Cost-Effective Real-Time Obstacle Detection and Avoidance for AGVs using YOLOv8 and RGB-D Sensors

Amar Medjaldi
Laboratory of Intelligent System (LSI), Faculty of Technology, University Sétif 1, Algeria
amar.medjaldi@univ-setif.dz

Yacine Slimani
Laboratory of Intelligent System (LSI), Faculty of Technology, University Sétif 1, Algeria
slimani_y09@univ-setif.dz (corresponding author)

Nora Karkar
Laboratory of Intelligent System (LSI), Faculty of Technology, University Sétif 1, Algeria
nora.karkar@univ-setif.dz

Received: 6 January 2025 | Revised: 4 February 2025 and 18 February 2025 | Accepted: 22 February 2025

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | DOI: https://doi.org/10.48084/etasr.10135

ABSTRACT

Recent advances in obstacle detection and avoidance technologies have significantly enhanced robotic navigation capabilities. This study presents a real-time obstacle detection and avoidance system leveraging YOLOv8 and RGB-D sensors. The system integrates Microsoft Kinect V1 to capture RGB and depth images, employing YOLOv8 for efficient real-time object detection and classification. Depth data are utilized to calculate object distances and positions, allowing accurate navigation decisions. Implemented on the Pioneer 3DX robot, the system demonstrates high efficiency, reliability, and adaptability. With a training dataset, the model achieves exceptional performance, attaining an accuracy of 92.6% across all object classes and a mAP@0.5 of 95%, However, the system was primarily tested in structured indoor environments, which may limit its generalization to unstructured outdoor settings. This cost-effective solution offers a practical approach to enhancing autonomous navigation and obstacle avoidance in real-world applications.

Keywords-YOLOv8; Pioneer 3DX robot; autonomous guided vehicles; real-time obstacle detection; RGB-D sensors; Microsoft Kinect V1

I. INTRODUCTION

In recent years, research on autonomous driving robots has gained significant attention in both academia and industry, driven by rapid advancements in information and communication technologies [1, 2]. This has led to a greater exploration of key technologies underlying this revolution, such as big data, artificial intelligence, and the Internet of Things (IoT). At the same time, industries that were at the forefront of previous industrial revolutions are receiving renewed focus. Smart factories, in particular, are spearheading a wave of innovation driving transformations across numerous domains. Researchers are closely monitoring these developments, as they are expected to bring substantial changes to a variety of sectors [3-5]. Autonomous navigation of robotic vehicles refers to their ability to move and operate without direct human intervention. This capability is achieved through the integration of various sensors, such as cameras, lidar, or sonar, which provide detailed data about the environment. These data are processed by embedded algorithms to determine the vehicle's position and orientation.

In environments such as manufacturing facilities, warehouses, and distribution centers, Autonomous Guided Vehicles (AGVs) are commonly used to transport raw materials and finished goods [6]. However, AGVs equipped only with optical sensors face challenges in navigating complex environments independently. Although cameras can capture features along a path, AGVs must also identify and avoid obstacles in real time, regardless of their shape, size, or color. The ability to recognize and adapt to obstacles in real time is critical for making decisions such as stopping, slowing down, or turning. Therefore, it is essential to develop AGVs capable of detecting and avoiding obstacles and recognizing objects in a human-like manner to operate autonomously and adaptively without human intervention [7, 8].

Convolutional Neural Networks (CNNs) have demonstrated exceptional performance in image recognition tasks, including object detection. The You Only Look Once (YOLO) algorithm is a CNN-based framework that is widely used in computer vision applications. YOLOv8 introduced significant improvements in accuracy and speed, making it an ideal choice for real-time obstacle detection in autonomous systems. This study presents a vision-based guidance system for autonomous vehicles using YOLOv8. The proposed approach was compared with existing deep learning-based control systems, emphasizing the advantages of the YOLOv8-based solution.

In the past decade, numerous techniques have been developed and applied in AGV navigation. In [9], these approaches were classified into two main types: local navigation, also referred to as conventional navigation, and global navigation, often relying on heuristic methods for pathfinding. In [10], an AGV motion control system was proposed based on magnetic tape navigation, where a magnetic tape on a floor serves as the guide. This system is notable for its ease of implementation, low cost, and high efficiency. Similarly, in [11], a navigation system was proposed for AGVs, using metal-line sensors for track guidance and RFID tags for localization. In [12], an AGV navigation approach used double magnetic nails, achieving reliable path tracking and stable accuracy through a fuzzy controller. In [13], a two-wheeled AGV with a laser-based (LIDAR) obstacle avoidance system was developed.

