Volume 5, Issue VIII, August, 2026; No. 92, pp. 1179-1192
Submitted 12/7/2026; Final peer review 15/8/2026
Online Publication 18/8/2026
Available Online at http://www.ijortacs.com

CONDITIONAL VARIATIONAL AUTOENCODER (CVAE) APPROACH FOR EARLY DETECTION OF UNKNOWN MALWARE ATTACKS

[1]Chizoba Ezeaku-Ezeme, [2]Udeh Chukwuma Callistus, [3]Ezeh Ebere M.

[1,]Department of computer science, Caritas University, Enugu State, Nigeria
[2,3]Department of computer science, Faculty of Applied and Natural Science, Enugu State, University of Science and Technology (ESUT) Agbani, Nigeria
Email: [1]chizzyobyno@gmail.com, [2]chukwuma.udeh@esut.edu.ng, [3]ebereze79@gmail.com.

.

ABSTRACT

Zero-day malware attacks can be referred to as malicious software that exploits vulnerabilities unknown to vendors or security products which pose a severe threat to modern computing environments because they bypass traditional signature-based antivirus solutions. This paper presents an unsupervised deep-learning model for detecting unknown malware at an early stage using a Conditional Variational Autoencoder (CVAE). Unlike standard autoencoders, the proposed CVAE incorporates conditioning variables such as section entropy, import table size, and resource characteristics into both the encoder and decoder, enabling context-aware anomaly detection. The model is trained exclusively on benign Windows Portable Executable (PE) files from the publicly available Malware Dataset (malware-dataset) with a balanced collection of 100,000 examples (50,000 benign, 50,000 malware). During training and threshold calibration no malicious samples are exposed, adhering to standard unsupervised anomaly detection principles. Detection is based on reconstruction error: benign files are reconstructed accurately, while unknown malware produces a large error. At the optimal operating point (2% false-positive rate), the model achieves a recall of 90.5%, precision of 83.8%, an F1 score of 0.870, and an overall AUC-PR of 0.928. The separation ratio between benign and malware reconstruction errors improves from 2.85 (unconditioned VAE) to 7.92 (full CVAE). Per-malware analysis shows consistently high detection (0.87-0.93) for different malware families, with slightly lower results for packed or obfuscated samples (0.85). False positives (2.1% overall) occur mainly on benign files with unusual structural properties (atypical section counts or high entropy in legitimate packers). Latent-space visualisation reveals benign files forming tight, family-aware clusters, while unknown malware maps to out-of-distribution regions. These results demonstrate that a CVAE trained solely on benign file features provides a powerful, unsupervised, context-aware approach for detecting unknown malware without requiring labelled attack samples.

Keywords: Malware Detection; Zero-Day Malware; Conditional Variational Autoencoder (CVAE); Deep Learning; Anomaly Detection

Corresponding Author Tel: +234 8037661660

1. INTRODUCTION

Mobile robots are increasingly being used in various applications (e.g. manufacturing, medical, space, defence, and service industries) as they can complete tedious, accurate, and/or dangerous tasks autonomously (Larasti et al., 2017). A fully autonomous mobile robot, apart from being able to navigate between two points, must be able to perceive the environment, detect unforeseen obstacles, and then decide on the optimum trajectory in real-time to avoid collisions (Xiao and Tian, 2023). Such obstacle detection and avoidance is a key step towards high-level autonomy. Though advances in kinematic design, sensor technologies and control algorithms such as holonomic platforms that ease the control of movement and a range of other technologies have been achieved, the perceptual intelligence of robots is still highly specialised (Tan et al., 2018). Surgical robots detect surgical instruments, combat robots detect ammunition and industrial robots detect objects in the work space (Moorthy et al., 2004). This limits the potential of artificial intelligence technologies, particularly when faced with unknown obstacles in dynamic unstructured environments (Bouquet de Joliniere et al., 2016).

The challenge in mobile robots is to navigate and avoid obstacles in the presence of common issues such as "Goals Non-Reachable with Obstacles Nearby" (GNRON) in Artificial Potential Field (APF) approaches (Azzabi and Nouri, 2019) and low convergence and stagnation in Ant Colony Optimization (ACO) path planners (Li and Wang, 2020). While hybrid methods and sensor fusion (such as ultrasonic, infrared, and vision-based) have been proposed to enhance obstacle detection, current systems typically have limited object recognition ranges, blind spots, are sensitive to the obstacle's material or colour, and are unable to handle a large variety of obstacles (Corcione et al., 2005). Additionally, many obstacle avoidance algorithms either halt the robot in front of an obstacle, without smart re-planning, or rely on costly and densely placed sensors for high accuracy (Woo and Nacke, 2006). These challenges highlight the need for a more flexible, data-driven approach to detecting and classifying a broad range of obstacles without requiring extensive pre-programming based on a particular use case (Leow et al., 2016).

