applied sciences

MDPI

Article

Autonomous Robotic Navigation Approach Using Deep Q-Network Late Fusion and People Detection-Based Collision Avoidance

Carlos Daniel de Sousa Bezerra [1,*], Flávio Henrique Teles Vieira [1,2] and Daniel Porto Queiroz Carneiro [2]

[1] Institute of Informatics (INF), Federal University of Goias (UFG), Goiás 74690-900, Brazil; flavio_vieira@ufg.br
[2] School of Electrical, Mechanical and Computer Engineering (EMC), Federal University of Goiás (UFG), Goias 74690-900, Brazil; dcarneiro@discente.ufg.br
* Correspondence: carlosdbez@discente.ufg.br

Abstract: In this work, we propose an approach for the autonomous navigation of mobile robots using fusion the of sensor data by a Double Deep Q-Network with collision avoidance by detecting moving people via computer vision techniques. We evaluate two data fusion methods for the proposed autonomous navigation approach: Interactive and Late Fusion strategy. Both are used to integrate mobile robot sensors through the following sensors: GPS, IMU, and an RGB-D camera. The proposed collision avoidance module is implemented along with the sensor fusion architecture in order to prevent the autonomous mobile robot from colliding with moving people. The simulation results indicate a significant impact on the success of completing the proposed mission by the mobile robot with the fusion of sensors, indicating a performance increase (success rate) of ≈27% in relation to navigation without sensor fusion. With the addition of moving people in the environment, deploying the people detection and collision avoidance security module has improved about the success rate by 14% when compared to that of the autonomous navigation approach without the security module.

Keywords: DQN; reinforcement learning; autonomous navigation; sensor fusion

Citation: de Sousa Bezerra, C.D.; Teles Vieira, F.H.; Queiroz Carneiro, D.P. Autonomous Robotic Navigation Approach Using Deep Q-Network Late Fusion and People Detection-Based Collision Avoidance. *Appl. Sci.* **2023**, *13*, 12350. https://doi.org/10.3390/app132212350

Academic Editor: Sungho Kim

Received: 22 October 2023
Revised: 7 November 2023
Accepted: 10 November 2023
Published: 15 November 2023

1. Introduction

Currently, the world is going through a digital transformation known as the fourth industrial revolution and also called Industry 4.0. The term "digital transformation" means relevant changes that are taking place in society due to the rapid adoption of technology. In this new world scenario, large companies are racing to invest in digital tools, such as artificial intelligence, embedded electronics, connectivity, and sensing, to leverage their profits and competitiveness.

The concept of Industry 4.0 emerged in Germany [1] and is based on nine pillars that support its thesis and revolutionize the industrial sector: Big Data Analysis, Autonomous Robotics, Simulation, Systems Integration, Internet of Things (IoT), CyberSecurity, Cloud Computing, Additive Manufacturing, and Augmented Reality. One of the main characteristics of the Industry 4.0 is the use of pillar technology for real-time monitoring and reduction of human intervention in production and, consequently, obtaining greater precision in certain tasks and safety for processes.

Industry 4.0 technology allows factories to use Autonomous Mobile Robots (AMRs) on their assembly lines. Unlike manually guided vehicles, AMRs do not need real-time human control or pre-planned routes to locate and move around. They are equipped with visual and non-visual sensors, as well as intelligent control architecture to handle dynamic environments [2]. These mobile robots can replace or assist the mediation of human labor within the production processes, especially in risk areas of the manufacturing process, where work safety can be compromised, generating accidents and costs for the entrepreneur.

The ability of autonomous navigation and decision making of these AMRs arises in the function of three major areas of mobile robotics: perception, planning, and action [3]. The sensors present in an AMR are used to sense (perceive) the navigation conditions. The planning area, on the other hand, refers to a list of algorithms embedded in the processors of these robots and are used to decide which path should be followed during the execution of the mission. The actuators are responsible for activating the electromechanical elements that move this robot, such as wheel motors and joints, among others.

According to [4], an important consideration in the area of perception is the proper choice of the type of sensor to be used, as well as its ideal configuration. These sensors will be used to mimic the human ability to feel and make decisions. In this way, grouping, or merging, sensors can be a good strategy for optimizing these capabilities, thus expanding the agent's ability to move properly within an environment. Currently, Machine Learning is an important ally in AMR decision making.

Machine Learning, a subfield of Artificial Intelligence (AI), is an area of study with great potential for application in learning tasks in autonomous systems. This science can be divided into three categories of problems: (a) Supervised Learning, (b) Unsupervised, and (c) Reinforcement Learning (RL).