Graph-based heuristic algorithms have also gained prominence in AGV navigation, particularly for pathfinding tasks [9, 14]. Popular algorithms include A* (A-star), D* Lite, and Dijkstra's algorithm [15]. These methods have shown effectiveness in various environments, further solidifying their role in autonomous navigation. In recent years, image-based navigation and object detection have emerged as critical areas of research [16-18]. In [19], the Microsoft Kinect sensor was integrated with the YOLO algorithm for object detection and classification, using the depth sensor to measure object distances. Although effective, this approach relied heavily on object distance estimation and YOLO-based detection, which posed limitations. In [20], a monocular method was proposed that combined object detection and distance estimation using an R-CNN-based regression network. In [21], a collision warning system was capable of identifying specific objects and generating alerts when they were within hazardous proximity. This system relied on monocular cameras and deep learning models, but its performance was highly dependent on environmental factors such as lighting and weather conditions. Additionally, it required extensive training data and achieved only moderate accuracy (60%) in distance estimation. In contrast to these approaches, the proposed system leverages both RGB and depth sensors, enabling operation in diverse environments. It provides accurate distance measurements and precise object positions, allowing better navigation control.

The detection of traffic signs has been an important focus in autonomous navigation systems, given their critical role in ensuring safe and efficient driving. In [22], an improved Faster R-CNN was used for traffic sign detection. This study enhanced recognition speed by simplifying the Gabor wavelet through a regional suggestion algorithm. Similarly, in [23], Faster R-CNN was used as the base detection model while Generative Adversarial Networks (GANs) were incorporated for data augmentation, improving the robustness of the system under varying conditions. In [24], the Libra R-CNN algorithm was combined with a balanced feature pyramid to improve the detection of traffic signs. This approach allowed the system to handle challenges such as occlusions and variations in sign appearance, demonstrating its effectiveness in diverse scenarios.

II. SYSTEM DESIGN AND ARCHITECTURE

Building on these advances, the proposed system integrates both RGB and depth sensors to enhance performance in real-world applications. Unlike traditional methods that rely solely on RGB images, the proposed approach leverages depth information to achieve precise object distance measurements and accurate positioning of traffic signs. This dual-sensor setup not only improves detection accuracy but also enhances the system's ability to operate effectively in low-light and visually challenging environments. Furthermore, the algorithm heavily relies on depth sensors to address the limitations of RGB-based detection, particularly in scenarios where conventional methods struggle due to poor lighting or adverse environmental conditions. By combining RGB and depth data, the proposed system ensures versatility and reliability for real-time traffic sign detection and navigation. This fusion of modalities allows the system to better distinguish between obstacles and relevant road elements, reducing false positives in object classification.

A. *Architecture of Perception System*

The design of a perception system can vary depending on the specific application and requirements. As illustrated in Figure 1, the primary sensor used for data acquisition is an RGB-D camera, which captures both RGB and depth images. The depth images are particularly useful for obstacle identification, as they allow the system to detect barriers by analyzing variations in distances. This information is essential for tasks such as path planning and collision avoidance. Additionally, the depth image can calculate the distance to objects within a defined area, such as within a bounding box. By analyzing depth data in the designated region, the system determines how far an object is from the camera, a capacity critical for applications such as grasping or object manipulation, where distance awareness is crucial.

On the other hand, the RGB images captured by the camera are processed using the YOLO deep learning algorithm. This algorithm can recognize and classify objects in real time after being trained on extensive datasets. Consequently, the perception system performs object recognition tasks by identifying and labeling various items in the environment. The RGB image also facilitates the creation of bounding boxes around detected objects. These bounding boxes define the size and boundaries of recognized objects, allowing precise localization and tracking. Furthermore, the system can estimate an object's size or dimensions based on the bounding box, making it suitable for applications that require object measurements or size-based analysis.

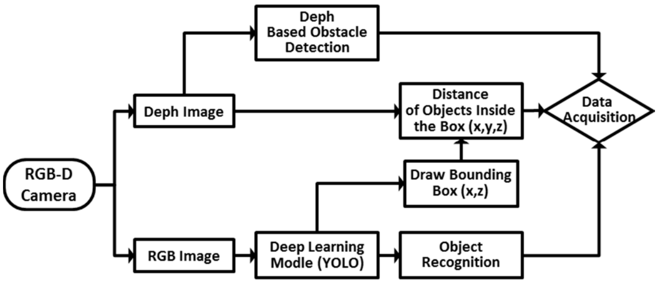


Fig. 1. System architecture.

In summary, the perception system in this study integrates RGB and depth data to enhance its capabilities. RGB images, processed by YOLOv8, are used for object recognition and bounding box generation, while depth images provide critical information for obstacle detection and distance estimation. Together, these components allow the perception system to collect and interpret data during the acquisition phase effectively. In this implementation, RGB and depth images were captured using Microsoft Kinect V1, improving the system's ability to perform real-time perception and navigation tasks.

B. *YOLO for Bounding Box and Identification*

YOLO is a modern object identification framework that offers numerous advantages over traditional methods. One of its key strengths is its approach to detection as a regression problem, which significantly improves speed. This eliminates the need for complex and time-consuming procedures, allowing the neural network to make rapid inferences on new images.

