Obstacle Avoidance Robot Using Convolutional Neural Network

Naseem Ahmed Abdo
School of Engineering
Asia Pacific University of Technology and innovation (APU)
Kuala Lumpur, Malaysia
TP041426@mail.apu.edu.my

Rosli Bin Yusop
School of Engineering
Asia Pacific University of Technology and innovation (APU)
Kuala Lumpur, Malaysia
rosli.yusop@@apu.edu.my

Raed Abdulla
School of Engineering
Asia Pacific University of Technology and innovation (APU)
Kuala Lumpur, Malaysia
raed@staffemail.apu.edu.my

***Abstract*-The aim of the project is to design an obstacle avoidance system by using neural network. The main functionality of the project is to predict the object detected and to find a path of avoidance the detected object. Due to the increase in autonomous vehicle centered around machine learning technology, expensive system which incorporates multiple sensors for the obstacle avoidance and object detection are made. To fill this gap, the project is designed to implement an object detection and obstacle avoidance system using a camera as the main component for detection. Jetson nano board is used as the main computer, NoirV2 camera is used as the main vision sensor, 3d printed structure for the body, two dc motor with a I2c based motor driver. SSD MobileNet v2 is used as the model for object detection, Jetson inference is the training guide which is used for the obstacle avoidance and object detection which is optimized to work with Jetson nano board providing a faster detection and fps. The project includes a web based Graphical user interface to control the robot and to monitor it. The project will solve the issues of expensive system and requirement of multiple sensors for object detection and obstacle avoidance.**

Keywords— navigation robot, mazes, breadth first search and depth first search, ultrasonic sensor

I. Introduction

With Different mobile robots have been widely used in various fields, such as life services, industrial services, with the continual advancement of science and technology. Some of the sectors includes production, schooling, leisure, and military room, etc. In different environments, such as rescue and mining, exploration in an unfamiliar environment is still a fundamental problem for mobile robots. Usually, the robot requires complex reasoning about the barriers and topological mapping of human-designed environments based on vision or depth sensor information. Vehicles need to sense and avoid 3D obstacles in real time to maneuver autonomously. To create a 3D environment map, various range sensors such as laser, stereo, and RGB cameras could be used. Due to every increasing research on reducing accidents while increasing productivity autonomous navigation is already present in many industries and it is the future of mobility. With the growth of machine learning, especially deep learning, it has become a study hotspot in recent years that the robot avoids obstacles through self-learning. Deep learning is an approach to end-to-end learning, which is a partnership between input and output through a deep learning network. That is, by applying a lot of data into the algorithm, the machine automatically learns the characteristics of the data.

The use of CNN for Autonomous Underwater Vehicles (AUVs) based on automatic obstacle avoidance was demonstrated by (Gaya et al 2016). Although the model is not evaluated in real time and the network is not end-to-end as it consists of intermediate procedures specifications. The major disadvantage of the existed obstacle avoidance systems is that the system requires multiple sensors, mostly combination of distance sensor such as ultrasonic or lidar sensors and vision-based sensor. An efficient algorithm is required for the communication between each sensor for a particular task. Along with these sensor complexities, hardware complexity adds up. To process huge amount of data requires an expensive processor, any discrepancy in the communication can results in fatal accidents. (Bhagat et al., 2016) proposed an obstacle avoidance robotic vehicle using ultrasonic sensor. Proposed system has shown that the built-in intelligent algorithm was able to avoid obstacle, the micro-controller redirects the robot to drive in an alternative direction, depending on the input signal obtained, by actuating the motors that are interconnected to it via a motor driver. (Liu et al., 2017) proposed a CNN- based vision model for obstacle avoidance of mobile robot. Proposed system was able to avoid the objects using end to end learning-based CNN model. The study mentioned above was done with controlled environment and for specific tasks, which bounds the usage of the developed algorithm for specific purpose. The former proposed system did not use the vision system which bounds the system to be used for very limited and well-structured environment, while the latter required expensive hardware to train the models while having no object identification algorithm.

Liu et al.(2017) proposed a CNN based vision model for obstacle avoidance of mobile robot. The proposed system present CNN based end-to-end learning model, which only takes the raw image obtained from the camera as an input. And the process converts the raw pixels to steering commands directly, commands for left turn, right turn and forward. A human remotely operated mobile robot collected training data that was manipulated to explore without colliding into obstacles in a structural environment. While Lee et al. (2016) proposed a monocular vision sensor-based obstacle detection algorithm. The system was designed to reduce the computational power required for the obstacle detection. Each individual image pixel is classified as belonging to either an obstacle or the floor at the bottom of the interest area. Although traditional methods rely on point tracking for obstacle detection geometric indications, the inverse perspective mapping (IPM) approach is used by the proposed algorithm. This research uses images taken from a forward-viewing mono camera as sensory inputs and odometry from robot wheel encoders and gyroscope.