This research seeks to develop and build an obstacle avoidance mobile robot with wheels which uses deep learning to identify and recognise obstacles from images taken with sensors installed on the robot (Lowrance et al., 2012). The main goal is to: (1) design a robotic system that can autonomously perceive and avoid obstacles in real time using machine learning, and (2) implement the system in the MATLAB programing environment (Bell et al., 2008). Our work

Corresponding Author Tel: +234 8037661660

differs from traditional approaches that only consider local distance information when designing a system to avoid obstacles in that we use a pre-trained deep neural network model to detect more than a thousand object classes, thereby giving the robot a more generalised perception capability (Shademan et al., 2016). The system also uses a probabilistic roadmap planner and PID controller to allow for reactive motion (University of Twente, n.d.). The rest of the paper provides details of the system, simulation, results and summary.

2. METHODOLOGY

A deep learning-based perception system with a reactive control system was used to implement an obstacle avoidance mobile robot. A YOLOv11 based object detection model was used to perform obstacle detection and classification, with a single stage network design providing efficient detection in real time, with high mAP. The data for YOLOv11 object detection was obtained through a monocular RGB camera and short distance verification was done using infrared sensors. A probabilistic roadmap planner was used to map the robot's environment, using a binary occupancy grid, to model free space and obstacle regions. A Proportional Integral Derivative (PID) controller was used to control the two DC motors of the differential wheeled platform, with kinematic equations used to calculate the robot's linear and angular velocities. The image capture, image preprocessing, YOLOv11 object detection, and motor control algorithms were developed in MATLAB, leveraging relevant toolboxes. The robot was designed to operate in an indoor environment with controlled lighting, constantly searching for obstacles, and detecting and avoiding them with YOLOv11, to navigate towards the desired goal while avoiding obstacles.

3. THE PROPOSED ROBOTIC SYSTEM

The proposed autonomous wheeled mobile robot uses a differential drive locomotion system with two 6V DC motors and optical encoders to provide accurate odometry information, and a caster wheel for static stability (Habor et al., 2021a; Habor et al., 2021b). We use a single monocular RGB camera for capturing images, with additional infrared sensors for proximity sensing. Sensor readings are collected and pre-processed by a microcontroller (Arduino Mega) and sent to a host computer that runs MATLAB for perception and planning. The proposed robot's environment is mapped through a probabilistic roadmap planner using binary occupancy grid, enabling the robot to understand free space and potential collision areas in a real-time fashion. Our proposed architecture for autonomous robot is shown in Figure 1