Furthermore, YOLO adopts a holistic perspective during both training and testing, processing the entire image at once. Unlike sliding-window and region-proposal-based approaches, YOLO integrates contextual information about object classes alongside their visual features, resulting in improved accuracy and efficiency. YOLO excels in recognizing common object representations, outperforming traditional methods such as Deformable Parts Models (DPM) [25]. YOLOv8 introduced additional improvements over YOLOv5, featuring enhanced functionality, including object identification, image classification, and image segmentation. A standout improvement is its anchor-free detection mechanism, which eliminates the reliance on preset bounding boxes. Instead, YOLOv8 determines the precise center of each object, reducing computational overhead and increasing speed [26].

YOLOv8 is available in five versions for different tasks: YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. The "x" version delivers the highest accuracy but is slower, whereas the "n" version is faster and smaller, ideal for resource-constrained applications. This study used the "Sign-detection dataset," which contains 8,316 images annotated with bounding boxes for various road signs (Figure 2) [27]. This dataset allowed effective training and testing of the YOLOv8 model, demonstrating its capacity to handle object detection in real-world scenarios.

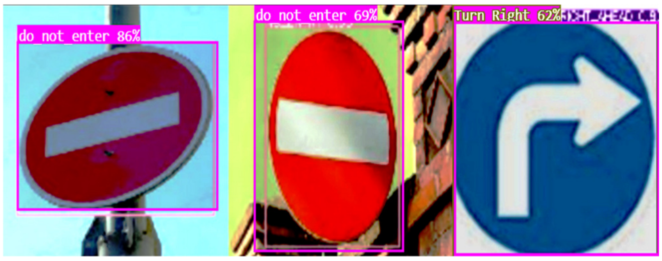


Fig. 2. Bounding boxes of road sign images.

C. *Object Detection and Distance Estimation*

The YOLO model, originally presented in [28], was employed to perform object detection and distance estimation. YOLOv8 introduces several architectural advances, including a pooling layer and a streamlined convolutional architecture to optimize computational efficiency. This model predicts three-dimensional tensors comprising bounding box coordinates, confidence scores, and class identifiers, which are essential for robust object detection.

YOLOv8 builds on the design principles of YOLOv5 and YOLOv7 ELAN to improve performance and flexibility. Its updated architecture includes a redesigned backbone network, an anchor-free detection head, and a novel loss function. These features improve the model's scalability and make it suitable for various applications. The architecture of YOLOv8 highlights its backbone, Feature Pyramid Network (FPN), and head components [29]. The backbone and neck sections of YOLOv8 draw inspiration from YOLOv7 ELAN, incorporating adjustments for enhanced performance. In the head section, YOLOv8 introduces a decoupled structure, separating classification and detection tasks. Additionally, it transitions from an anchor-based to an anchor-free detection mechanism, reducing dependency on preset bounding boxes and improving efficiency. The loss function employs TaskAlignedAssigner and DistributionFocalLoss to increase detection accuracy.

For distance estimation, YOLOv8 relies on depth data obtained from the RGB-D camera. However, the accuracy of depth measurements may vary due to external factors, such as lighting conditions and surface properties. To validate the depth estimation's reliability, tests were carried out under normal conditions. The results, compared to actual physical measurements, demonstrate the model's robustness in object detection and localization tasks.

D. *Image Processing*

The vision system processes visual data from the embedded camera by capturing images and applying a series of computational techniques. Several software libraries facilitate this process, including Torch3vision, VXL, Library Java VIS [30], LIT-Lib, and OpenCV [31]. OpenCV, an open-source library developed by Intel, is particularly notable for its extensive real-time visual processing capabilities, multi-language support, and cross-platform compatibility. With more than 2,500 optimized algorithms covering areas such as machine learning, object tracking, and augmented reality [32, 33], OpenCV was the main tool used for image processing.

To extract meaningful information, the system combines bounding box coordinates with depth images calibrated through Kinect V1. This integration allows accurate object localization and distance calculation, which are essential for depth-based applications. By analyzing the bounding box and depth data, the system determines the 3D spatial coordinates of detected objects (x, y, z), facilitating tasks such as object interaction and precise spatial mapping.

Using the acquired data, object class, distance, and position, the system generates velocity commands to control the Pioneer 3DX robot's movements. These commands regulate linear and angular velocities, allowing the robot to navigate effectively by moving forward, backward, rotating, or avoiding obstacles as required. The flowchart in Figure 3 outlines this process, demonstrating how the robot takes precise actions based on the collected information.