Tai & Liu (2018) proposed a mobile robot's exploration through CNN based reinforcement learning. The proposed system main aim was to overcome the problem of the robot mobility in an unknown environment, and to enforce the probability of the reinforcement learning for the vision control system based on the evidence that the reinforcement learning has recently outperforms many different algorithms for machine learning tasks. Ma, Xie & Huang (2019) proposed a convolutional neural network-based obstacle detection for unmanned surface vehicle. The proposed system main aim was to overcome the problem of low detection and classification accuracy of obstacles in the visual inspection of Unmanned surface vehicle. A bidirectional feature of pyramid networks was suggested by the proposed system, incorporating ResNet hybrid network architecture and enhanced DenseNet. Caraiman et al.(2017) proposed a computer vision base obstacle avoidance, to provide the visually impaired person a 3d representation of the surrounding environment.

Tai, Liu & Liu (2016) proposed a deep-network solution towards model-less obstacle avoidance. The proposed system main aim was to overcome the obstacle avoidance system which uses the controlled environment design by constructing a local map. The proposed system demonstrates the efficiency of a hierarchical structure that fuses a decision process with a convolutionary neural network (CNN). The system was a highly compact network structure that takes raw depth images as input and produces control commands as output from the network, achieving a model-less behavior of obstacle avoidance. In real-world indoor settings, we test our approach.

Ahmed, Mahmood & Yeasin (2019) proposed an obstacle avoidance system for visually impaired people using RNN and CNN. The proposed system main aim was to overcome the problem faced by visually impaired and sighted person while wayfinding. The proposed system provided an assistive technology solution for reusable wayfinding for the visually impaired with obstacle avoidance. The system uses a RNN model to predict navigational behaviors, and fine-tuned CNN model for obstacle avoidance. The system created a phone application to create, share and reuse the navigation system.

Zhou et al.(2019) proposed a novel obstacle avoidance system based on LiDAR and CNN image processing approach. The proposed system main aim was to overcome the traffic accidents which are increasing every year due to human mistakes. The proposed system approach is towards a recognition of intelligent pattern generated using LiDAR by counting the number of point cloud data.

The Internet of Things (IoT) is expected to connect to the Internet more than 75 billion devices in 2025 (IHS). Internet of things, or IoT, is one of the biggest advancements made in recent technological times (Abdulla et al., 2020; Eldemerdash et al., 2020; Lakshmanan et al., 2020; Rasheed et al., 2021). Balakrishna, K., & Rajesh, N. (2022) designed a remote monitored solar powered grasscutter robot with obstacle avoidance using IoT.

The main aim of this project is to design and develop a prototype to make a robot that can avoid obstacles by using a neural network. The challenges with the existing obstacle avoidance systems are that most of the system uses sensors such as LiDAR, RADAR and Vision sensors combine. These are expensive sensors and requires an expensive piece of hardware to work on. Also use of this sensor excludes the small business sector as they cannot afford to have expensive equipment's. Use of ultrasonic sensor or infrared sensors for obstacle avoidance can only be justified for controlled environment and for very specific environments. Mobility main parameter is the environment and the best way to observe an environment is with the vision. The obstacle avoidance accuracy can be increase with the better prediction of the objects and implementing algorithm to avoid those objects. Object detection will be performed, and the obstacle will be identified and avoid showing the effectiveness of the algorithm. The focus of this research is to develop an intelligent model to predict detection of the objects and to avoid efficiently those objects behaving as an autonomous robot. An android app will be developed to monitor the environment of the proposed Obstacle avoidance system.

II. DESIGN METHODOLOGY

Fig 1. shows the proposed system block diagram where the vision sensor set as per required, dataset directory which includes the object images required for model training and object detection. Object identification which includes the convolutional neural network model and algorithm for object detection, obstacle avoidance which includes the algorithm for avoiding the obstacles from the input by object detection phase and the system monitoring is done through GUI using Jypter lab. In Fig 1, the block diagram of the obstacle avoidance scheme is shown. The main computer in the device, which is Jetson nano is responsible for connecting the camera with the camera serial interface. Camera is powered using the 20W power adapter, while the Jetson Nano is powered using the jetson nano's built-in power supply. To ensure the object detection on the known side of the frame is much simpler, camera resolution is set to a specific resolution. The dataset was made by creating separate folders for each of the objects and taking several photographs of the objects to prepare a dataset. Dataset was made up of those items that are found on a regular basis, and several other objects which were not common were also included in the dataset.

