Development of An Intelligent Mobile Robot System for Object and Obstacle Detection

Osita Miracle Nwakeze, Ogochukwu Okeke & Ike Mgbeafulike

Abstract

This study presents the development of an intelligent autonomous mobile robot system for object detection and classification using the combination of Faster Regional-Convolutional Neural Network (R-CNN) with a Double Deep Q-Learning Network (DDQN) for decision-making in a dynamic environment. The system adopts the Open ImageV5 data-set for testing and training, improving model performance through the use of strong preprocessing and augmentation methods. While the DDQN allows for optimum path planning and manoeuvring decisions in stochastic mobility circumstances, the faster R-CNN guarantees excellent detection accuracy by improving object categorisation and bounding box predictions. The suggested concept was put into practice and assessed in a gaming environment that mimics actual circumstances. The results presents a consistent performance in braking and stopping distances, a processing speed of 5–10 frames per second, and good detection accuracy (90–95%). The system's generalisability and dependability were confirmed using a 5-fold cross-validation approach, which also confirmed that it is appropriate for real-time applications. With applications in robotic systems, security, and environmental monitoring, this study demonstrates the possibility of combining deep learning with reinforcement learning for autonomous navigation.

Keywords: Autonomous Mobile Robot, Faster R-CNN, Double Deep Q-Learning Network (DDQN), Object Detection, Path Planning, Reinforcement Learning

1. INTRODUCTION

U The basis for further sophisticated behaviours carried out by the artificial agent (robot entities or software algorithms that can autonomously execute orders and actions in real or simulated settings) is mobile robot navigation, or perceptual recognition of the surroundings. In earlier conventional navigation technologies, path-planning duties were accomplished by mobile robot

navigation algorithms using data from external sensors (lasers, cameras, radar, etc.). Numerous industrial applications have demonstrated the widespread usage of this classic technology. The Global Navigation Satellite System (GNSS) is one example of how it has also been very successful (Gyagenda et al., 2022). However, the aforementioned technologies continue to have serious drawbacks and are inapplicable; they cannot be used in contexts that are complicated, unfamiliar, and constantly changing. The map-based navigation method requires the environment to be surveyed and mapped beforehand, which takes a lot of effort and raises the bar for environment modelling accuracy. However, classical navigation has significant challenges in accurately simulating complex and dynamic settings. Similarly, the resilience of the navigation algorithm is also impacted by the accumulation of noise during the modelling process and the lack of sensor precision (Li, 2023).

In order to overcome the aforementioned challenge, several researchers are dedicated to integrating Simultaneous Localisation and Mapping (SLAM) with conventional navigation systems in order to generate unfamiliar and intricate settings (Durrant-Whyte and Bailey, 2006). The mobile robot will develop an efficient route to reach its destination and finish the navigation job based on the positioning algorithm that anchors its real-time location.

A potential substitute strategy for navigating dynamic, uncertain, and partially visible situations is provided by learning techniques like Deep Reinforcement Learning (DRL). DRL is a machine learning method where an agent learns a behavioral strategy that maximises the numerical reward it receives from its surroundings through trial and error. In order to improve its policy, the agent "learns" through a certain number of training events in which it engages with its surroundings. In Atari games and the game of Go, DRL algorithms have demonstrated remarkable ability in learning Markov decision processes (Mnih et al., 2015, 2016; Silver et al., 2016, 2017).

The autonomous navigation problem has recently been referred to as a Markov Decision Process (MDP) using DRL. Using Deep Dyna-Q learning, an agent represented by a point in 2D space was simulated for ER evacuation (Zhang et al., 2021). Dyna-Q is a DRL method that combines model-free Q-Learning with model-based reinforcement learning to learn the optimal action value function (Sutton and Barto, 2018). The authors trained a Deep Q-Network (DQN) to predict the value of moving in one of eight different directions based on the agent's location inside the evacuation scenario. The algorithm was able to navigate from any starting point to the destination in a time-efficient manner in obstacle-containing situations with favourable and aversion zones. Robotic navigation tasks in static environments where the robot's exact location is always known are a good fit for this learning

scenario. However, for a robotic system, it is frequently unreasonable to assume a static environment, accurate localisation, and complete knowledge of obstacle configurations (Blum et al., 2022).