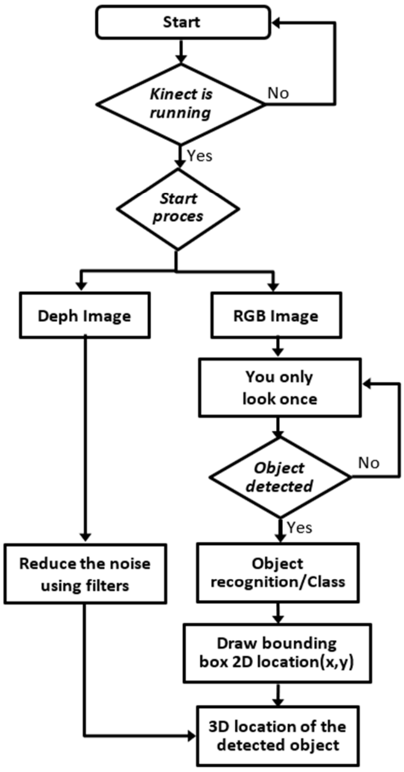


Fig. 3. Object recognition and location flowchart.

E. System Implementation

The aforementioned system was successfully implemented and rigorously tested using the Pioneer 3DX model, a wheeled mobile robot widely utilized in research and experimentation, as shown in Figure 4. The Pioneer 3DX is well-known for its versatility, maneuverability, and adaptability, making it an ideal platform for validating the functionality and performance of the developed system.

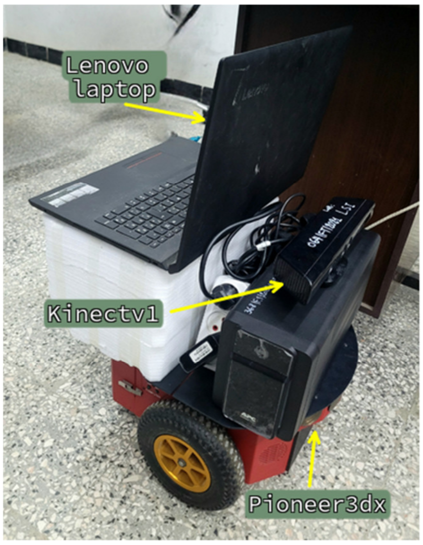


Fig. 4. System implementation of Kinect V1 on the robot Pioneer 3DX.

Extensive testing on this platform allowed for a comprehensive evaluation of the system's capabilities and its effectiveness in real-world scenarios. The experiments highlighted the system's ability to process data and control the robot with precision, demonstrating its reliability and consistency in practical applications. By using the Pioneer3DX robot, the experimental results were robust and reproducible, providing a solid foundation for scientific analysis and supporting further advances in the field. The platform's established reputation ensured that the results were both credible and aligned with industry standards, facilitating future applications and developments.

The Pioneer 3DX robot is equipped with two independently driven wheels (one on the right and one on the left) and a rear, off-centered orientable wheel that supports mechanical balance [34]. The driving wheels have a radius r, and are separated by an axle of length l.

The robot's position in the plane is described relative to an inertial reference frame $FR(O, X, Y)$ by the position vector [34]:

$$q = [x, y, \theta, \varphi_D, \varphi_G]^T \tag{1}$$

where x and y are the Cartesian coordinates of the robot's center, $\theta \in [-\pi, +\pi]$ is the orientation of the robot relative to the x-axis, and φ_D and φ_G are the angular displacements of the right and left wheels, respectively.

To integrate the Kinect V1 sensor for capturing RGB and depth image data, the "ros-noetic-freenect-launch" package was used within the Robot Operating System (ROS). The Kinect sensor was calibrated using a checkerboard to align the depth and RGB image coordinates accurately. For robot control, the ROSARIA node was employed, which provides an interface for Adept Mobile Robots, including the Pioneer3DX in this case.

The aim is to control the linear and angular speeds of the robot by controlling the linear speed of the right and left wheels [34]:

$$\begin{bmatrix} V_G \\ V_D \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{2} \\ 1 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \qquad (2)$$

where v and ω are the linear and angular speeds of the robot, and V_G and V_D are the linear speeds of the left and right wheels.

III. EXPERIMENTAL RESULTS

Experiments were carried out under two distinct conditions to evaluate the system's performance and effectiveness.

A. *Training the YOLOV8 Model*

The training phase of the YOLOv8 model was a critical component of the validation process, providing a foundation for evaluating its performance before deployment in real-world scenarios. This study used a dataset called "Road signs Computer Vision Project" [27], published in June 2023, and contained 30 different classes. The number of classes was reduced to 15, keeping only the classes needed to navigate the robot, while the number of images selected is 8316, divided into three sets for training, testing, and validation tasks. The training phase of YOLOv8 was a critical component of the validation process, providing a foundation for evaluating its performance before deployment in real-world scenarios. YOLOv8 introduces several advances that enhance its effectiveness in object detection and distance estimation tasks.

1) Overall Accuracy

The model achieved an accuracy of 92.6% across all object classes, demonstrating its reliability in detecting diverse elements.