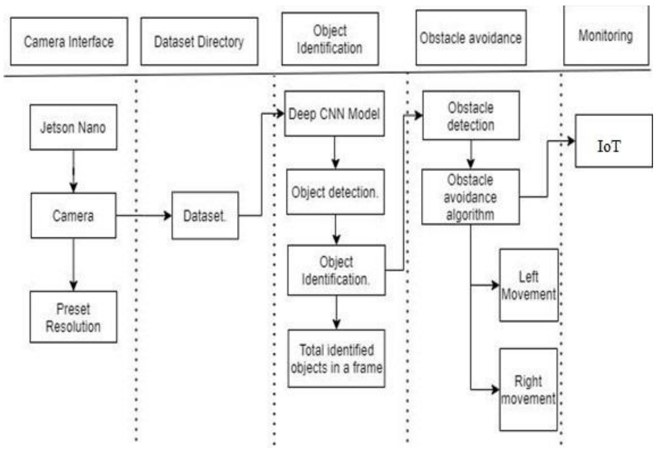

Fig. 1. Block diagram of the proposed system

To be able to identify the object, which is the former part of the obstacle avoidance system, Deep CNN model is built and trained on the Jetson Nano. The detected object is aligned with the directory and labelled. Once that is completed, the detected object is labelled with a block on the frame. The multiple object detection algorithm is developed to make sure that multiple objects can be detected in a single frame. When the object is expected to be observed but the object cannot be found, new data is added to the object's dataset, which results in improving the detection system.

When a detection is made the obstacle avoidance algorithm controls the movement of the robot to avoid the detected target. The algorithm locates the coordinates of the object inside the frame based on the locations of other objects. This is the algorithm that determines the robot's movement. In other words, if the detected object is on the right side of the frame from the centre, the object will move to the left side of the frame and if the detected object is on the left side of the frame from the centre, the object will move to the right side of the frame. In addition, the robot is able to distinguish free space and will be able to drive accordingly. In order to be able to keep track of the movement of the robot, the robot monitoring and programming was connected and made through a local server.

Fig 2. shows the construction details of the overall system. The system main supply is the Lion battery with a voltage of 7.4 volts the two 18650 battery is connected in parallel. Motor driver which operates on I2C connectivity gets the power from the power supply where the Input Vcc pin is connected to the positive terminal of the battery and the Gnd is connected to negative terminal of the battery. The motor drives contain four ports for motor driver two for each side, in which only one port of each side is connected to the motors. The motor drive I2C pin is connected to pin 27 and 28 of the jetson nano which is the I2C pin of jetson nano. The Oled pin Vcc and Gnd is connected to the positive and negative terminal of the battery, while the scl and sda pin is connected to the pin 27 and 28 of the jetson nano. The jetson nano is powered using the barrel jack pin connected to the power supply.

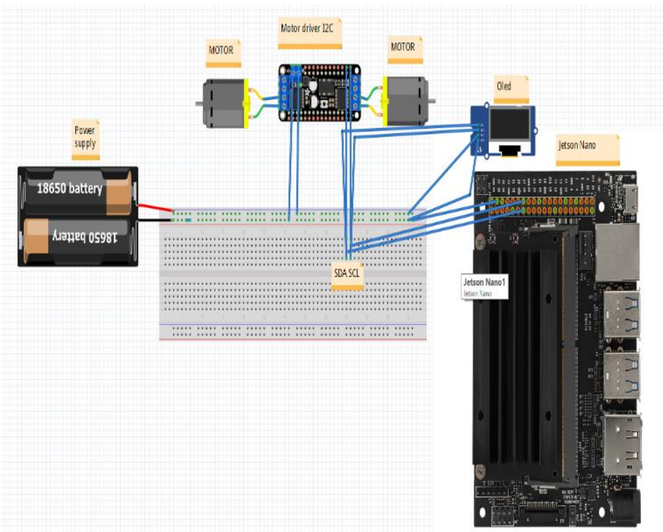

Fig. 2. Construction details

Fig 3. shows the flowchart of the proposed system, in which the initial phase is to get the Jetson platform ready for the project, taking into consideration the needs of the system. Several libraries of Pytorch, jetson inference, and cuda toolkit will be available on the framework. For robot motion autonomy, it is necessary to keep an eye on the robot through Xrdp or VNC. Now that the output frame has been set, the next step is to set the camera's exposure and exposure parameters to get the best and fastest results. Multiple objects will be arranged in any conceivable angle, and then the photographs will be taken. Each object will have its own separate folder which will be needed later for training the model.