In this study, the Reinforcement Learning method for the agent's environment exploration is Double Deep Q-Learning (DDQL) (Sutton and Barto, 2011). The Double Q-Learning addresses the issue of overestimation of the Q-value in basic Q-Learning, whilst the Q-Learning algorithm thrives on determining appropriate measures to decide in a given scenario (Van-Hasselt et al., 2016). The activities that are tried and the states that are investigated determine how accurate the Q-values are. As a result, at the start of the training, an agent lacks the knowledge they need to decide what to do. Selecting the optimal course of action based on the largest Q-value may be noisy and result in false positives. In order to isolate action selection from target Q-value production, the Double Deep Q-Learning Network employs two distinct networks, namely the Deep Q Network and the Target Network. As a result, DDQN considerably reduces the overestimation of Q-values, which helps an agent learn steadily through faster training and demonstrates the superiority of this approach over alternative learning algorithms (Bin-Issa et al., 2021). Therefore, by selecting the largest Q-value while exploring the same environment, DDQN demonstrates to be appropriate for an autonomous agent to make judgements for optimum traversal. The epsilon greedy strategy is the method by which the agent continuously establishes the greatest Q-value for navigation (Sutton and Barto, 2011).

While navigating, this activity also include identifying and categorising impediments. Through the sensor, it gathers information from barriers on rocky, uneven, and lumpy terrain. To put the aforementioned suggestion into practice, we primarily employed a visual sensor. The agent receives the sensory input and makes a choice depending on the conditions it has been given. The prototype mainly uses Faster R-CNN (Erhan et al., 2014) as the mobile robot attempts to detect and identify things while travelling. At the moment, Faster R-CNN is a notable object classification algorithm. To find region suggestions, the R-CNN and Fast R-CNN algorithms use a selective search technique. Faster R-CNN (Ren et al., 2016) is superior than its predecessors since it does not use the selective search technique and instead learns the region proposals. It is faster and more appropriate for real-time object detection as it uses a convolutional network for both region suggestion and object detection.

2. METHODOLOGY

Deep Reinforcement Learning has been used in a number of highway decision-making strategies (Liao et al., 2020; Nageshrao et al., 2019); in this

study, the deep Q-Learning approach is combined with the Faster R-CNN method to enable an autonomous agent to detect and steer clear of obstacles while travelling. The fusion of these two techniques for autonomous manoeuvre combines the advantages of these two approaches in autonomous mobile robot navigation, even though the deep Q-Learning and Faster R-CNN algorithms have shown success for autonomous mobility strategy and object classification, respectively. The suggested reinforcement learning-based autonomous mobile robot concept is depicted in Figure 1. In order for an autonomous mobile robot to make manoeuvring decisions while categorising and avoiding objects and obstacles along its path, the suggested model combines the Double Deep Q-Learning Network (DDQN) and Faster R-CNN. A gaming environment that closely resembles the real-world situation is used to test the proposed approach.