2) mAP@0.5

With a mean Average Precision (mAP) of 95%, YOLOv8 surpassed other state-of-the-art algorithms, showcasing its potential for real-time object detection tasks. The training results, presented in Figure 5, highlight the model's robust performance.

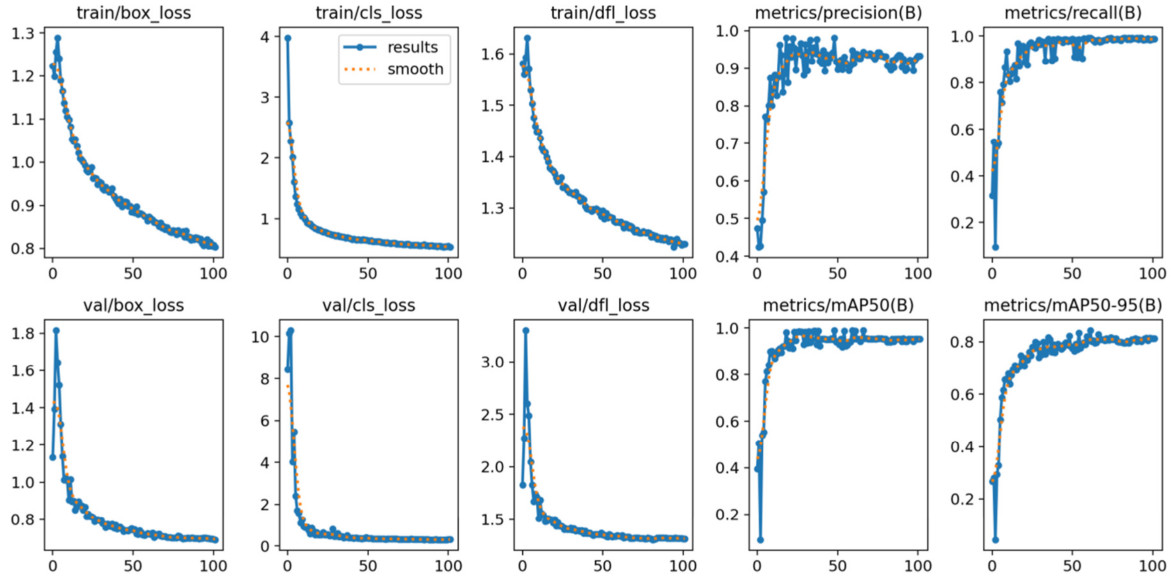


Fig. 5. Training the YOLOv8 model.

3) Loss Curve Convergence

Rapid and stable convergence of the loss curves reflects the effectiveness of the optimization process and the quality of the training data.

4) Distance Estimation Validation

To complement its object detection capabilities, YOLOv8 relies on depth data to estimate object distances. However, depth measurement accuracy can be influenced by factors such as lighting conditions, object surface properties, and sensor calibration. To validate the depth estimation accuracy, tests were carried out under normal conditions. Table I shows the results, comparing depth data with actual physical measurements.

TABLE I. ESTIMATED DISTANCE USING THE KINECT SENSOR

Test	**α1**	**α2**	**α3**	**α4**	**α5**	**α6**	**α7**	**α8**	**α9**
Depth distance (m)	0.51	0.73	0.85	1.47	1.70	2.01	2.25	3.01	4.21
Measured distance (m)	0.51	0.74	0.85	1.47	1.70	2.01	2.25	3.00	4.19

These results confirm the reliability of the YOLOv8 model for both object detection and distance estimation tasks. Minor deviations in depth measurements, such as in α2 and α8, remain within acceptable margins, further validating the robustness of the model in practical applications.

B. Experiments for YOLOV8 in Detecting Objects

The first condition involved an experiment using the YOLOV8 object detection model, which achieved successful object detection outcomes. During the experiment, the YOLOV8 model demonstrated its ability to detect objects accurately within captured images. This model proved effective in identifying and localizing various objects of interest. Leveraging the advanced algorithms and architecture of YOLOv8, the system achieved satisfactory results in terms of object detection accuracy and speed. The model's ability to handle real-time object detection tasks ensured timely and reliable identification of objects in the given context. The successful performance of the YOLOv8 model confirms its suitability for object detection applications, highlighting its potential in various scientific and practical domains.