RL is a powerful machine learning tool for AMR navigation. According to [5], this technique can be applied in sequential systems, where, for example, the agent (a robot) is an environment (for example a production process), possesses certain states (its positions), and the agent performs actions. The actions of this agent generate feedback signals that measure the quality of the action in the environment; this value is called the reward. The objective is to maximize the expected reward value for a given state and action pair. In summary, the agent tends to learn by trial and error and will always seek to maximize the good rewards [6]. One of the advantages of RL is the possibility of navigating a mobile vehicle without pre-planning a route, as the agent learns by iteration and is able to generalize new actions with a possible change in the environment.

In this way, the current research work aims to investigate sensory fusion techniques for broad applications involving the navigation of AMRs. Thhe autonomous navigation approach is proposed, using the Deep Reinforcement Learning algorithm combined with the following sensory fusion strategies: Interactive Model [7] and Late Fusion [8,9]. Sensory fusion of images is performed and obtained by an RGB-D camera with non-visual navigation sensors, both coupled to the AMR. Sensory fusion, which uses artificial neural networks, generates the unified control signal processing for agent training via reinforcement learning, specifically using the DDQN algorithm (Double Deep Q-Networks). The security module is a separate trained algorithm, specifically using supervised learning and embedded in the sensor fusion architecture. Its objective is to avoid robot collisions when the navigation environment becomes dynamic.

The main hypothesis of this article is based on the need for precision in certain navigation missions. It is assumed that these missions cannot be solved only with sensors linked to vision, even with the use of efficient control techniques, such as reinforcement learning. That is, non-visual sensors would increase the perception capacity of AMR through multimodal fusion techniques. For this, the impact of two modern fusion techniques on mission development is evaluated.

The justification for the elaboration of this work is the incipience in the technical literature of sensory fusion methods allied to reinforcement learning for autonomous navigation purposes. Most of the methods are applied in isolated tasks, such as the classification of objects in the scene and identification of routes and pedestrians, among others. In addition, the work uses recent methods that were not previously applied to the robotics area, such as the Iterative method that originated in the medical area for classifying diseases.

The article is organized as follows: Section 2 presents related works, and Section 3 presents the proposed problem, as well as the implementation methodology. Section 4 contains the results. Finally, Section 5 concludes the current study and presents future studies.

2. Related Works

This section presents some recent works related to the state of the art in the area of sensor (or data) fusion applied to autonomous mobile robots (AMRs).

According to [10], Machine Learning techniques can be used to fuse sensory information. It is worth noting that the area of sensory fusion (or multimodal) is not only applied to AMRs. Both sensor fusion and information fusion can be defined as the process of management and treatment of data from various types of sources to improve some specific criteria for decision tasks [4].

The authors in [4] present a comprehensive literature review on the methods of sensor fusion in autonomous vehicles, specifically in the areas of perception, localization, and mapping using machine learning. The authors highlight recent applications in the literature in pedestrian and road detection tasks through the fusion of the LIDAR sensor with images. Other applications are also highlighted, such as positioning with proprioceptive sensors, such as GPS, merged with images. In addition, they present the categorization of the fusion methods existing in the literature. A point that draws attention in the literature review is the absence of reinforcement learning techniques combined with data fusion; the techniques presented are supervised learning. The authors in [11] explain that the area of multimodal fusion applied to robotic systems is still emerging, which reinforces the thesis that in the area of reinforcement learning, this application is still incipient as well.

Currently, the medical field, allied with computer and artificial intelligence researchers, employ the use of data fusion to increase the accuracy of several diagnoses that use images (magnetic resonance, X-ray, among others) and tabular data, such as blood tests and a patient's gender and sex. The method described in [7] was called Interactive and uses a channel wise multiplication technique to create attention mechanisms from tabular data in convolutional neural networks, thus unifying the processing of tabular data with images resulting from clinical examinations.

In [12], the authors provide a comprehensive review of path-planning strategies for mobile robot navigation. They discuss different algorithms, their applications, and their effectiveness in varied environments, highlighting techniques for overcoming obstacles and achieving goals efficiently. However, our work is distinguished by not only reviewing or comparing existing methods, but also by proposing an innovative approach that integrates sensor data fusion with reinforcement learning techniques, specifically through the implementation of a Double Deep Q-Network (DDQN). Furthermore, while [12] focuses on path-planning strategies in general, our study advances the field by incorporating a real-time collision detection and avoidance module, particularly for people in motion, which is essential for safe operation in dynamic and populated environments.