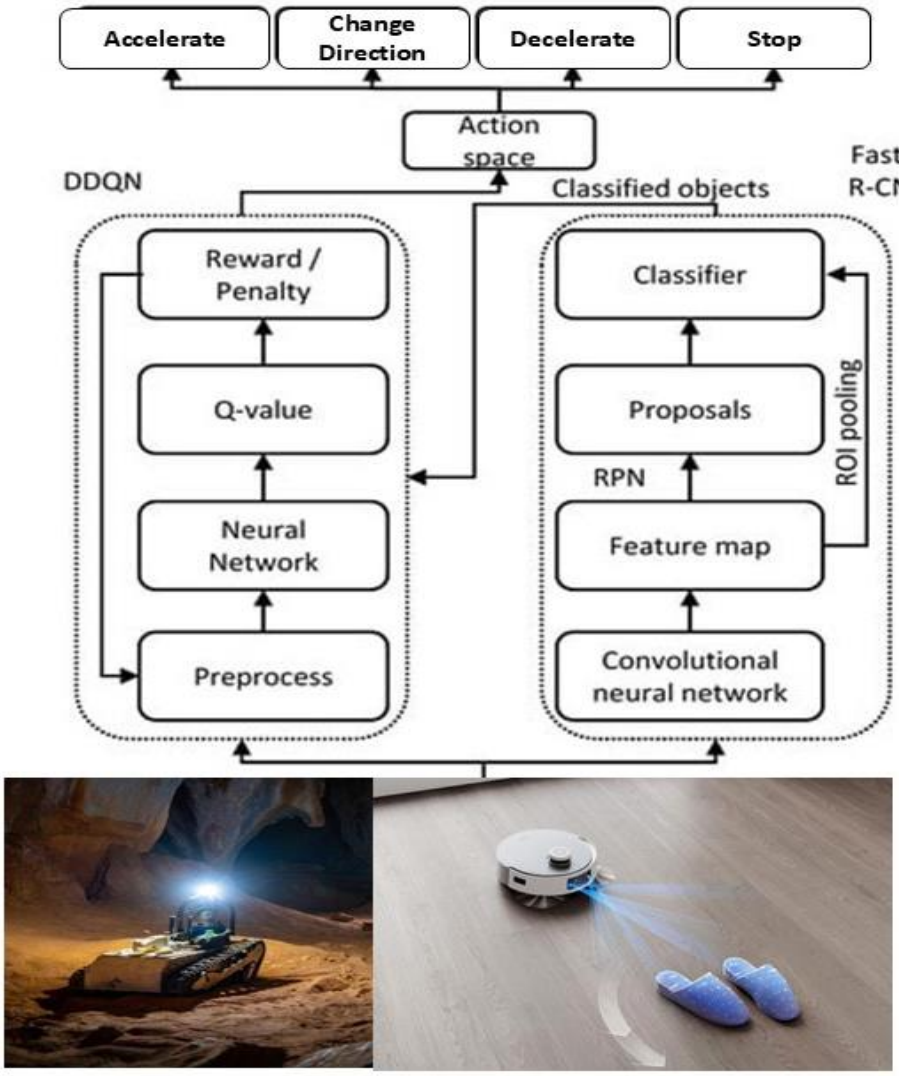

Figure 1: Architecture of the Proposed System

2.1 Data Acquisition

Experiments are carried out using a massive data collection called Open Image V5 (Kuznetsova et al., 2020). It is a free index that is created by Google. It has 600 box-able object types and is enhanced with clear visuals. Bounding boxes, visual relationships, object segmentation, complete validation (40,520 images), and test sets are all included in the 1,732,042 training photos. Our Object Classifier does not, however, require all 600 classes. Using "OIDv4_ToolKit," a number of unique classes such as people, shoes, tables, chairs, walls, etc. are distinguished from these 600 classes. Our model is trained using 8335 extracted photos, with at least 1000 images in each class.A total of 2360 photographs are taken from the data set for testing, with at least 300 images in each class. These photos' labels are then compiled into a.XML file.

2.2 Data Preprocessing

As seen in Figure 2, eight vision sensors, all of which are indicated in blue, collect the pictures of the environment. The model will not be ready with the sky and the front sections of the mobile robot because the output images from vision sensors are altered. Those images are resized to 160 x 320 pixels (3 YUV channels) in accordance with the Google Colab concept. These images are standardised, with 1.0 subtracted from the picture information that was isolated by 127.5. This is to preserve congestion and improve the performance of slopes, as stated in the Model Scheme section.