A version of the Convolutional Neural Network (CNN) algorithm will be used to build the model. The model will be trained with the images and the testing will be done to see if the accuracy exceeds the minimum threshold. If the accuracy is greater than the minimum threshold, the model will be used for real-time object detection and recognition. In the event that the object is observed but unable to be named, the respective object will be recorded in various angles, and the dataset will be modified. The position of the identified object will be initialised if the object is on the road. If there are no objects on the left side of the frame, the robot will move to the left; if there are objects on the right side of the frame, the robot will move to the right. After the robot has avoided the obstacle, it will automatically orient itself.

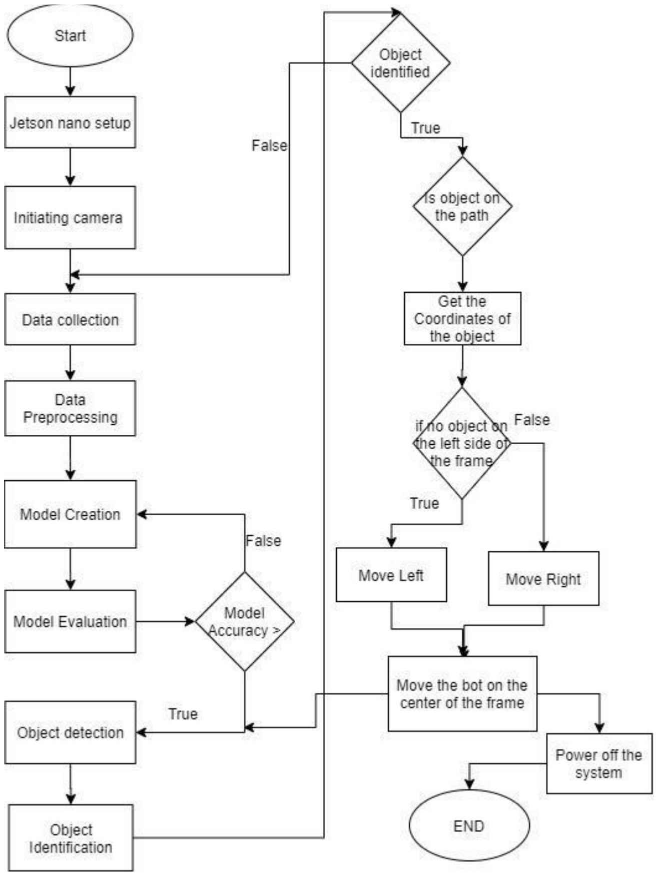

Fig. 3. Flow chart of the proposed system

Fig 4., illustrates the flowchart of the Deep convolutional neural network (CNN) algorithm. This is the first step in the algorithm, in which the dataset directory is loaded. The dataset directory houses all of the image files for each object in its respective folder. The next step is to break the dataset into a training dataset and a research dataset. The training dataset is the input for the Input layer of Deep CNN, which is the first layer since it implements a kernel operation on images using convolution. The Convolutional layer is analogous to a kernel that is applied to images before doing a linear operation that involves multiplication between the input and a certain set of weights, which is known as convolution. The computation being used (in this case, the image input size 5*5*1, image kernel size is 3*3*1, and the resulting layer feature size is 3*3*1) yields an output layer that is equal to the computed input layer. After the max pooling layer processes the feature, it passes the information to the classifier. The classifier then reduces the size of the convolved feature, improving the computing power it would take to process the data. Additionally, the Max pooling layer is also used for determining the dominant element.

Increasing the Fully Connected layer's connectivity (for the most part) is a very inexpensive way of getting better at using the characteristic non-linear combinations described by the convolutional layer performance. As such, the Fully Connected layer is able to comprehend a possibly non-linear function in that space. The model can distinguish between dominant and some low-level features in images over a series of cycles, and then it can classify those features using the SoftMax Classification technique. The last layer is the classification layer, which identifies the data and classifies it for the required performance. The model, and all the test results from the dataset, will be used to determine whether the model works correctly. The real-time image of the camera will be used to further process the results and provide the information on detected objects. This will lead to object recognition, along with image processing.