The authors in [13] introduce "Neural RRT*", a methodology that combines neural networks with the RRT* algorithm to improve optimal path planning. Their results indicate that machine learning can be effectively integrated into path-planning algorithms to optimize routes in complex environments. In contrast, our study employs sensory data fusion in conjunction with DDQN for autonomous navigation, focusing on detecting and avoiding collisions with moving humans, a practical and critical challenge not directly addressed by "Neural RRT*". Furthermore, our approach is validated in a dynamic environment that simulates real conditions, where the robot's performance is improved not only in terms of path planning, but also in safe interactions with the shared environment.

The work of [14] focuses on path planning for UAVs (Unmanned Aerial Vehicles), using various optimization approaches to address the unique challenges of aerial navigation. The authors offer a detailed analysis of methodologies to optimize the path of UAVs, considering factors such as flight dynamics and airspace restrictions. In contrast, our study explores the terrestrial autonomous navigation of mobile robots, a distinct area with its complexities, such as interaction with humans and the need to avoid obstacles in real-time. Our contribution is particularly notable in the implementation of data fusion methods to improve robot sensory perception and the introduction of a safety module that allows harmonious coexistence with humans in shared environments.

In [15], the use of deep reinforcement learning for fusing multimodal data in the context of action recognition is explored. Their innovative method demonstrates the potential of DRL to effectively combine different types of sensor data, such as visual and inertial, to improve recognition accuracy. While their research focuses on the application of DRL to action recognition, our work also includes the use of multimodal data fusion in autonomous navigation. We employ similar deep learning techniques not for recognition tasks, but to enhance the decision-making process of mobile robots in complex and dynamic environments.

In [16], an application of force vision sensor fusion in a learning-based approach for robots that perform the task of pulling doors is presented. Their research highlights the advantages of sensor fusion in improving the interaction of robots with physical objects in their environment. While their application is specific to door-pulling tasks, our research takes a broader approach to autonomous navigation. The fusion strategies we develop are not limited to specific tasks, but are designed to improve the general mobility and safety of robots in human-centric environments.

While the authors in [17] focus on sensor fusion applications in high-speed scenarios such as the Indy Autonomous Challenge 2021 and CES 2022, by employing classic methods such as the Extended Kalman Filter (EKF), our study advances the field by integrating cutting-edge techniques in deep learning and reinforcement learning. Sensor fusion in our work is not limited to theories and simulations, but extends to encompass practical applications in robotic systems operating in dynamic environments, a crucial direction for modern autonomous robotics.

The authors in [18] discuss the problem of motion-planning efficiency for mobile robots in complex environments such as mazes and S-shaped corridors by employing a Generalized Voronoi Diagram (GVD)-based heuristic path planning method to guide the sampling of Rapidly-exploring Random Trees (RRTs). This method enhances the feature extraction of free space and heuristic path-planning, contributing to real-time motion planning. In contrast, our work focuses on sensor data fusion from GPS, IMU, and RGB-D cameras with deep learning techniques for autonomous navigation of mobile robots, further complemented by a collision avoidance module for moving people detection, increasing safety and efficiency in dynamic environments.

In [19], a Deep Reinforcement Learning (DRL) solution for UAV path planning in dynamic environments is proposed, focusing on UAV survival under threats like radar detection and missile attacks. Their approach, based on Dueling Double Deep Q-Networks (D3QN), uses situation maps to make navigation decisions. While this work focuses on dynamic aerial environments and external threats, our work aims for autonomous terrestrial navigation, utilizing sensor data fusion and computer vision to safely interact in environments with the presence of people.

In [20], a systematic review of AI techniques applied to trajectory planning for UAV swarms is provided. They highlight the growing trend of publications and the evolution of predominant techniques. While their paper offers an overview of AI techniques in a swarm context, our work specifically applies deep learning to sensor data fusion in a single mobile robot, focusing on autonomous and safe interaction in environments that may be shared with humans, showing gains in mission success rates with the implementation of a safety module.

The work of [21] focuses on multi-story autonomous navigation for service robots, emphasizing the localization of elevator buttons through computer vision. They deploy a multi-story SLAM system that detects elevator buttons in real-time, enabling the robot to autonomously navigate between floors of a building. This work specializes in indoor navigation challenges and employs computer vision algorithms to overcome common issues in elevators. In comparison, our work is concerned with mobile robots in varied environments, employing sensor data fusion and deep learning to ensure safe and efficient navigation, especially in the presence of moving humans.

In [22], autonomous UAV path planning for target coverage problems using artificial intelligence methods, including genetic algorithms, ant colony optimization, Voronoi diagrams, and clustering methods, are discussed. They focus on enhancing the genetic algorithm's initial population to expedite convergence and prevent UAV crashes by integrating Voronoi vertices and cluster centers as navigational waypoints. In contrast, our work concentrates on the autonomous navigation of ground-based mobile robots through sensor data fusion using a Double Deep Q-Network and a collision avoidance module that detects moving people using computer vision techniques. While they optimize UAV path efficiency and terrain collision avoidance, our study enhances sensor fusion applications for mobile robots, with a demonstrated improvement in mission success rate when integrating safety modules that consider dynamic human presence.