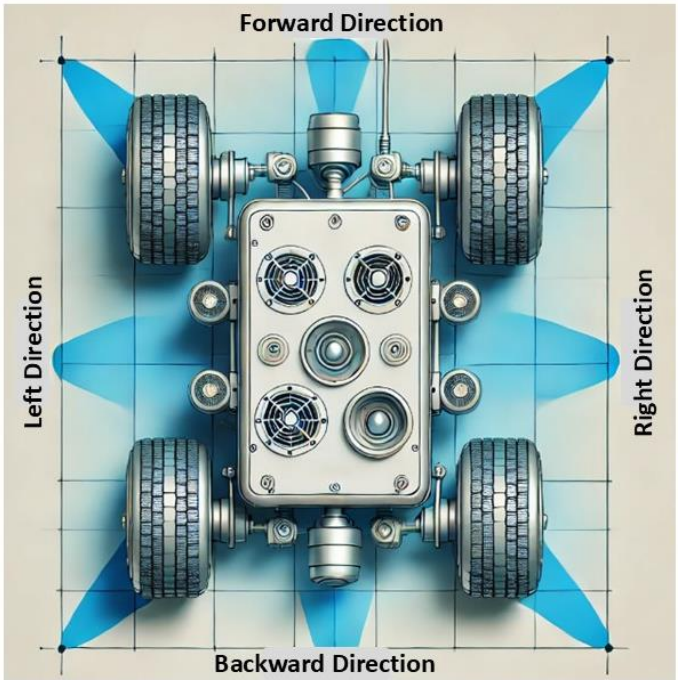

Figure 2: Image of the Mobile Robot with vision sensors

2.3 Faster R-CNN for Object Classification

In a unique manner, the R-CNN applies a neural classifier to a distantly featured input image by trimming its calculated box proposition. It is more expensive than other methods since it necessitates a large number of crops, which results in overlap calculations. Nevertheless, this issue can be reduced by using Fast R-CNN. Fast R-CNN smoothes the highlighting extraction process, cuts the image from a middle layer, and feeds the entire image to a feature extractor. R-CNN and Fast R-CNN formerly needed an external proposal generator. The neural network can now perform the same task with more efficiency. It is typical for the image to include boxes that beat to one another; these are referred to as "anchors." To keep an outline, scales and perspective proportions are necessary. A model is then created to predict each anchor by (i) estimating a discrete class of unique anchors and (ii) accumulating counterweight forecasts that are necessary to move the anchor in order to put it in the ground truth bounding box.

Anchor identification has good effects on computation and accuracy. The anchors from the data-set are now calculated by tiling boxes over the picture according to different sizes and ratios, as opposed to utilising clustered ground-truth boxes. The key benefit of employing a network is that the prediction may be created using the picture's tiled indications and shared parameters. As a conventional sliding window technique, it is notable (Ren et al., 2016; Szegedy et al., 2014). The algorithm for the Faster R-CNN system is presented in Algorithm 1 as:

Algorithm 1: Faster R-CNN Algorithm

i. Start

ii. Input Image

iii. Image Processing (Resizing and Normalization of pixel values)

iv. Feature Extraction using pre-trained CNN

v. Region Proposal Network for Feature Mapping

vi. For each position in the feature map, generate multiple anchor boxes of different scales and aspect ratios

vii. Classify each anchor box as foreground or background

viii. Adjustment of bounding box regression for anchor boxes for better localization

ix. Region of Interest (ROI) pooling

x. Use ROI to convert different-sized proposals into fixed sizes

xi. Object classification for assigning class label to the proposal

xii. Bounding box regression for further refinement of bounding box coordinates

xiii. Eliminate duplicate and overlapping bounding boxes using Non-Maximum Suppression (NMS)

xiv. Apply NMS for retainment of the most confident predictions for each object

xv. Output classified objects with bounding box coordinates

xvi. Stop

2.4 Distributional Agent for Autonomous Mobile Robot using Double Deep Q-Learning Network (DDQN)

As an extension of his previous proposal, which relates to Deep Q Network (DQN) (Hasselt, 2010), V. Hasselt developed the concept of DDQN (Van-Hasselt et al., 2016). There are only few Q-Learning-based techniques, and the DQN is one of them. These Q-Learning-based methods have overestimation problems due to estimate mistakes. Overestimation leads to overoptimistic fee evaluation and performance degradation. In any event, the DDQN approach not only reduces the too optimistic revere assessment, but it also provides preferable performance over DQN on a few simulated accustoms. DDQN focusses on two Q-functions as a motivator and disengages from excerpt and interpretation measurements. According to Bin-Issa et al. (2021), the objective regard states of DQN and DDQN are:

$$y^{DQN} = R_t + Y_{A_{t+1}} maxQ(S_{t+1}, A_{t+1}; \theta^-) \quad (1)$$

$$y^{DDQN} = R_t + YQ\left(S_{t+1}, argmazQ_{A_{t+1}}(S_{t+1}, A_{t+1}; \theta^-); \theta^-\right) \quad (2)$$

2.5 Markov Decision Process for Path Circulation

In this study, the secure path for self-governing mobility is expressed by the Markov Decision Process. In every step, the actor chooses his own action, and a prize is promptly awarded for that choice. As previously mentioned, the Markov Decision Process (MDP) is chronicled by the tuple {S, P, A, R, γ}. A brief overview of MDP is provided below for easier comprehension (Bin-Issa et al., 2021).