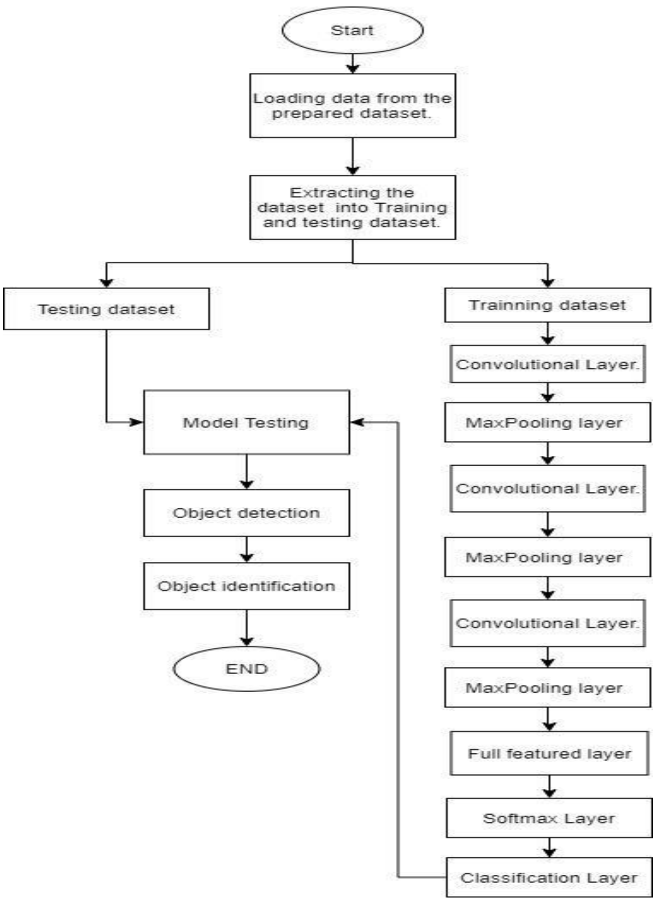

Fig. 4. Flow chart of the CNN

Fig 5. shows the hardware prototype of the proposed system, the components can be seen connected as required and placed accordingly on the designed 3d structure.

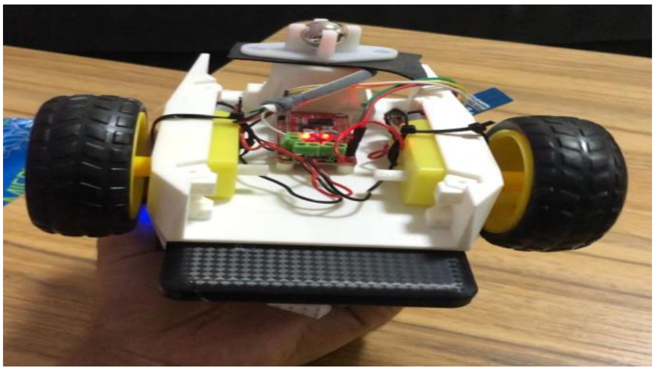

Fig. 5. Hardware prototype

III. PERFORMANCE TESTING AND SIMULATION RESULTS

A. *Network Model Testing*

The test ensures that the selected network model for the object detection has better performance than the other network models when running on the selected hardware jetson nano. This test is very important since the faster the frame rate and object detection will be the better the obstacle avoidance system will perform and enhance the overall efficiency of the system. To execute the test, there were several parameters that were set, and for all the model same settings were applied as shown in Table I. For the Deep learning inference performance, the training guide selected is Jetson inference which is developed especially for jetson platform. The batch size which is used for the specific number of samples required to work before requiring an update on the internal parameters was set at size of 1. Half-precision floating point format (FP16) was used as the precision type.

TABLE I. NETWORK MODEL PARAMETERS

Parameter	Value
Computer	Jetson nano
Training guide	Jetson Inference
Batch Size	1
Precision Type	Fp16
Library	Nvidia TensorRT accelerator

TABLE II. NETWORK MODEL COMPARISON TESTING

Instance	Model type	Output size	Output frame rate (Fps)
1	ResNet-50	224 * 224	36
2	SSD ResNet-18	960*544	5
3	SSD Mobilenet-V2	300 *300	39
4	Inception V4	299 * 299	11
5	Tiny YOLO V3	416* 416	25
6	Open Pose	256*256	14
7	VGG-19	224 * 224	10
8	Super Resolution	481 * 321	15
9	U-Net	1 *512 *512	18
10	SSD Mobilenet-V2	480 * 272	27

Fig 6. shows the network model test that were performed for 10 different models. The model with the highest frame rate

was SSD MobileNet-V2 with output size of 300 * 300 getting a frame rate of 39 fps. The lowest frame rate was SSD ResNet-18 whose output size was 960 * 554 and a frame rate of 5fps. It was seen that when the SSD MobileNet -V2 output size was increased to 480 *272 the frame rate was reduced to 27 fps almost a decline of 13 % in performance. Considering all the other models mostly the models were having a frame rate of less than 20 which is almost 50 % decline in performance. After comparing the model SSD MobileNet-V2 was selected was the model for object detection.