The work of [8] offers a thorough review of multi-modal perception techniques for autonomous driving. Our research applies fusion methodologies directly within the framework of reinforcement learning for autonomous robot navigation. Our contribution specifically addresses the integration of sensory inputs (GPS, IMU, and RGB-D camera) to facilitate the robot's decision-making process in real-time scenarios with pedestrian traffic. We provide empirical evidence of our method's success, showcasing approximately a 27% increase in mission completion rates through sensor fusion and a further 14% improvement with the addition of our collision avoidance module. This not only illustrates the practical implementation of sensor fusion, but also highlights its importance in the context of interactive environments where safety and adaptability are paramount.

The distinction of our approach lies in the use of deep neural networks and reinforcement learning algorithms that are essential for the intelligent and adaptive navigation of robots, paving the way for significant advances in interaction and safety in environments shared with humans. While comprehensive implementation remains a goal for the future, current phases of our research already demonstrate the feasibility and potential of our methods. Furthermore, we even slightly explore in this paper Sim-to-Real knowledge transfer as a proof of concept for the practical applicability of our automated learning strategies, confirming the adaptability and efficiency of the proposed algorithms. This aspect highlights the relevance of our contribution to autonomous navigation, where the challenges are not only theoretical, but also practical and operational.

The next section describes some technologies that allow the implementation of efficient navigation systems, such as reinforcement learning and sensor fusion.

3. Enabling Technologies for Autonomous Navigation

This section presents modern algorithms for controlling AMRs using artificial neural networks, specifically using reinforcement learning. Initially, the DQN (Deep Q-Networks) and DDQN (Double Deep Q-Networks) techniques, known as value-based methods, are described. Subsequently, some techniques related to sensor fusion are presented.

3.1. Introduction to Reinforcement Learning

Reinforcement Learning (RL) is an area of machine learning that captivates with its promise of enabling agents to learn optimal behaviors through trial-and-error interactions with an environment, aiming to maximize a cumulative reward [23]. The allure of RL lies in its versatility; it can be applied to a wide range of problems, from strategic games like Go and chess to real-world applications such as autonomous driving and robotics [24,25].

At its core, RL involves an agent that makes decisions, an environment that responds to those decisions and provides rewards, and a goal that the agent strives to achieve. The agent's mission is to learn a policy: a strategy for choosing actions based on the current state of the environment that will accumulate the highest possible reward over time.

The beauty of reinforcement learning comes from its adaptability and feedback-driven learning process. Unlike other machine learning paradigms, an RL agent learns from the consequences of its actions rather than from explicit instruction, which resembles the way living beings learn from experience in the real world. This learning process makes RL a

powerful tool for developing sophisticated and autonomous systems that improve their performance with experience [23].

Now, let us delve into a specific RL methodology known as Deep Q-Networks (DQN) [24], which combines traditional reinforcement learning with deep learning's ability to handle high-dimensional sensory input.

3.2. Deep Q-Networks (DQN)

The reinforcement learning system aims to provide knowledge to a given agent through interactions in an environment. The qualities of this agent's actions are measured through the reward function. The state-action value function $Q(s, a)$ quantifies the rewards obtained by this agent in relation to its states and actions, both in the short and in the long term. In this way, the agent learns π action policies that maximize the value function. The function $Q(s, a)$ is a variation of the temporal learning algorithm proposed by Bellman and is given by:

$$Q^{\pi}(s, a) = r + \gamma \ \max Q^{\pi}(s', a'), \tag{1}$$

where r is the immediate obtained reward and γ a discount factor for future rewards; s, s', a, and a' are the states and actions of the present and future, respectively.

The DQN algorithm, a type of Reinforcement Learning (RL) technique, incorporates the parameters of a neural network into its algorithm, with synaptic weights (θ) and bias. Normally, the network used is deep (more than three layers) and aims to estimate the optimal $Q(s, a)$ function, displayed in Equation (1). The DQN does not need a tabular representation to store and map agent states and actions, like traditional Q-Learning. In an environment with many states and actions, for example, images captured by a camera, tabular representations are not feasible. The DQN state action mapping is provided by the neural network itself. In addition, unlike supervised learning, in DQN, the data are dynamic and updated at each step of the algorithm's integration.

$$Q(s, a) \approx \hat{Q}(s, a, \theta) \tag{2}$$

The DDQN algorithm, proposed by [26], is a variation of the traditional DQN. This algorithm helps to reduce stock overestimation problems that may arise as a result of the regression process. Thus, the DDQN uses one more neural network than the traditional DQN, called the Q_{target} network, which helps in the process of estimating future actions.