1) First Experiment Results (Figure 6)

- The experiment utilized input from a Kinect camera, which provided visual data for further analysis and processing.
- The distance to the stop sign was accurately measured and recorded during the experiment. Utilizing the depth information obtained from the Kinect camera, the system was able to calculate the precise distance between the robot and the stop sign.
- The speed of the Pioneer 3DX robot was carefully controlled and monitored throughout the experiment. By adjusting the robot's velocity parameters, the system ensured controlled and consistent movement during data collection and subsequent analysis.
- After the robot was set in motion, new inputs were captured by the Kinect camera, providing updated visual data to the system. These new inputs allowed the system to continuously perceive the environment and adapt its operations accordingly.
- The updated distance from the stop sign was recalculated based on the new inputs received after the robot's movement. This allowed for real-time assessment and tracking of the distance between the robot and the stop sign as the robot traversed its environment.
- The speed of the Pioneer 3DX robot was dynamically adjusted in response to changing conditions and input. This allowed the system to regulate the robot's movement speed, ensuring safe and efficient navigation while maintaining the desired experimental parameters. These initial experimental results demonstrate the successful integration of the Kinect camera input, accurate distance measurement, real-time adaptation to new inputs, and dynamic control of the robot's speed. This sets the foundation for further analysis and exploration in subsequent experiments and reinforces the potential of such technologies in scientific research and practical applications. According to Figure 7, the robot moved at its initial pace for a distance of 1 m before detecting a stop sign, at which point it obeyed the command and came to a halt. MobileSim was used to sketch the robot's path to test the method by placing numerous immobile obstacles in the robot's path.

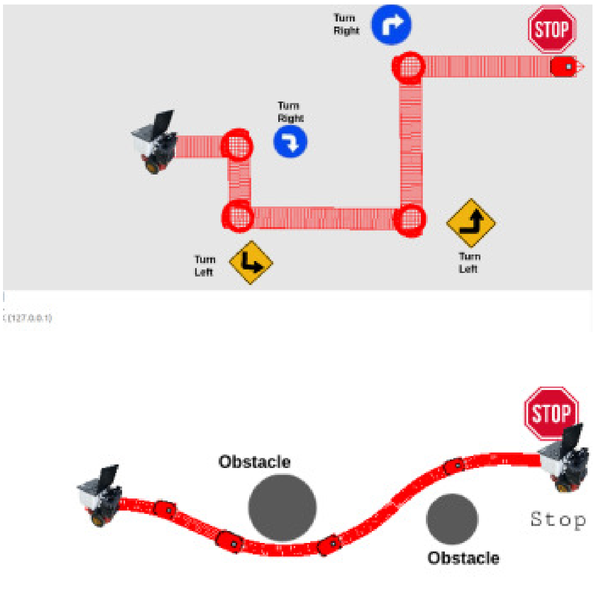
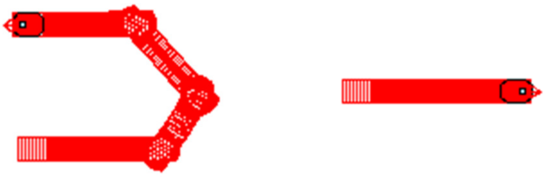


Fig. 6. Path taken by the robot to avoid obstacles.

2) Experiments for Objects that Cannot be Detected

In this experiment, the system was tested against fast-moving objects and walls that cannot be detected by YOLOv8, as well as in low light conditions. The robot was kept safe and avoided endangering people or the environment.



(a) (b)

Fig. 7. Second experiment's results against undetected obstacles.

In the experimental setup, the robot behavior was programmed to respond to the proximity of obstacles within a range of 0.6 m. When an obstacle approached within this specified range, the robot initiated a stop command and a turning motion to find a clear path, as shown in Figure 7(a). This adaptive strategy aimed to facilitate obstacle avoidance, particularly in the context of navigating near walls.

Using this reactive approach, the robot effectively stopped its forward movement when it detected an obstacle within the predetermined range, as shown in Figure 7(b). This temporary pause allowed the robot to assess the environment and determine the optimal direction for evasive action. By actively searching for an unobstructed path through the turning motion, the robot aimed to identify a viable route to resume its intended trajectory.

These results highlight the importance of employing adaptive techniques to enable autonomous systems, such as the robot in this experiment, to respond intelligently to environmental changes. By integrating obstacle detection, decision-making algorithms, and appropriate motion control, robots can navigate complex environments while minimizing the likelihood of collisions and achieving their intended goals.

IV. CONCLUSION

The experimental results confirmed the suitability of the proposed system for deploying AGVs utilizing RGB-D sensors. The proposed approach provides flexibility and adaptability for various tasks without relying on predefined maps or expensive metal line sensors, as previous studies [11]. What sets this work apart is the integration of YOLOv8 with RGB-D sensors for real-time obstacle detection and avoidance, offering a balance between accuracy, computational efficiency, and cost-effectiveness**,** being an alternative to the more expensive LIDAR sensors used in other approaches [13]. By incorporating the YOLO object detection algorithm, the system ensures efficient object detection across different distances, enhancing AGV control and ensuring safe and efficient operations.