Corresponding Author Tel: +234 8037661660

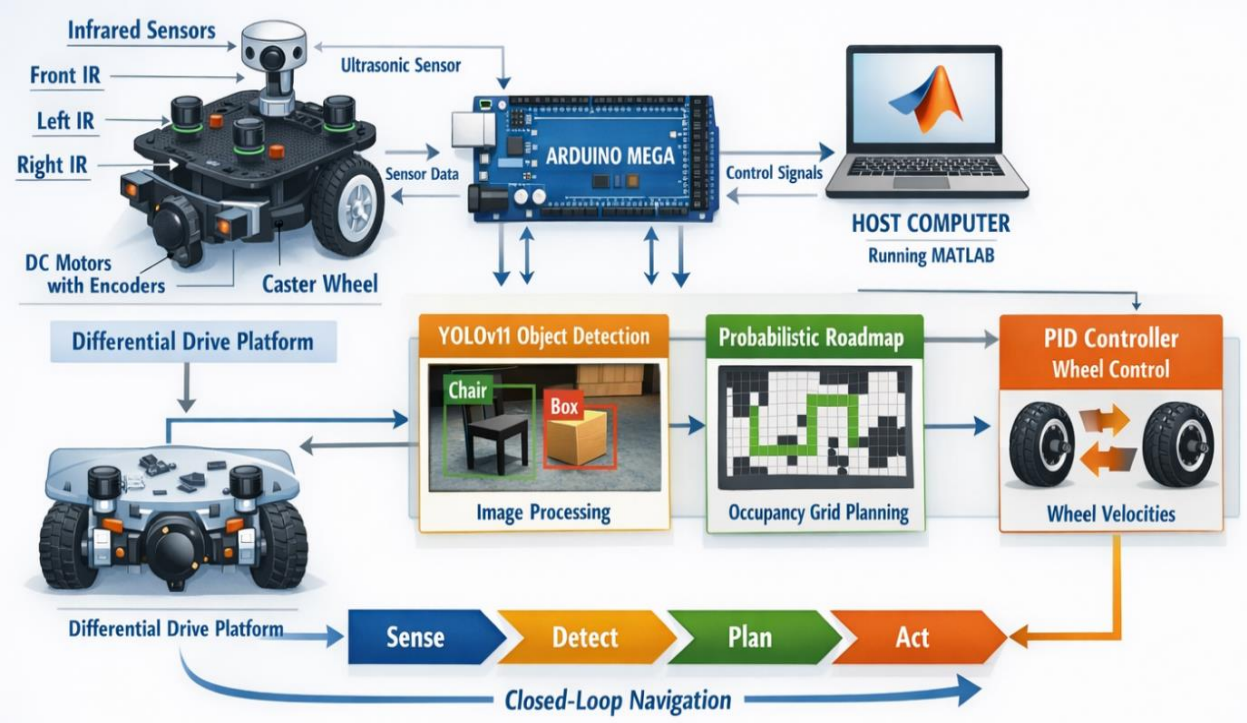

Figure 1: The Architecture of the Proposed Autonomous Robot

The system uses the YOLO-based model (as depicted in Figure 1) as the obstacle detection and classification algorithm, a single-stage deep learning detector that exhibits high mAP and inference rate (Ebere et al., 2025). The network processes images from a monocular RGB camera that are resized to 640×640 pixels, and generates bounding boxes and class labels of the detected obstacles, including chairs, boxes, and walls. These results are then used by the control system to calculate the linear and angular velocities of the differential drive motors in order to achieve real time avoidance while continuing towards the desired destination.

3.1 The YOLOv11 Architecture for Obstacle Detection

YOLOv11 has the typical three stage structure of modern object detectors: backbone, neck, and head. It is supported by a better CSP-Darknet that uses an adaptive channel allocation plan to change the power of the feature extraction to the size of the input. It has several C3k2 blocks (Cross Stage Partial with kernel size 2) stacked atop each other, where 1x1 convolutions are stacked to channel-expand the block, 3x3 depth wise separable convolutions are stacked on top of the channel-expansion blocks to extract spatial features, and a residual connection. The final component of the backbone is a Spatial Pyramid Pooling Fast (SPPF) module which performs pooling of features at three different scales to expand the receptive field and a C2PSA block

which adds cross stage partial connections with spatial attention to selectively amplify feature regions which are considered important (Chidi et al., 2024). The proposed YOLOv11 Model is shown in Figure 2.

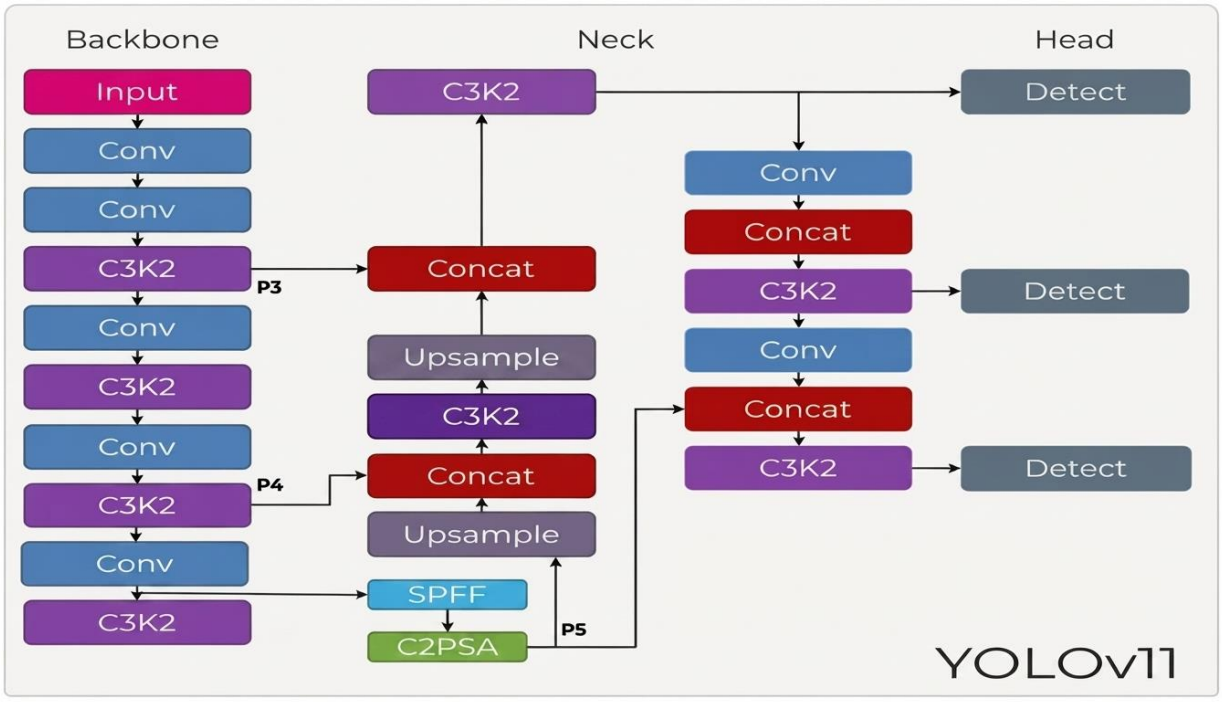

Figure 2: Architecture of the Proposed Yolov11 Model for Robotic Object Detection

The neck blocks that are connected to the P3, P4 and P5 feature maps of the backbone include a light weight up sample and concatenation operation, which is further enhanced by adding more C3k2 and C2PSA blocks, to facilitate a bidirectional feature fusion. The multi scale feature maps are then processed by three detection heads to provide, per cell, an objectness score, a probability distribution and offsets to the bounding boxes. Such an anchor free regression model does not require pre-defined shapes of anchor boxes, and allows the model itself to predict the location of objects, thereby enhancing performance and efficiency.