- s e S denotes the constrained state space that can hold a greyscale image from the actor's vision sensors.

- S × A × S → [0, 1] is the evolution behaviour defined by P(s 0ks a).
- A is a specific reaction area that an actor can use.
- R(s, a): S × A → R specifies reward behaviour.
- The rebate element is described by γ, where γ → [0,1] for the postponed incentive.

By using Deep Neural Networks, MDP states s e S may be used for high dimensional perceptions (Alam et al., 2020). Eight vision sensors are positioned at the front of Figure 2, which refers to the idea of all-encompassing inclusiveness.

There are five specific actions for the autonomous mobility actor. Forward, left, right, stop, and deceleration are all included in the definite reaction region A. 5 kph is added to or subtracted from the running actor acceleration for forward and deceleration. The range of the actor's acceleration is between 30 and 80 kph. In order to maintain a safe distance from the front mobile robot, the actor automatically modifies the acceleration for mobile robots in a certain separation. The "stop" action occurs instantly when the mobile robot in front of us abruptly slows down or when another mobile robot abruptly cuts in front of our representative mobile robot.

3. ARCHITECTURE OF THE PROPOSED MODEL

The actor's main goal, π(a]s, is to plan the discerning state, S, and take the related action on the response region, A. In a scenario with stochastic mobility, the entire activity will be driven. In any case, the model must meet certain requirements in order to do this planning: (i) focus and capture large characteristics from the images captured by three vision sensors; and (ii) evaluate the climate's typical arbitraryness in order to select a certain activity.

To meet the primary requirement, the network must identify spatio-balanced data recovered from vision sensors. This cycle is directed by using CNN. In contrast to images, CNN is renowned for its ability to extract spatial characteristics. Furthermore, two-dimensional three convolutional layers are used to refine large spatial vision sensor images into ocular component vectors. Furthermore, the DDQN technique may be used to satisfy the following scenario. This method is used for stochastic mobility circumstances. The totaFull creates a restorative circulation for each action. complete linked layer with θ's help. The longing of quantiles, $\sum i\ qi\theta i(s,a)$, can be used to evaluate the Q(s,a) choice. Figure 3 shows the flow diagram of the suggested DDQN scheme for the suggested method. This suggested network is constructed using Keras.