The overestimation problem in the traditional DQN algorithm occurs because the algorithm always chooses the maximum value $Q(s', a')$, as expressed in Equation (1). However, it does not always obtain the maximum value of $Q(s', a')$, meaning that it obtains the best policy, since the DQN deals with estimates, and this maximization process contains uncertainties mainly at the beginning of the algorithm exploration. The consequence of overestimation is the development of low quality policy and instability in training.

In DDQN, we have two independent neural networks: the Q^{π} and Q_{target} network. The modified TD Equation is described below:

$$Q^{\pi}(s, a) = r + \gamma \ (Q_{target}(s', \ argmax_{a'} \ Q^{\pi}(s', a')) \tag{3}$$

As it can be seen from Equation (3), the function $Q^{\pi}(s', a')$ is applied to a neural network responsible for evaluating the best action a. The function Q_{target} is applied to another neural network to estimate the expected value using the best action estimated by $Q^{\pi}(s', a')$.

In the realm of reinforcement learning, the distinction between on-policy and off-policy methods is pivotal in the design of algorithms for training agents. Off-policy methods, such as the DDQN, offer significant advantages that align with our objectives of creating a robust and computationally efficient navigation system for autonomous robots.

The DDQN, an extension of the standard DQN algorithm, inherits its predecessor's power but with an improved stability in learning. This stability stems from the DDQN's

ability to decouple the selection and evaluation of the action, which mitigates the overestimation of Q-values that often plagues the original DQN. This characteristic is particularly beneficial in complex decision-making environments, where overestimation can lead to suboptimal policy development.

One of the primary reasons for selecting an off-policy method like DDQN is its sample efficiency [26]. Off-policy algorithms can learn from the experiences of past policies, utilizing data more effectively by learning from observations generated by a behavior policy not necessarily aligned with the current policy being improved [27]. This reuse of experience accelerates learning and makes the most of each interaction with the environment.

Moreover, the off-policy nature of DDQN allows for greater flexibility in learning. It can learn from experiences collected by other agents or from human demonstrations, facilitating a more diverse and comprehensive learning process. This is particularly advantageous when dealing with the Sim-to-Real transfer, as it enables the agent to benefit from simulated experiences, as well as real-world interactions.

Furthermore, DDQNs are computationally similar to DQNs, making them an attractive choice for applications where computational resources are a concern. They do not require significant additional resources compared to DQNs, yet they provide a marked improvement in learning performance and policy quality.

3.3. Deep Sensor Fusion

Data Fusion, or sensory information, is a skill constantly used by human beings in their day-to-day activities that mainly requires the physical understanding of the objects around them. An example of data fusion is the union of the visual senses with touch (the ability to touch) for dexterity and object recognition.

According to [4], sensory fusion is the task of managing and coupling data and information obtained through different types of sensors to improve a specific criterion or decision making in a given process. The fusion technique normally occurs computationally through an algorithm that performs this task. The merged sensory data are enriched, improved, and more reliable than that generated by individual sensors separately.

In the field of sensing and perception in robotics, there are classic approaches of sensor fusion through probabilistic methods such as the Bayesian inference and Kalman Filter.

In [11], it is highlighted that with the growth of the technology of multimodal sensors, that is, sensors that provide data in different dimensions, sizes, and particularities, the application of Bayesian methods began to present difficulties. These methods normally work with small dimensions or data with homogeneous characteristics. To overcome this problem, the field of Machine Learning and Artificial Intelligence has been useful and the target of current research in sensory fusion.

There are currently several modern sensor fusion schemes listed in the literature [4,11]. The three main fusion schemes, represented in Figure 1, are: (i) Early Fusion, (ii) Feature Fusion, and (iii) Late Fusion. In the Early Fusion technique, the data from different sensors are unified in their initial form—in their raw form (raw level). Theoretically, this fusion occurs after the immediate reading of the sensors. Feature Fusion, as its name implies, is a feature extractor (usually a shallow neural network) used after reading the raw sensor data to merge the main properties of that particular data. In Late Fusion, multiple classifiers (usually a deep neural network) are used before data union, which means that this technique is as close as possible to a decision level.