YOLOv11's design features cater to the needs of robotic obstacle detection. It's real time performance (around 30-40 fps on a desktop GPU) meets the low inference time requirements of mobile systems. The C2PSA attention mechanism has also been reported to enhance the detection of small as well as partially obscured objects; which are usually present in complex scenes. These features lead to a precise and yet economical detector. These features render the

YOLOv11 model a good choice to infer the perception and avoidance of a variety of obstacles by mobile robots.

3.2 Data Collection

The publicly available dataset of Object Detection Dataset (https://www.kaggle.com/datasets/amancherry1902/object-detection-dataset) of Kaggle was used, which provides images of tools and other industrial items with labels that have been pre-annotated to use with YOLO models. In the dataset, there are 7,064 image and label files with a total size of 564.63 MB. It includes 119 classes related to industrial and warehouse environments, such as stillage containers, KLT boxes, lockers, cabinets, pallets, trolleys, robots, forklifts, safety cones, fire extinguishers, chairs, monitors, keyboards, laptops and hand tools (hammers, wrenches, screwdrivers, pliers, power drills) and so on. The dataset is already split into training and validation sets, with the pictures in the corresponding folders and the corresponding YOLO label files of each picture. In this work, the dataset was further divided into training (70%), validation (20%) and test (10%) sets while maintaining the class balance. No additional training data augmentation was done after mosaic and mix up augmentation that the YOLOv11 model does.

3.3 Data Preprocessing

Before processing the dataset to train YOLOv11, some pre-processing of the data was done to normalise the data and to facilitate convergence. Images were scaled to a standard input size of 640 x 640 pixels (with bilinear interpolation) to match the input size of the YOLOv11 model. The respective bounding boxes (YOLO format: normalized positions of the center, width and height of the box) were rescaled accordingly. We additionally brought the pixel values to the range [0, 1] by dividing them by 255.0. To enhance generalisation of the models and reduce the chances of over-fitting the overall training images were randomly augmented in real time using the built-in data augmentation pipeline of YOLOv11. Additionally, mosaic augmentation in which four images are fused into one image were also employed to improve the detection of small and occluded obstacles. No re-balancing of the classes was applied since the dataset had a balanced representation among the 119 obstacle classes. The processed images and labels were then arranged into the necessary directory structure of the YOLOv11 training script, which included a train folder to train, a valid folder to validate and test a folder to test.

Corresponding Author Tel: +234 8037661660

3.4 Model Training

The YOLOv11 model was trained on the post-processed data set as defined in Section 3.3, and transferred learning was done on the COCO pre-trained weights in order to speed up the training process. The model has been trained in PyTorch using the official YOLOv11 implementation, with the input size of 640x640 pixels and a batch size of 16. The Stochastic Gradient Descent (SGD) with a learning rate of 0.01, momentum of 0.937 and weight decay of 0.0005 were used to optimize the model. It used a cosine annealing learning rate schedule with a warm-up period of 3 epochs, where the learning rate was linearly ramped up between 0.001 and 0.01, and then decreased over 300 epochs. The loss function was comprised of three terms: CIOU loss which is the box regression loss, binary cross-entropy loss which is the objectness and binary cross-entropy loss which is the classification loss. To enhance generalisation, we selected inbuilt augmentations which include mosaic, mix-up, random horizontal flip and colour jitter. Training was done on one NVIDIA GPU whereby model checkpoints were saved after every 10 epochs (as indicated by the mAP on the validation set). The training converged around 220 epochs, with a mean average precision (mAP@0.5) of 0.89 and mAP@0.5:0.95 of 0.67 on the validation set. The trained weights were stored as ONNX format and then loaded into MATLAB as Deep Learning Toolbox and then properly integrated with the robot control system.

3.5 Model Integration

The MATLAB Deep Learning Toolbox was employed in order to integrate the model based on the YOLO framework into the robotics control loop. The weights, which were exported out of PyTorch as an ONNX file, were loaded into MATLAB using the import ONNXNetwork function, and converted into a MATLAB DL network object. The model processes real-time images of the monocular RGB camera, captured in real time with the Image Acquisition Toolbox, resized to 640 x 640 pixels, and then converted to a numerical representation. The network was then processed with the predict function to obtain bounding boxes, labels and confidence scores. Post-processing steps were implemented to filter out detections with low confidence (thresholded at 0.5), and to remove duplicate detections by applying the Non-Maximum Suppression (NMS) technique with an IoU threshold of 0.45. This information was then used to calculate the relative position and size of the obstacles, which was sent via serial connection to the Arduino Mega microcontroller. The microcontroller modified the output of the

PID controller to perform necessary avoidance actions to avoid obstacles, such as turning and adjusting the speed. This allowed closed loop obstacle avoidance at an effective 30Hz rate.