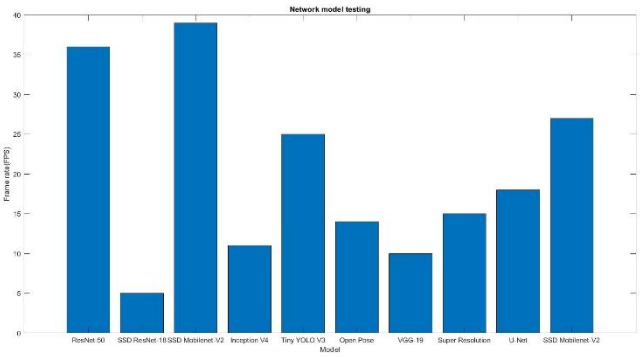

Fig. 6. Network model test

B. *Custom Object Training Test*

This test ensures that the model selected for training can also detect the custom object which were not included in the main library. For the training test to occur multiple pictures were taken, and each object were directed to individual directory. To execute the test, there were several parameters that were set, and for all the model same settings were applied as shown in Table III. For the Deep learning inference performance, the training guide selected is Jetson inference which is developed especially for jetson platform. The batch size which is used for the specific number of samples required to work before requiring an update on the internal parameters was set at size of 1. Half-precision floating point format (FP16) was used as the precision type. Programming platform was Python and the network model selected was SSD MobileNet V2.

TABLE III. NETWORK MODEL PARAMETERS

Parameter	Values
Computer	Jetson nano
Training guide	Jetson Inference
Batch Size	1
Precision Type	Fp16
Programming platform	Python
Network model	SSD MobileNet V2
Library	Nvidia TensorRT accelerator

TABLE IV. NETWORK MODEL COMPARISON TESTING

Instance	Epoch	Validation Loss	Average accuracy
1	1	7.751	82.8
2	10	5.379	89.34
3	20	4.836	92.17
4	30	4.219	93.82
5	40	3.691	94.61
6	50	2.931	95.37
7	70	2.843	95.92
8	80	2.971	95.28
9	90	3.328	92.81
10	100	3.742	91.70

Fig 7. shows the Validation Loss and Fig 8. shows the accuracy prediction for the custom object detection test that were performed for different epoch ranging 1 through an interval of 10 until 100 epochs. Validation loss was calculated by summing up the validation regression Loss and validation Classification Loss. The higher the value for validation loss the worse the system is and the higher the value for accuracy the better the system works. The highest value of validation loss was at 1st epoch where the validation loss was 7.751. The lowest accuracy was at 1st epoch at the value of 82.8%. The lowest value of validation loss was at 70th epoch where the value of validation loss was 2.931 and the highest accuracy also was at the 70th epoch reaching an accuracy of 95.92%. It can be seen from the graph that the validation loss and accuracy go higher until the 70th epoch and then it was lowering down, this statement can be justified since increasing the epoch does improve the accuracy of the system but only for the case where the model is underfitted, after 70th epoch the model was increasing towards the overfitting which causes the accuracy to be lower. Hence the model was trained until the 70th epoch.