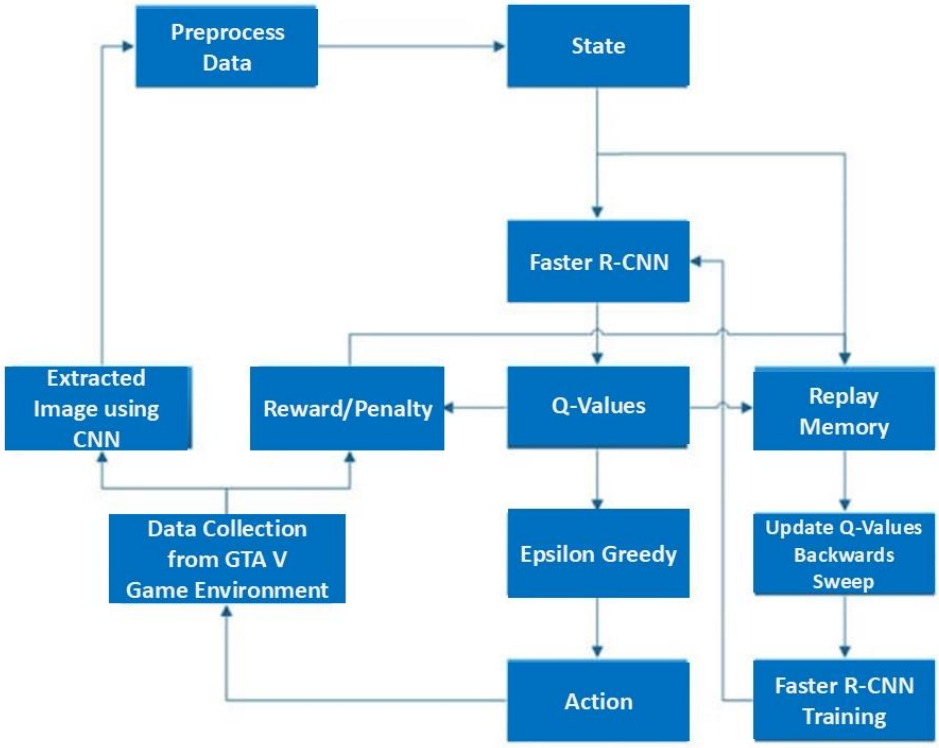

Figure 3: The Architectural Flow Diagram of the Proposed System

3.1 Model Training

The gaming environment and the images gathered from the agent's cameras have been used for agent training. The study's presumptions include the use of recognised items for categorisation and sunshine for good eyesight. But for diversity, we used the Python generator in conjunction with the associated growth process to generate an infinite number of images. In addition to being utilised indiscriminately, these photographs have had their properties altered to provide the agent with a variety of situations, such as flipping images from left to right or altering their splendour and shadows. The following are arbitrary detail modifications (Bin-Issa et al., 2021).

- Arbitrarily select right, centre or left picture.
- Steering angle is adapted by +0.2 for left picture.
- Steering angle is adapted by −0.2 for right picture.
- Arbitrarily flip picture right/left.
- Arbitrarily convert picture horizontally with steering angle accommodated (0.002 per pixel shift).
- Arbitrarily convert picture vertically.

- Arbitrarily added shadows.
- Arbitrarily changing picture splendor (lighter or more obscure).

It is helpful to prepare the recuperating mobility condition by using the left and right photos. When dealing with difficult bends, the level interpretation is useful.

3.2 Hyperparameters Optimization

The network is built using the Colab paradigm from Google. Google Colab uses it to carry out an autonomous test from beginning to end. In general, the Google Colab model is preserved. Inconsistent use of the deep convolutional network can explain supervised image distribution or relapse problems. As a result, altering the preparatory images to portray the greatest result is the main focus. In any event, essential adjustments have been made to avoid over-fitting and to include impartiality in order to construct the prediction accurately in order to obtain the best result. Additionally, the model has been updated to include the associated acclimation.

- To make gradients easier to work with and avoid saturation, the lambda layer is accustomed to standardising input images.
- To prevent over-fitting, an additional dropout layer is added after the convolution layers.
- In order to ensure linearity, ReLU was then run for actuation capacity.

The network is prepared for greater precision using the Adam Optimiser at a $1 \times 10{-}5$ learning rate, epsilon 0.0001, and 32 sets of mini-batches. All data sources are standardised into [1,−1], and Xavier Initialiser has been used to initialise network demands. The loss function has been evaluated using mean squared errors in order to achieve the accuracy of the guidance plot forecast for each image.

4. SYSTEM IMPLEMENTATION

Setting up the development environment is the first step in implementing an intelligent computer vision-based environmental monitoring and people detection system using Google Colab. This entails obtaining the dataset, turning on GPU acceleration for speedier computations, and loading necessary libraries like TensorFlow, OpenCV, and NumPy. Using programs like OIDv4, the dataset may be created by obtaining pertinent picture classes like "Person" or "Vehicle." To guarantee an effective learning process, images are pre-processed by shrinking to uniform dimensions, normalising their pixel values, and dividing them into training, validation, and testing sets.

A Faster R-CNN architecture, which integrates object categorisation, region proposal, and feature extraction, is used to construct the model. The foundation for feature extraction from input photos is a pre-trained VGG16 network. By creating bounding boxes and fine-tuning their coordinates, the Region Proposal Network (RPN) finds possible object areas. The classification layer further processes the RPN's outputs to forecast precise bounding box positions and name items that are discovered. A Double Deep Q-Learning Network (DDQN) is used to optimise the robot's activities in the surveillance environment, guaranteeing flexibility to changing situations and improving the system's decision-making.