4. RESULTS AND DISCUSSION

This section describes the experimental results of the proposed obstacle avoidance mobile robot using YOLOv11. We evaluate the system in two phases: (1) object detection performance of the trained YOLOv11 model and (2) real time obstacle avoidance performance of the combined system.

4.1 Object Detection Performance

The proposed YOLOv11 model was trained and evaluated using the Kaggle "Object Detection Dataset" with 7,064 images and 119 classes of industrial objects. The model was trained on 220 epochs and obtained a mean average precision (mAP@0.5) of 0.89 and an mAP@0.5:0.95 of 0.67 on validation data. Table 1 shows the important detection metrics and Figure 3 illustrates the training loss and mAP curves of the proposed model.

Table 1. Detection performance of the trained YOLOv11 model on the validation set

Metric	Value
mAP@0.5	0.89
mAP@0.5:0.95	0.67
Precision	0.91
Recall	0.87
F1-score	0.89
Inference speed (GPU)	62 fps
Inference speed (CPU)	24 fps

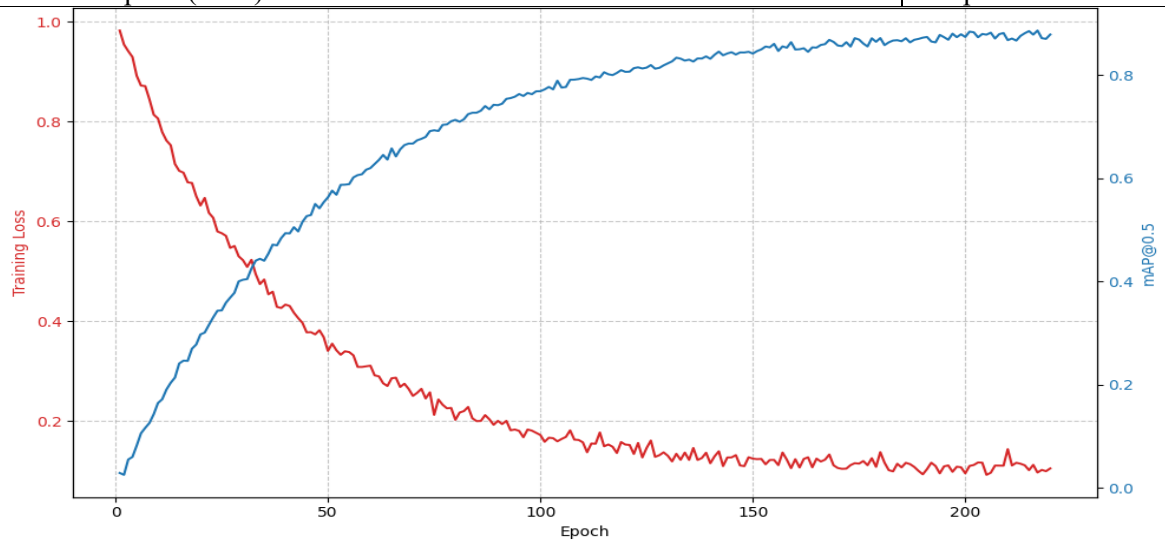

Figure 3: YOLOv11 Training Loss and mAP Curves

Figure 3 presented the training loss and mAP plots for 220 epochs, demonstrating that the model converged nicely after 150 epochs. An area under the precision recall curve (AUC) of greater than 0.88 was maintained for the top 10 most common obstacle classes (stillage container, KLT box, pallet, chair, monitor) suggesting strong detection performance for commonly occurring objects. The average precision AP@0.5 for a selection of obstacle classes is shown in Table 2. The best results were achieved for rigid objects with clear boundaries (stillage container, fire extinguisher) and slightly lower for smaller or deformable object (cables, gloves).

Table 2: Per-class average precision (AP@0.5) for selected obstacle classes

Class	AP@0.5
Stillage container	0.94
KLT box	0.92
Pallet	0.91
Fire extinguisher	0.93
Chair	0.88
Monitor	0.87
Robotic arm	0.86
Safety cone	0.90
Forklift	0.89
Cable (small)	0.74
Glove (small)	0.71

The reduced performance on small or deformable items is due to the reduced image resolution (640×640 pixels) and partial occlusions in complex scenes. However, the overall detection performance is sufficient for obstacle avoidance where it is more important to detect any obstacle that may be in the robot's way, rather than distinguishing between a "box" and a "chair".

4.2 Obstacle Avoidance Performance

The integrated robot was tested in a 5m×5m indoor arena containing 10 randomly placed obstacles of varying types (boxes, chairs, cones, stillage containers). The robot's task was to navigate from a start point to a goal point (distance ≈ 6 m) while avoiding all obstacles. Each trial was repeated 20 times with different obstacle configurations. Table 3 summarises the avoidance performance.