Integration of RGB-D sensors allows for precise detection and positioning of objects, significantly improving the overall maneuverability and control of AGVs. Additionally, the YOLO object detection algorithm improves the reliability and accuracy of the system, surpassing the capabilities of earlier methods [19]. This combination of RGB-D sensors and the YOLO algorithm highlights the system's potential for practical AGV applications, delivering cost-effective and accurate object detection over varying distances. These findings contribute to the advancement of autonomous navigation systems for AGVs and open opportunities for further research and development in this field.

ACKNOWLEDGMENT

This research was supported by DGRSDT - MESRS, Algeria.

REFERENCES

[1] M. Khatun, M. Patel, R. Jung, and M. Glas, "A simulation-based analysis for a transverse guidance for highly automated driving functions," in *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Ankara, Turkey, Jun. 2021, pp. 1–6, https://doi.org/10.1109/HORA52670.2021.9461313.

[2] G. Ullrich and T. Albrecht, Automated Guided Vehicle Systems: A Guide - With Practical Applications - About The Technology - For Planning. Springer Fachmedien, 2023.

[3] L. Yan, H. Duan, and X. Yu, Eds., Advances in Guidance, Navigation and Control: Proceedings of 2020 International Conference on Guidance, Navigation and Control, ICGNC 2020, Tianjin, China, October 23–25, 2020, vol. 644, 2022.

[4] A. Abdulhafez and G. Harit, "Guide Me: Recognition and Servoing on Mobiles," *Arabian Journal for Science and Engineering*, vol. 43, no. 12, pp. 7359–7372, Dec. 2018, https://doi.org/10.1007/s13369-018-3084-7.

[5] J. Betz *et al.*, "Autonomous Vehicles on the Edge: A Survey on Autonomous Vehicle Racing," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 458–488, 2022, https://doi.org/10.1109/OJITS.2022.3181510.

[6] H. Fazlollahtabar and M. Saidi-Mehrabad, *Autonomous Guided Vehicles: Methods and Models for Optimal Path Planning*, vol. 20. Springer International Publishing, 2015.

[7] J. Chen, S. Wang, T. Zhou, L. Xiong, and X. Xing, "Study on Safety Analysis Method for Take-over System of Autonomous Vehicles," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, Las Vegas, NV, USA, Oct. 2020, pp. 1972–1977, https://doi.org/10.1109/IV47402.2020.9304599.

[8] K. Wang *et al.*, "The adaptability and challenges of autonomous vehicles to pedestrians in urban China," *Accident Analysis & Prevention*, vol. 145, Sep. 2020, Art. no. 105692, https://doi.org/10.1016/j.aap.2020.105692.

[9] M. Aizat, A. Azmin, and W. Rahiman, "A Survey on Navigation Approaches for Automated Guided Vehicle Robots in Dynamic Surrounding," *IEEE Access*, vol. 11, pp. 33934–33955, 2023, https://doi.org/10.1109/ACCESS.2023.3263734.

[10] F. Liu, X. Li, and Y. Wang, "Design of Automatic Guided Vehicle Motion Control System Based on Magnetic Navigation," in *2018 Chinese Control And Decision Conference (CCDC)*, Shenyang, China, Jun. 2018, pp. 4775–4779, https://doi.org/10.1109/CCDC.2018.8407957.

[11] V. Rubanov, D. Bushuev, E. Karikov, A. Bazhanov, and S. Alekseevsky, "Development a low-cost navigation technology based on metal line sensors and passive RFID tags for industrial automated guided vehicle," *Journal of Engineering and Applied Sciences*, vol. 15, no. 20, pp. 2291–2297, 2020.

[12] Q. F. Yan, H. Yan Hu, T. P. Hang, and Y. Si Fu, "Path Tracking of INS AGV Corrected by Double Magnetic Nails Based on Fuzzy Controller," in *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Chongqing, China, Oct. 2019, pp. 1732–1735, https://doi.org/10.1109/IMCEC46724.2019.8984131.

[13] Z. Ma, O. Postolache, and Y. Yang, "Obstacle Avoidance for Unmanned Vehicle based on a 2D LIDAR," in *2019 International Conference on Sensing and Instrumentation in IoT Era (ISSI)*, Lisbon, Portugal, Aug. 2019, pp. 1–6, https://doi.org/10.1109/ISSI47111.2019.9043674.

[14] G. Fragapane, R. de Koster, F. Sgarbossa, and J. O. Strandhagen, "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda," *European Journal of Operational Research*, vol. 294, no. 2, pp. 405–426, Oct. 2021, https://doi.org/10.1016/j.ejor.2021.01.019.

[15] A. N. A. Rafai, N. Adzhar, and N. I. Jaini, "A Review on Path Planning and Obstacle Avoidance Algorithms for Autonomous Mobile Robots," *Journal of Robotics*, vol. 2022, no. 1, 2022, Art. no. 2538220, https://doi.org/10.1155/2022/2538220.