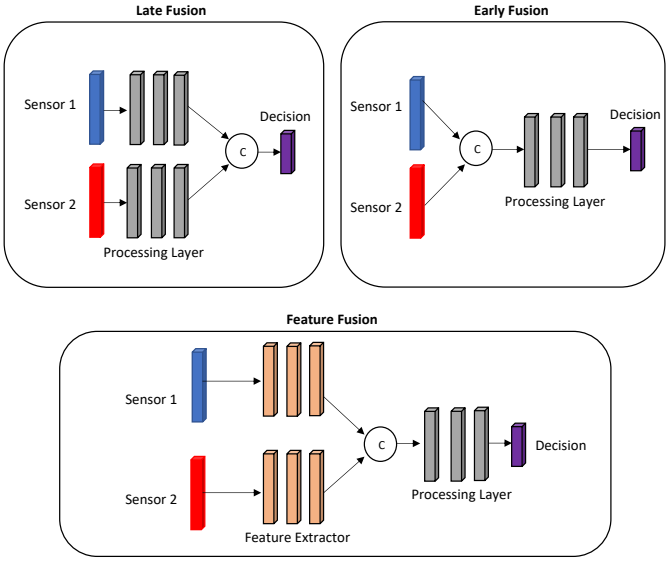

Figure 1. Representation of current sensory fusion methods.

4. Proposal—DDQN with Sensor Fusion

The purpose of this section is to describe the sensory fusion proposals that work in conjunction with the reinforcement learning technique, specifically the DDQN, to autonomously navigate AMRs.

Proposed Fusion Methods

This article uses two sensory fusion techniques combined with the controller based on reinforcement learning, namely: (i) Late Modified Fusion (Late DDQN) and (ii) Interactive Fusion (Interactive DDQN). The sensors used in AMR navigation are: (i) RGB-D camera, (ii) Global Positioning System (GPS), (iii) Inertial Measurement Unit, and (iv) Endoder. All these sensors are merged with the respective techniques mentioned above. Soon after, the result of the fusion is combined in a neural network linked to the DDQN reinforcement learning method. The output of the DDQN network is the actions applied to the actuators (motors) of the AMR. Figure 2 presents the flowchart referring to the proposed method.

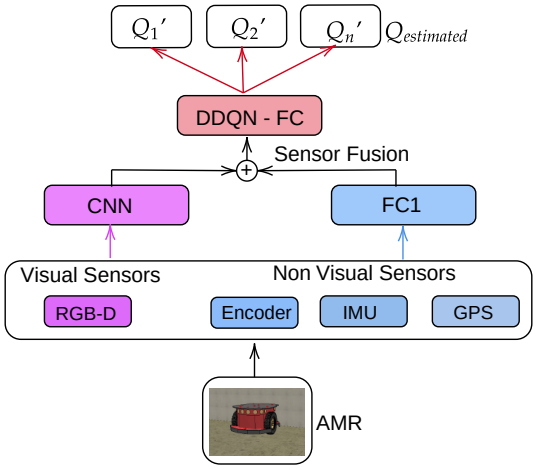

Figure 2. Reinforcement learning-based sensor fusion—Late Fusion DDQN.

The AMR used is a differential robot: the Pioneer 3-DX model. Computer simulations were performed using Coppelia software version 4.3.0 (rev.12). This simulator has a remote API for remote communication to integrate other programming languages, thus allowing the use of the Python language. All algorithms in this article were developed in Python version 3.8.13.

The term Modified Late Fusion is adopted in this paper because the method used in this research has a feature extractor before the application of a Fully Connected neural network, that is, a convolutional neural network (CNN) responsible for processing the images captured by the visual sensor. This way, the method intersects with the Feature Fusion illustrated in Figure 1. The FC1 network is a Fully Connected network responsible for processing the data, referring to non-visual sensors.

Interactive Fusion, as explained in the related works section, is a method proposed by [7] and originally applied in the medical field. In Interactive Fusion, image-related exams are processed by a CNN network, where the authors select for this purpose an architecture known in the literature as VGG-13. On the other hand, exams related to tabular data, that is, data from clinical exams such as blood and categorical data such as sex, among others, are processed by a Fully Connected network. The fusion, or combination of data, occurs specifically in the intermediate convolutional layers of VGG-13, so that the tabular network (Fully Connected) multiplies its output values by the feature maps obtained by CNN. The author calls this technique the Channel Wise Multiplication and claims that iterative convolutional maps are created, similar to attention mechanisms. The technique surpassed, in terms of performance, a Traditional Late Fusion.

In this article, the method initially proposed by [7] is modified in order to carry out the task of sensor fusion in autonomous navigation. The CNN used to extract the characteristics of the image obtained by the visual sensors is shallower than a VGG-13, since the visual image for the perception of the AMR does not have as many characteristics as a clinical MRI exam used by [7]. The same CNN network is adopted for Late Fusion and Interactive Fusion. Table 1 shows the CNN configuration used.