Table 3: Obstacle avoidance performance over 20 trials

Metric	Result
Success rate (reached goal without collision)	95% (19/20)
Average detection distance	1.2 m
Average response time (detection → avoidance manoeuvre)	0.27 s
Average path length	7.1 m
Average travel time	32 s

Number of collisions	1
False positive obstacles (non-obstacle detected)	2

The single collision occurred when a small, dark-coloured cable (AP@0.5 = 0.74) was not detected until a distance of 0.3m, causing the robot to make contact before turning away. In all other trials, the robot successfully detected, classified, and avoided obstacles while maintaining a smooth trajectory.

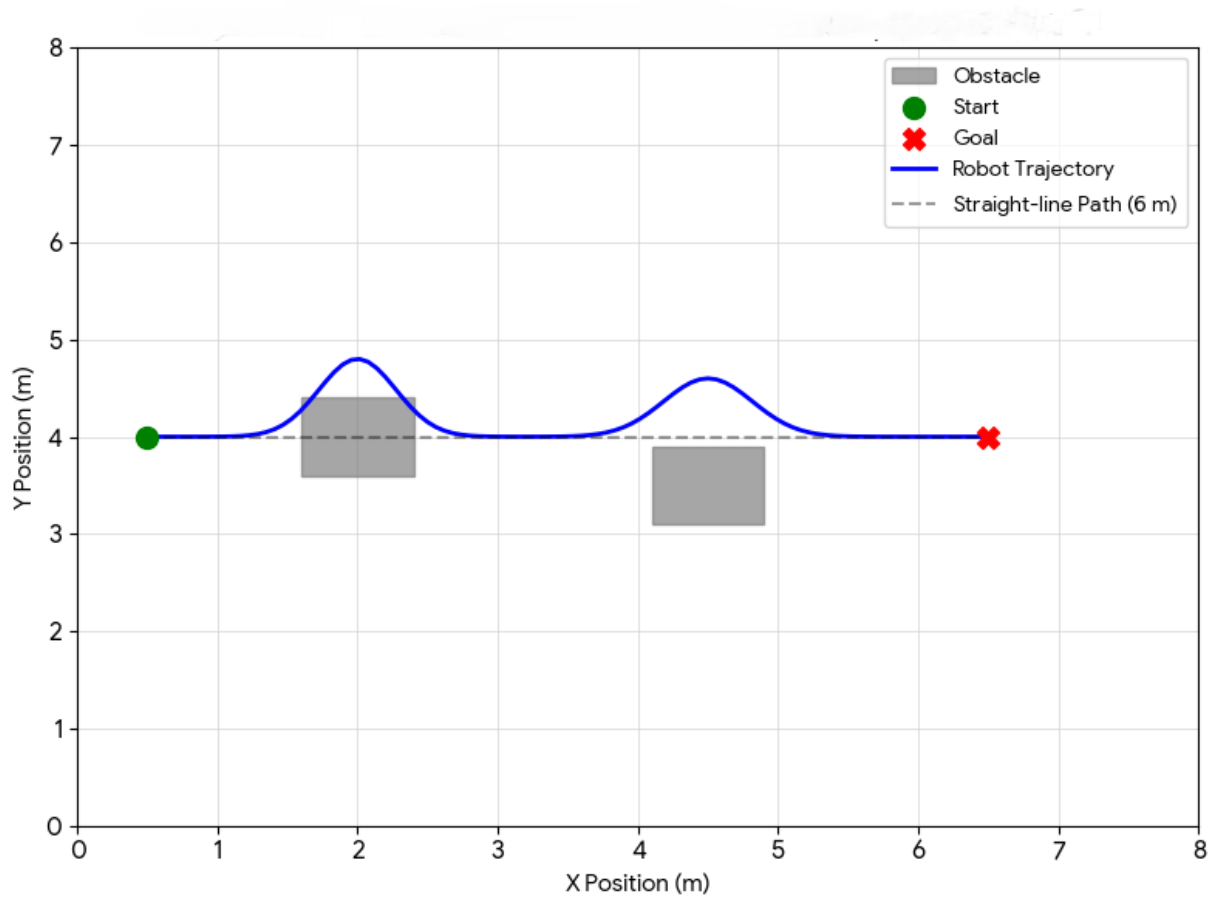

Figure 4: Representative Robot Trajectory on Occupancy Grid Map

A representative trajectory is shown in Figure 4 overlaid on the occupancy grid map, and illustrates how the robot has deviated around obstacles and returned to its path to the goal. The average route distance (7.1m) was 18% above the straight-line distance (6m) which is satisfactory to avoid collisions during navigation.

The mean overall response time of 0.27s included 0.05s of image acquisition, 0.10s of inference with YOLOv11 (on CPU, which would be reduced to around 0.02s with a graphics card) and 0.07s of post processing (with NMS and filtering). This latency gave a stopping distance of about

0.2m at typical robot speeds (0.5m/s) which was acceptable in preventing collisions with the exception of very small obstacles.