Lastly, the system is trained using reinforcement learning methods for the DDQN and tagged pictures for the Faster R-CNN. The Faster R-CNN is trained with mean squared error for bounding box regression and categorical cross-entropy for classification. As the DDQN agent engages with its surroundings, it minimises exploration and maximises cumulative rewards, thereby enhancing its performance. The accuracy and efficacy of the system are assessed using unseen data after training. The end result is a strong surveillance system that can identify and categorise people or things in real time, opening the door for security and environmental monitoring applications.

5. SYSTEM RESULTS

The system's performance across important metrics shows promise, demonstrating its capacity to make decisions and conduct surveillance in real time. The system may function well in dynamic contexts where quick detection and reaction are crucial because to its processing speed of 5–10 frames per second (FPS). This performance is the outcome of using Google Colab's GPU acceleration and optimisation algorithms. Faster R-CNN's integration with the DDQN significantly improves the system's responsiveness by enabling it to make prompt judgements to adjust to changing circumstances in addition to properly detecting objects. This is especially helpful for jobs involving human identification and autonomous navigation.

The system's excellent success rate of 90–95% in terms of detection accuracy demonstrates its dependability in recognising barriers and people. A key factor in lowering false positives and negatives is the application of the Faster R-CNN model, which was trained on the Open Image V5 dataset with substantial data augmentation. Robustness is guaranteed even in congested situations because to the capacity to categorise several item types and improve bounding box predictions. The outcomes confirm that the technology is appropriate for high-precision applications including environmental

surveillance and security monitoring.

A 5-fold cross-validation approach was used to further validate the system's output. Five equal subsets of the dataset were created, four of which were utilised for training and the fifth as a test set. With only a slight variation, the average detection accuracy across the folds showed that the algorithm performs effectively when applied to unknown data. The 5-fold validation results table for total stopping distance, braking distance, and detection accuracy is shown in Table 1.

Table 1: Performance Result Validation

Fold	Detection Accuracy (%)	Avg Braking Distance (meters)	Avg Total Stopping Distance (meters)
1	93.2	6.5	8.7
2	92.8	6.3	8.5
3	94.0	6.7	9.0
4	91.9	6.4	8.6
5	93.7	6.6	8.8
Avg	**93.1**	**6.5**	**8.7**

The consistency and dependability of the system are validated in Table 1. The system's performance in a variety of situations is validated by the detection accuracy, which stays over 91% throughout all folds, and the low fluctuation in the braking and total stopping lengths. This comprehensive assessment guarantees that the system can be implemented successfully in practical situations.

6. CONCLUSION

To sum up, this study effectively illustrates the creation and deployment of an intelligent autonomous mobile robot system that combines the Double Deep Q-Learning Network (DDQN) for decision-making with Faster R-CNN for object identification. Utilising these cutting-edge techniques, the system achieves fast reaction times and excellent detection accuracy (90–95%), allowing for dependable real-time navigation and obstacle avoidance. Eight vision sensors are integrated to provide thorough environmental awareness, and strong preprocessing and data augmentation methods enhance model

performance under demanding and changing circumstances. Additionally, the system's adaptability to stochastic mobility conditions confirms that it is suitable for use in environmental surveillance and autonomous navigation.

The 5-fold cross-validation findings show that the system is reliable and generalisable, with safe braking and stopping distances and consistent detection accuracy across a range of test circumstances. These results demonstrate how well the model can manage a variety of real-world scenarios while maintaining efficiency and safety. The study lays the groundwork for future investigations into enhancing system durability, scalability, and deployment in increasingly complex situations in addition to highlighting the possibilities of deep learning and reinforcement learning in autonomous systems. This study advances intelligent autonomous systems for useful, real-world security, surveillance, and navigation applications.

REFERENCES

Alam, M., Kwon, K. C., Abbass, M. Y., Imtiaz, S. M., & Kim, N. (2020). Trajectory-based air-writing recognition using deep neural network and depth sensor. Sensors, 20(2), 376.