[16] Y. Chen, W. Zheng, Y. Zhao, T. H. Song, and H. Shin, "DW-YOLO: An Efficient Object Detector for Drones and Self-driving Vehicles," *Arabian Journal for Science and Engineering*, vol. 48, no. 2, pp. 1427–1436, Feb. 2023, https://doi.org/10.1007/s13369-022-06874-7.

[17] M. Ding *et al.*, "Learning Depth-Guided Convolutions for Monocular 3D Object Detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 11669–11678, https://doi.org/10.1109/CVPR42600.2020.01169.

[18] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical Depth Distribution Network for Monocular 3D Object Detection," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 8551–8560, https://doi.org/10.1109/CVPR46437.2021.00845.

[19] D. H. Dos Reis, D. Welfer, M. A. De Souza Leite Cuadros, and D. F. T. Gamarra, "Mobile Robot Navigation Using an Object Recognition Software with RGBD Images and the YOLO Algorithm," *Applied Artificial Intelligence*, vol. 33, no. 14, pp. 1290–1305, Dec. 2019, https://doi.org/10.1080/08839514.2019.1684778.

[20] Y. Zhang *et al.*, "A regional distance regression network for monocular object distance estimation," *Journal of Visual Communication and Image Representation*, vol. 79, Aug. 2021, Art. no. 103224, https://doi.org/10.1016/j.jvcir.2021.103224.

[21] H. Saleh, S. Saleh, N. T. Toure, and W. Hardt, "Robust Collision Warning System based on Multi Objects Distance Estimation," in *2021 IEEE Concurrent Processes Architectures and Embedded Systems Virtual Conference (COPA)*, San Diego, CA, USA, Apr. 2021, pp. 1–6, https://doi.org/10.1109/COPA51043.2021.9541452.

[22] F. Shao, X. Wang, F. Meng, J. Zhu, D. Wang, and J. Dai, "Improved Faster R-CNN Traffic Sign Detection Based on a Second Region of Interest and Highly Possible Regions Proposal Network," *Sensors*, vol. 19, no. 10, Jan. 2019, Art. no. 2288, https://doi.org/10.3390/s19102288.

[23] W. Huang, M. Huang, and Y. Zhang, "Detection of Traffic Signs Based on Combination of GAN and Faster-RCNN," *Journal of Physics: Conference Series*, vol. 1069, no. 1, Dec. 2018, Art. no. 012159, https://doi.org/10.1088/1742-6596/1069/1/012159.

[24] Z. Zhao, H. Liu, and D. Cao, "Improved Traffic Sign Detection Algorithm Based on Libra R-CNN," *Journal of Mechanical Engineering*, vol. 57, no. 22, pp. 255–265.

[25] G. Jie, Z. Honggang, C. Daiwu, and Z. Nannan, "Object detection algorithm based on deformable part models," in *2014 4th IEEE International Conference on Network Infrastructure and Digital Content*, Beijing, China, Sep. 2014, pp. 90–94, https://doi.org/10.1109/ICNIDC.2014.7000271.

[26] A. Tripathi, V. Gohokar, and R. Kute, "Comparative Analysis of YOLOv8 and YOLOv9 Models for Real-Time Plant Disease Detection in Hydroponics," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 17269–17275, Oct. 2024, https://doi.org/10.48084/etasr.8301.

[27] "Road signs Computer Vision Project," *Roboflow*. https://universe.roboflow.com/mjrob/road-signs-ohan1.

[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[29] J. Motsch, S. Benammar, and Y. Bergeon, "Interior mapping of a building: A real-life experiment with Microsoft Kinect for Windows v1 and RGBD-SLAM," in *2017 International Conference on Military Technologies (ICMT)*, Brno, Czech Republic, May 2017, pp. 728–732, https://doi.org/10.1109/MILTECHS.2017.7988852.

[30] X. Hu *et al.*, "Real-time detection of uneaten feed pellets in underwater images for aquaculture using an improved YOLO-V4 network," *Computers and Electronics in Agriculture*, vol. 185, Jun. 2021, Art. no. 106135, https://doi.org/10.1016/j.compag.2021.106135.

[31] M. Cazorla and D. Viejo, "JavaVis: An integrated computer vision library for teaching computer vision," *Computer Applications in Engineering Education*, vol. 23, no. 2, pp. 258–267, 2015, https://doi.org/10.1002/cae.21594.

[32] K. Pulli, A. Baksheev, K. Kornyakov, and V. Eruhimov, "Real-time computer vision with OpenCV," *Communications of the ACM*, vol. 55, no. 6, pp. 61–69, Jun. 2012, https://doi.org/10.1145/2184319.2184337.

[33] G. R. Bradski and A. Kaehler, *Learning OpenCV: computer vision with the OpenCV library*. O'Reilly, 2011.

[34] S. Khesrani, "Contribution à la modélisation et à la commande des systèmes robotiques," Ph.D. dissertation, University Ferhat Abbas - Sétif 1, Algeria, 2023.