Table 1. CNN structure.

CNN Layer	Filters	Kernel	Stride
1st Layer	16	(5, 5)	5
2nd Layer	32	(3, 3)	2
3rd Layer	32	(2, 2)	2

The images used in our application are captured by an RGB-D camera, processed in grayscale and then subjected to a technique known as "skipframe" for efficient handling and representation. In this process, specific hues in grayscale images represent varying depths, capturing crucial spatial information provided by the RGB-D camera. These depth-integrated grayscale images are then stacked, forming an input of the dimensions $64 \times 64 \times 3$, which is analogous to a standard RGB image but instead encapsulates sequential frames with depth information. This stacking serves multiple purposes: it retains temporal context between consecutive frames, which is vital for understanding the dynamics of the environment, and it aligns the input dimensions with the standard input expected by typical CNN architectures.

Given the nature of our input data, our custom CNN architecture was designed to meet these specific requirements. The architecture comprises three convolutional layers, strategically configured with varying filter sizes and pitches to efficiently process grayscale images with enhanced depth. The first layer, with its larger core (5×5) and width of 5, serves to quickly reduce spatial dimensions while capturing the broader features and depth variations present in stacked images. Subsequent layers with smaller cores and advancements work to refine these features and enhance finer details. This design ensures that the network remains lightweight and computationally efficient, a critical consideration for deployment on the NVIDIA Jetson Nano, which, while powerful, has limited computational resources compared to larger GPU configurations. Reduced network complexity facilitates faster training and real-time inference, crucial for the timely and accurate decision making required in robotic navigation. By adapting our CNN to the specific characteristics of our depth-integrated grayscale images and the computational

constraints of our hardware, we achieve a balance between performance and efficiency, ensuring robust and responsive behavior in real-world robotic applications.

The neural network proposed to learn the $\hat{Q}$ function is composed of 2 Fully Connected layers (DDQN-FC) located after the sensor fusion network, as shown in Figure 2. The two layers have, respectively, 512 ReLU activation function neurons and 3 linear activation function neurons. The evaluation metric (loss function) is the Mean Square Error (MSE) between the calculated Q_{target} (see Equation (1)) and the one estimated by the neural network $\hat{Q}$.

Figure 3 presents the scheme of the Interactive Fusion method adapted to the navigation problem. It is worth noting that in Late Fusion, the characteristics obtained through the CNN are added (concatenated) directly with those obtained by the non-visual sensors. In the Interactive method, the main idea is to apply attention mechanisms to the image that guides the robot's pose, as well as the direction it should follow.

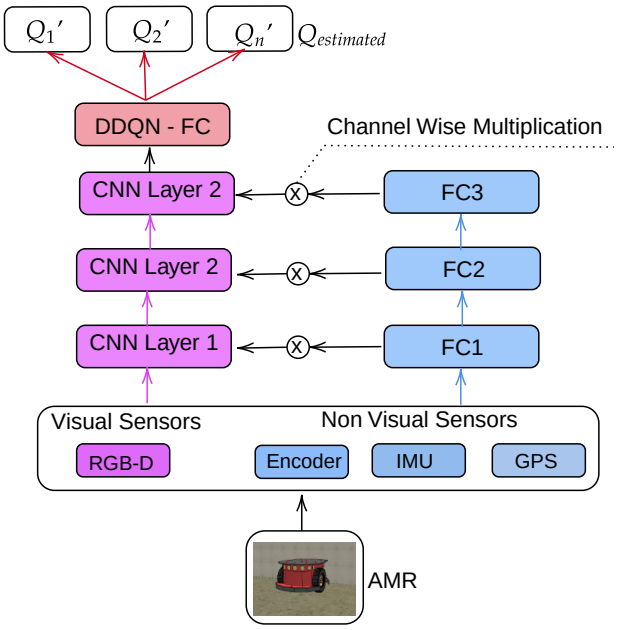

Figure 3. Interactive Fusion DDQN.

5. Security Module

The purpose of using the safety module is to avoid collisions between the mobile robot and people moving in the environment during the navigation process. For this, the security module algorithm is inserted into the autonomous navigation system to support the main controller that uses the sensor fusion algorithm. The essence of how the security module works is based on a supervised learning process, involving the classification of objects using computer vision.

Every time a person approaches the robot, the safety module activates, thus taking place a braking process to avoid a collision. If the subject moves away or out of the camera's field of view, the main sensor's fusion controller reverts to acting as the main control. The safety module has priority to perform braking over commands from the main controller.