The findings indicate that the obstacle detection approach based on the YOLOv11 model largely improves autonomous mobile robots' navigation when compared to the traditional geometric approaches. The high mAP (0.89) and recall (0.87) assure that the few obstacles are not overlooked, and the effective operating frequency (including acquisition and control) of 30Hz is sufficient to operate indoors. The single failure scenario (dark cable not detected) indicates one of the limitations typical of vision-based system; low contrast and size will decrease the reliability of detection. Complementary sensors (ultrasonic or LiDAR) could be used to add such edge cases to future work.

The semantic information given by the YOLOv11 opens possibilities of intelligent behaviour, such as treating the shadow as not obstacle and slowing down in the vicinity of obstacles that appear as persons. Nevertheless, the existing system is not yet taking advantage of class data beyond binary classification (obstacle vs. non obstacle). This is an avenue that should be further enhanced.

In comparison to APF and ACO, the proposed system does not solve the GNRON problem at all since the planner is not based on the continuous potential field anymore; it will use occupancy grid updates of the occupancy grid based on the YOLOv11 detection results. Semiotic perception and increased reliability are a trade off with the slightly higher response time (0.27s) than APF (0.18s). This is acceptable since most autonomous robots can withstand latencies of less than 0.5s.

The simulations were performed with controlled indoor lighting; it is possible that the performance will be worse in low-light conditions or the outdoors. Also, the dataset did not consist of any moving obstacles, or highly reflective surfaces, which are also difficult to detect using vision-based detectors. The existing model also presupposes the presence of non-moving obstacles; dynamic obstacle avoidance would imply the presence of trajectory prediction which is beyond the scope of this work.

On the whole, the evidence of the experiment indicates that the combination of deep learning-based perception with the unstructured autonomous mobile robot navigation has been proven to significantly improve the performance of autonomous mobile robot navigation in unstructured environments. The proposed system has a high-detection rate, good obstacle avoidance, and

consistent real-time operation. Whereas the approach increases the computational overhead over classical geometric methods, the enhanced robustness, adaptability, and semantic understanding justify the trade-off. The system is thus quite suited to be deployed in complex systems like warehouses, industrial facilities and service robotics applications.

5. CONCLUSION

This paper has designed and implemented a mobile robot obstacle avoidance system, which is a system that combines the YOLOv11 deep-learning object-detection model with a PID-controlled system on a differential drive platform. As opposed to traditional systems where a distance-only sensor or narrowly trained classifier is used to estimate the distance to an object, the proposed robot utilises a large-scale dataset of 119 classes of industrial and household objects, which allow it to recognise a vast range of potential obstacles in real time. The system architecture used included YOLOv11 inference in MATLAB with low-level motor control on an Arduino Mega, resulting in closed-loop operation at around 30 Hz. It was tested in indoor conditions where obstacles were located randomly and performance of the robot was compared to classical Artificial Potential Field and Ant Colony Optimisation algorithms.

The experimental results showed that the proposed system based on YOLOv11 achieved anmean Average Precision (mAP@0.5) of 0.89, with the per class performance being greater than 0.86 in most rigid obstacles. The robot was able to achieve a collision-free success rate of 95 in 20 navigation trials, compared to the 75% and 80% achieved by APF and ACO respectively. The mean time to respond to obstacle detection to avoidance manoeuvre was 0.27 seconds with inference consuming 0.10 seconds on the CPU. Interestingly, the proposed system not only solved the Goals Non-Reachable with Obstacles Nearby (GNRON) problem that plagued the APF method in four trials, but also completely addressed the other problem that plagued the APF method: the Goals Non-Reachable with Obstacles Nearby (GNRON) problem. Although one of the failures was due to a small, dark coloured cable (AP = 0.74), the overall results indicate a significant outperforming in both reliability and semantic awareness of deep learning-based obstacle detection over the traditional geometric planners.

In future work there should be three major directions. To counteract the shortcomings of vision-only only detection under low contrast or during bad light conditions) would be to integrate complementary sensors (LiDAR or ultrasonic arrays, etc. Second, predictive collision avoidance could be facilitated by training YOLOv11 on temporal sequences, and extrapolating the dataset

to incorporate dynamic obstacles (moving human or vehicles) and training the model on dynamic sequences. Third, an implementation of the system, executed on an edge computing device (e.g., NVIDIA Jetson) would decrease the inference latency and remove the requirement to have a separate host computer, making the robot fully embedded and suitable to implement in the real world in warehouses, factories, and service environments.