Bin-Issa, R., Das, M., Rahman, M. S., Barua, M., Rhaman, M. K., Ripon, K. S. N., & Alam, M. G. R. (2021). Double deep Q-learning and Faster R-CNN-based autonomous vehicle navigation and obstacle avoidance in dynamic environment. Sensors, 21(4), 1468. https://doi.org/10.3390/s21041468

Blum, P., Crowley, P., & Lykotrafitis, G. (2022). Vision-based navigation and obstacle avoidance via deep reinforcement learning.

Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localization and mapping: Part I. IEEE Robotics & Automation Magazine, 13(2), 99–110.

Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. (2014). Scalable object detection using deep neural networks. In Proceedings of the CVPR (pp. 24–27). Columbus, OH, USA.

Gyagenda, N., Hatilima, J. V., Roth, H., & Zhmud, V. (2022). A review of GNSS-independent UAV navigation techniques. Robotics and Autonomous Systems, 104069.

Hasselt, H. V. (2010). Double Q-learning. In Proceedings of the Advances in Neural Information Processing Systems (pp. 2613–2621). Vancouver, BC, Canada.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Duerig, T., & Ferrari, V. (2020). The Open Images Dataset V4. International Journal of Computer Vision, 128(7), 1956–1981.

Li, H. (2023). Mobile robot navigation based on deep reinforcement learning: A brief review. Journal of Physics: Conference Series, 2649, 012027. https://doi.org/10.1088/1742-6596/2649/1/012027

Liao, J., Liu, T., Tang, X., Mu, X., Huang, B., & Cao, D. (2020). Decision-making strategy on highway for autonomous vehicles using deep reinforcement learning. IEEE Access, 8, 177804–177814. https://doi.org/10.1109/ACCESS.2020.3018117

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. Nature, 518, 529–533. https://doi.org/10.1038/nature14236

Nageshrao, S., Tseng, H. E., & Filev, D. (2019). Autonomous highway mobility using deep reinforcement learning. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC) (pp. 2326–2331). Bari, Italy.

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., et al. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529, 484–489. https://doi.org/10.1038/nature16961

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., et al. (2017). Mastering the game of Go without human knowledge. Nature, 550, 354–359. https://doi.org/10.1038/nature24270

Sutton, R. S., & Barto, A. G. (2011). Reinforcement learning: An introduction. MIT Press.

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.

Szegedy, C., Reed, S., Erhan, D., Anguelov, D., & Ioffe, S. (2014). Scalable, high-quality object detection. arXiv preprint, arXiv:1412.1441.

Van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double Q-learning. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (pp. 2326–2331). Phoenix, AZ, USA.

Zhang, Y., Chai, Z., & Lykotrafitis, G. (2021). Deep reinforcement learning with a particle dynamics environment applied to emergency evacuation of a room with obstacles. Physica A: Statistical Mechanics and its Applications, 571, 125845. https://doi.org/10.1016/j.physa.2021.125845

Author Information: Osita Miracle Nwakeze is affiliated with Department of Computer Science, Chukwuemeka Odumegwu Ojukwu University, Uli, Anambra State, Nigeria. *Email*: ma.nwakeze@coou.edu.ng

Ogochukwu Okeke is of the Department of Computer Science, Chukwuemeka Odumegwu Ojukwu University, Uli, Anambra State, Nigeria. *Email*: ogoookeke@yahoo.com

Ike Mgbeafulike is affiliated with Department of Computer Science, Chukwuemeka Odumegwu Ojukwu University, Uli, Anambra State, Nigeria. *Email*: ij.mgbeafulike@coou.edu.ng

CITING THIS ARTICLE

APA

Nwakeze, O. M., Okeke, O., & Mgbeafulike, I. (2025). Development of An Intelligent Mobile Robot System for Object and Obstacle Detection. *Global Online Journal of Academic Research* (*GOJAR*), 4(2), 7-21. https://klamidas.com/gojar-v4n2-2025-01/.

MLA

Nwakeze, Osita Miracle, Okeke, Ogochukwu, & Mgbeafulike, Ike. "Development of An Intelligent Mobile Robot System for Object and Obstacle Detection". *Global Online Journal of Academic Research* (*GOJAR*), vol. 4, no. 2, 2025, pp. 7-21. https://klamidas.com/gojar-v4n2-2025-01/.