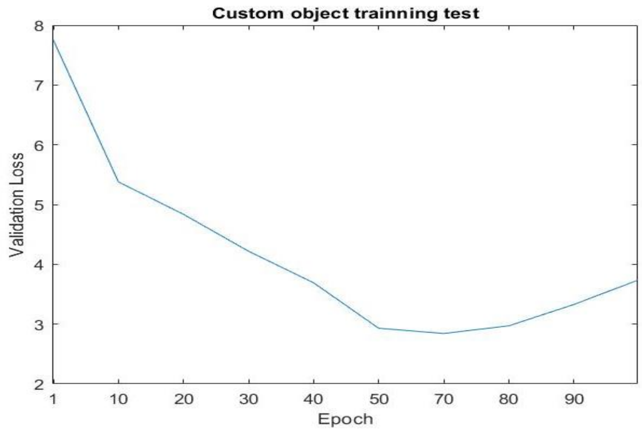

Fig. 7. Validation loss test

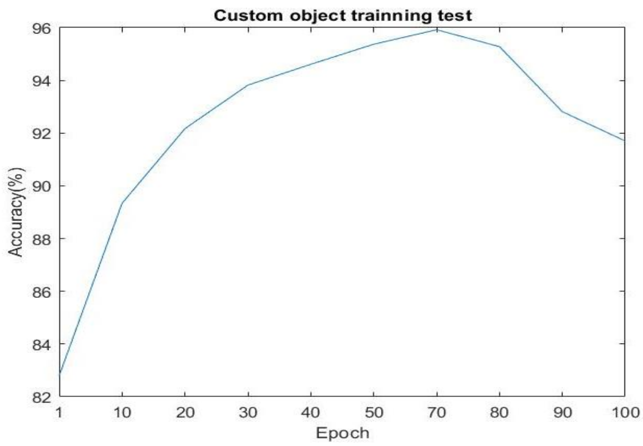

Fig. 8. Accuracy prediction for custom object detection

C. Obstacle Avoidance Accuracy Test

This test ensures the effectiveness of the obstacle avoidance system. For the training test to occur several different objects were placed and the objects were keep relocated in the test area after certain interval to test how many times the bot was collided with the objects within the respected time. To execute the test, there were several parameters that were set, and for all the model same settings were applied as shown in Table V. For the Deep learning inference performance, the training guide selected is Jetson inference which is developed especially for jetson platform. The batch size which is used for the specific number of samples required to work before requiring an update on the internal parameters was set at size of 1. Half-precision floating point format (FP16) was used as the precision type. Programming platform was Python and the network model selected was SSD MobileNet V2.

TABLE V. NETWORK MODEL PARAMETERS

Parameter	Values
Computer	Jetson nano
Training guide	Jetson Inference
Batch Size	1
Precision Type	Fp16
Programming platform	Python
Network model	SSD MobileNet V2
Library	Nvidia TensorRT accelerator

TABLE VI. OBSTACLE AVOIDANCE PERFORMANCE TESTING

Instance	Trial Time(mins)	Collision Count	Average detection accuracy (%)	Average collision distance(m)
1	2	0	92.3	0.033
2	4	0	96.2	0.081
3	6	1	94.6	0.019
4	8	0	92.8	0.037
5	10	0	95.3	0.092
6	12	3	91.2	Error
7	14	0	95.92	0.041
8	16	0	95.28	0.049
9	18	2	92.81	0.014
10	20	2	91.70	0.012

Fig 9. shows the Collision test where the number of times the bot collied with the obstacles. For the first two-time span test the bot does not collided, the first collision occurred during the 6 minutes time span. The maximum collision occurred was during the 12 minutes time span where the bot collided for 3 times against the object. Fig 10. shows the average accuracy (%) during the trial time the highest average accuracy achieved was during the 4 mins time span where the accuracy reached 96.2(%) and the lowest average accuracy was during the 20 minutes time span trail run where the accuracy was 91.70. Overall, the accuracy was above 90% which shows the accurate performance of the developed obstacle avoidance system.