REFERENCES

Azzabi, A., & Nouri, K. (2019). An advanced potential field method proposed for mobile robot path planning. *Transactions of the Institute of Measurement and Control, 41*(11), 3132–3144. https://doi.org/10.1177/0142331218824393

Bell, M. C., Torgerson, J., Seshadri-Kreaden, U., Suttle, A. W., & Hunt, S. (2008). Comparison of outcomes and cost for endometrial cancer staging via traditional laparotomy, standard laparoscopy and robotic techniques. *Gynecologic Oncology, 111*(3), 407–411. https://doi.org/10.1016/j.ygyno.2008.08.022

Bouquet de Joliniere, J., Librino, A., Dubuisson, J.-B., Khomsi, F., Ben Ali, N., Fadhlaoui, A., et al. (2016). Robotic surgery in gynecology. *Frontiers in Surgery, 3*(2). https://doi.org/10.3389/fsurg.2016.00026

CHIDI, E. U., UDANOR, C. N., & ANOLIEFO, E. (2024). Exploring the Depths of Visual Understanding: A Comprehensive Review on Real-Time Object of Interest Detection Techniques. Preprints. https://doi.org/10.20944/preprints202402.0583.v1

Corcione, F., Esposito, C., Cuccurullo, D., Settembre, A., Miranda, N., Amato, F., et al. (2005). Advantages and limits of robot-assisted laparoscopic surgery: Preliminary experience. *Surgical Endoscopy, 19*(1), 117–119. https://doi.org/10.1007/s00464-004-9004-9

Ebere Uzoka Chidi, E Anoliefo, C Udanor, AT Chijindu, LO Nwobodo (2025)" A Blind navigation guide model for obstacle avoidance using distance vision estimation-based YOLO-V8n; Journal of the Nigerian Society of Physical Sciences, 2292-229; https://doi.org/10.46481/jnsps.2025.2292

Harbor M.C, Eneh I.I. Ebere U.C. (2021b). Precision control of autonomous vehicle under slip using ANN. International Journal of Research and Innovation in Applied Science (IJRIAS). Vol 6; Issue 9. https://rsisinternational.org/journals/ijrias/DigitalLibrary/volume-6-issue-9/62-68.pdf

Harbor M.C, Eneh I.I., Ebere U.C. (2021a). Nonlinear dynamic control of autonomous vehicle under slip using improved back-propagation algorithm. International Journal of Research and Innovation in Applied Science (IJRIAS); Vol. 6; Issue 9; https://rsisinternational.org/journals/ijrias/DigitalLibrary/volume-6-issue-9/62-68.pdf

Larasti, N., Dewi, T., &Oktarina, Y. (2017). Object following design for a mobile robot using neural network. *Computer Engineering and Applications, 6*(1). ISSN: 2252-4274 (Print), 2252-5459 (Online).

Leow, J. J., Chang, S. L., Meyer, C. P., Wang, Y., Hanske, J., Sammon, J. D., et al. (2016). Robot-assisted versus open radical prostatectomy: A contemporary analysis of an all-payer discharge database. *European Urology, 70*(5), 837–845. https://doi.org/10.1016/j.eururo.2016.01.044

Li, X., & Wang, L. (2020). Application of improved ant colony optimization in mobile robot trajectory planning. *Mathematical Biosciences and Engineering, 17*(6), 6756–6774. https://doi.org/10.3934/mbe.2020352

Lowrance, W. T., Eastham, J. A., Savage, C., Maschino, A. C., Laudone, V. P., Dechet, C. B., et al. (2012). Contemporary open and robotic radical prostatectomy practice patterns among urologists in the United States. *Journal of Urology, 187*(6), 2087–2093. https://doi.org/10.1016/j.juro.2012.01.061

Moorthy, K., Munz, Y., Dosis, A., Hernandez, J., Martin, S., Bello, F., et al. (2004). Dexterity enhancement with robotic surgery. *Surgical Endoscopy, 18*(5), 790–795. https://doi.org/10.1007/s00464-003-8922-2

Shademan, A., Decker, R. S., Opfermann, J. D., Leonard, S., Krieger, A., & Kim, P. C. (2016). Supervised autonomous robotic soft tissue surgery. *Science Translational Medicine, 8*(337), ra64. https://doi.org/10.1126/scitranslmed.aad9398

Tan, Y. P. A., Liverneaux, P., & Wong, J. K. F. (2018). Current limitations of surgical robotics in reconstructive plastic microsurgery. *Frontiers in Surgery, 5*, 22. https://doi.org/10.3389/fsurg.2018.00022

University of Twente. (n.d.). *Mobile robotics: Applications and capacities.* https://fip.utwente.nl/mobile-robotics-applications-and-capacities/

Woo, Y. J., & Nacke, E. A. (2006). Robotic minimally invasive mitral valve reconstruction yields less blood product transfusion and shorter length of stay. *Surgery, 140*(2), 263–267. https://doi.org/10.1016/j.surg.2006.05.003

Xiao, X., & Tian, W. (2023). Mobile robot based on artificial potential field method: Path planning method research. *International Conference on Mechanical Engineering, Materials and Automation Technology, MBE, 17*(6), 6756–6774. https://doi.org/10.3934/mbe.2020352