through stereo vision or structured light-based techniques.

REFERENCES

[1] Devos, A., Ebeid, E. and Manoonpong, P., 2018, August. Development of autonomous drones for adaptive obstacle avoidance in real world environments. In 2018 21st Euromicro conference on digital system design (DSD) (pp. 707-710). IEEE.

[2] Woods, A.C. and La, H.M., 2015. Dynamic target tracking and obstacle avoidance using a drone. In Advances in Visual Computing: 11th International Symposium, ISVC 2015, Las Vegas, NV, USA, December 14-16, 2015, Proceedings, Part I 11 (pp. 857-866). Springer International Publishing. Vancouver

[3] Liang, Q., Wang, Z., Yin, Y., Xiong, W., Zhang, J. and Yang, Z., 2023. Autonomous aerial obstacle avoidance using LiDAR sensor fusion. Plos one, 18(6), p.e0287177.

[4] Park, J. and Cho, N., 2020. Collision avoidance of hexacopter UAV based on LiDAR data in dynamic environment. Remote Sensing, 12(6), p.975.

[5] Ramasamy, S., Sabatini, R., Gardi, A. and Liu, J., 2016. LIDAR obstacle warning and avoidance system for unmanned aerial vehicle sense-and-avoid. Aerospace Science and Technology, 55, pp.344-358.

[6] Santos, M.C., Santana, L.V., Brandao, A.S. and Sarcinelli-Filho, M., 2015, June. UAV obstacle avoidance using RGB-D system. In 2015 International Conference on Unmanned Aircraft Systems (ICUAS) (pp. 312-319). IEEE.

[7] Ricardo, S., Bein, D. and Panagadan, A., 2017, January. Low-cost, real-time obstacle avoidance for mobile robots. In 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 1-7). IEEE.

[8] Aldao, E., González-deSantos, L.M., Michinel, H. and González-Jorge, H., 2022. Uav obstacle avoidance algorithm to navigate in dynamic building environments. Drones, 6(1), p.16.

[9] Robinson, C. and Jang, G., Developing a 2D LiDAR Collision Avoidance Algorithm for a Quadcopter.

[10] Gyenes, Z., Bölöni, L. and Szádeczky-Kardoss, E.G., 2023. Can genetic algorithms be used for real-time obstacle avoidance for lidar-equipped mobile robots?. Sensors, 23(6), p.3039.

[11] Tu, G.T. and Juang, J.G., 2023, January. UAV path planning and obstacle avoidance based on reinforcement learning in 3d environments. In Actuators (Vol. 12, No. 2, p. 57). MDPI.

[12] Dong, T., Zhang, Y., Xiao, Q. and Huang, Y., 2023. The Control Method of Autonomous Flight Avoidance Barriers of UAVs in Confined Environments. Sensors, 23(13), p.5896.

[13] Sannapaneni, B., Shaswat, M. and Kumaar, A.N., 2017, September. Learning from demonstration algorithm for cloth folding manipulator. In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1339-1398). IEEE.

[14] Kumaar, A.A. and TSB, S., 2010. Intelligent lighting system using wireless sensor networks. arXiv preprint arXiv:1101.0203.

[15] Kumaar, A.N. and Sudarshan, T.S.B., 2011, November. Mobile robot programming by demonstration. In 2011 Fourth International Conference on Emerging Trends in Engineering Technology (pp. 206-209). IEEE. Vancouver