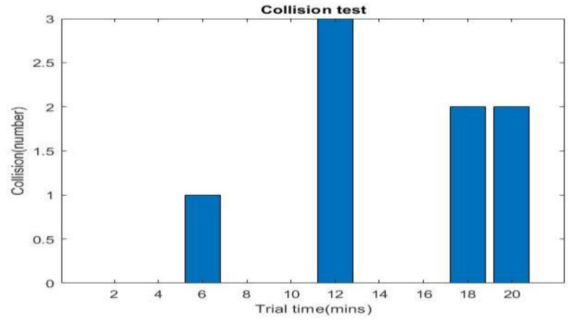

Fig. 9. Collision test

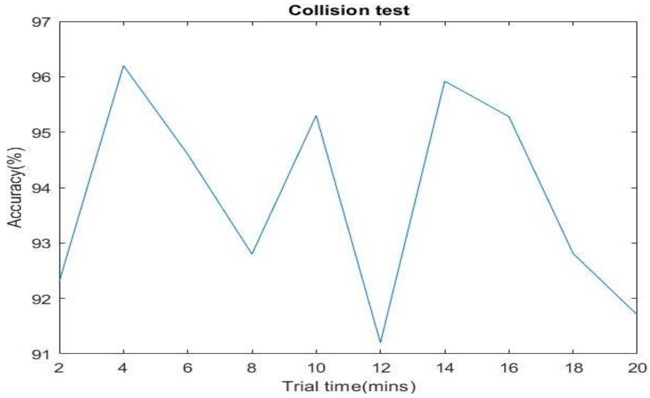

Fig. 10. Accuracy (%) performance test

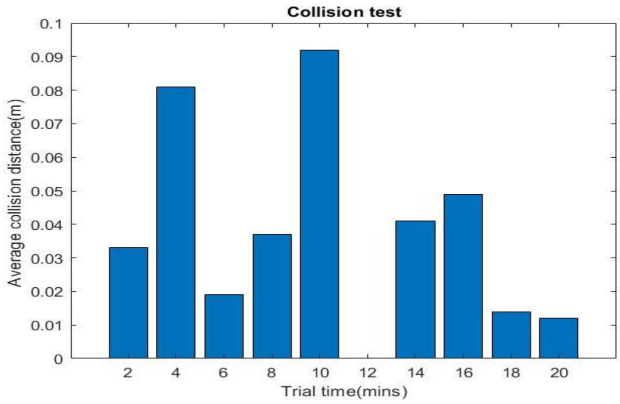

Fig. 11. Average collision distance test

REFERENCES

Abdulla, R., Abdillahi, A., & Abbas, M. K. (2018). Electronic toll collection system based on radio frequency identification system. International Journal of Electrical and Computer Engineering, 8(3), 1602.

AHMED, F., MAHMUD, M. AND YEASIN, M., 2019. RNN and CNN for Way-Finding and Obstacle Avoidance for Visually Impaired. 2019 2nd International Conference on Data Intelligence and Security (ICDIS).

Balakrishna, K., & Rajesh, N. (2022). Design of Remote Monitored Solar Powered Grasscutter Robot with Obstacle Avoidance using IoT. Global Transitions Proceedings.

CARAIMAN, S., MORAR, A., OWCZAREK, M., BURLACU, A., RZESZOTARSKI, D., BOTEZATU, N., HERGHELEGIU, P., MOLDOVEANU, F., STRUMILLO, P. AND MOLDOVEANU, A., 2017. Computer vision for the visually impaired: the sound of vision system. In Proceedings of the IEEE International Conference on Computer Vision Workshops (pp. 1480-1489).

DANIEL, A., 2018. Object Recognition Using Deep Learning. International Journal of Engineering Research and, V7(06).

Eldemerdash, T., Abdulla, R., Jayapal, V., Nataraj, C., & Abbas, M. K. (2020). IoT Based Smart Helmet for Mining Industry Application.

International Journal of Advanced Science and Technology, 29(1), 373–387.

GAYA, J.O., GONÇALVES, L.T., DUARTE, A.C., ZANCHETTA, B., DREWS, P. AND BOTELHO, S.S., 2016, October. Vision-based obstacle avoidance using deep learning. In 2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR) (pp. 7-12). IEEE.

IHS. Internet of Things (IoT) Connected Devices Installed Base Worldwide from 2015 to 2025 (In Billions). Available online: https://www.statista.com/statistics/471264/iot-number-of-connecteddevices-worldwide/

Lakshmanan, R., Djama, M., Selvaperumal, S. K., & Abdulla, R. (2020). Automated Smart Hydroponics System Using Internet of Things. International Journal of Electrical and Computer Engineering, 10(6), 6389–6398. https://doi.org/10.11591/IJECE.V10I6.PP6389-6398

LEE, G. AND CHWA, D., 2017. Decentralized behavior-based formation control of multiple robots considering obstacle avoidance. Intelligent Service Robotics, 11(1), pp.127-138.

LIU, C., ZHENG, B., WANG, C., ZHAO, Y., FU, S. AND LI, H., 2017. CNN-Based Vision Model for Obstacle Avoidance of Mobile Robot. MATEC Web of Conferences, 139, p.00007.

MA, L., XIE, W. AND HUANG, H., 2020. Convolutional neural network based obstacle detection for unmanned surface vehicle. Mathematical Biosciences and Engineering, 17(1), pp.845-861.

Rasheed, W. M., Abdulla, R., & San, L. Y. (2021). Manhole Cover Monitoring System Over IOT. Journal of Applied Technology and Innovation, 5(3), 1–6.

R Chatila, G Andrade, S Lacroix and A Mallet. (1999) Motion control for a planetary rover. Proceedings of the International Conference on Field and Service Robotics. pp.381-388.

TAI, L. AND LIU, M., 2016. Mobile robots exploration through cnn-based reinforcement learning. Robotics and Biomimetics, 3(1).

ZHOU, C., LI, F., CAO, W., WANG, C. AND WU, Y., 2018. Design and implementation of a novel obstacle avoidance scheme based on combination of CNN-based deep learning method and liDAR-based image processing approach. Journal of Intelligent & Fuzzy Systems, 35(2), pp.1695-1705.