The security module uses the ResNet-50 [28] convolutional neural network to detect the shape of humans in the scene. Basically, the ResNet-50 is used as a binary classifier, which verifies the existence of close or distant people (or no people) in the scene. As it is necessary to have a sense of depth in the scene, the RGB-D camera is used as a visual sensor. The ResNet50 network output function is a sigmoid. If the output value of Resnet50 is greater than 0.5, it is assumed that there is a person near the robot. If it is less than this value, the path is considered free, and the sensor fusion algorithm operates the navigation control. The idea is to train the ResNet-50 with a dataset of RGB-D images. We propose to

carry out this training separately from the main controller that uses sensor fusion. Images of real people were combined with synthetic images of people collected in the Coppelia simulator, thus developing a customized dataset.

The considered dataset contains more than 3000 RGB-D frames acquired in a Universität Freiburg's hall from three vertically mounted Kinect sensors. The data mainly consist of images of people walking upright and standing, seen from different orientations and with different levels of occlusions. From these data, images were selected with people relatively close to the sensor and far from the sensor. Figure 4 presents an image with synthetic people collected in the Coppelia simulator. On the other hand, Figure 5 presents an image of real people obtained from the dataset in [29].

Figure 4. Images of synthetic people on the scene.

Figure 5. Images of real people [29].

Figure 6 shows the proposal for inserting the safety module into the sensor fusion controller. It is possible to see that the security module can have priority actions in the control system in case of the detection of people close to the robot.

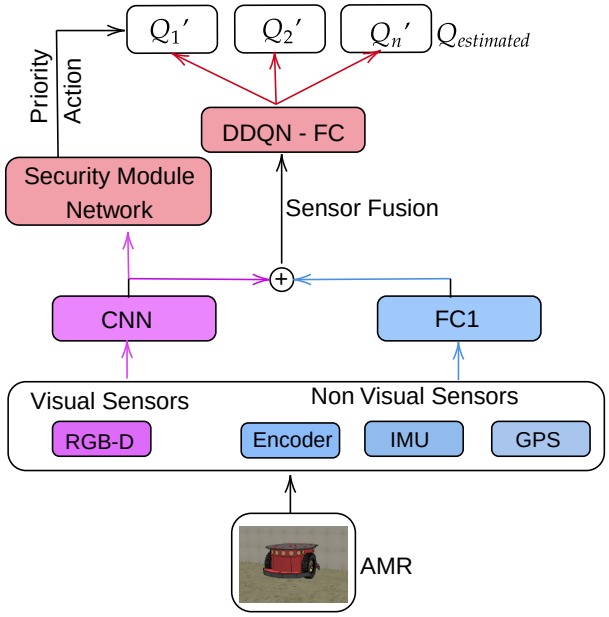

Figure 6. Proposal for the insertion of the security module in the fusion control system.

Proposed Mission

The mission given to AMR is to navigate in an industrial environment autonomously. The AMR has no knowledge of the environment map and does not have a pre-defined trajectory planning. He must learn to navigate via reinforcement learning.

This industrial environment has four workstations, as shown in the figure in Section 6.1. The AMR must pass precisely between the colored circles that indicate the position of the workstation, respectively, in green (station 1), white (station 2), green again (station 3), and black (station 4) in a sequential order.

This mission may be an indication of AMR's ability to perform tasks compatible with the need of industry 4.0, such as collecting process data and autonomously navigating, among others. If the AMR does not follow this indicated sequence, that is, follows the correct trajectory but does not pass between the circles, it suffers a punishment, and the training episode is interrupted. The reward function, which guides the AMR learning process, is given by:

$$R = \begin{cases} dist_{k-1} - dist_k & \text{if active} \\ -1, & \text{if collided} \\ +1, & \text{hit target} \end{cases} \quad (4)$$

$dist_k$ and $dist_{k-1}$ are, respectively, the distances from the robot to the target at the instant k. This calculation allows you to reward actions that increasingly reduce the distance to the target (workstations). If the robot collides with a wall, it suffers a -1 point penalty. The targets vary in relation to the sequence that the robot must navigate, as explained above.

6. Results

In this section, the results regarding the implementation of sensory fusion methods combined with reinforcement learning will be presented. Initially, the results without sensor fusion are displayed and later those with sensor fusion.

6.1. Performance of Learning Methods by Reinforcement without Sensor Fusion

As a justification for the use of the DDQN method, before carrying out its implementation, a simpler alternative was tested, that is, a more elementary reinforcement learning algorithm, such as DQN. Initially, only one perception sensor, the RGB-D camera, was used. Thus, in the initial tests, there was no implementation of sensor fusion. Figure 7 presents the result of implementing the DQN through the Rewards Obtained × Training Episodes graph.

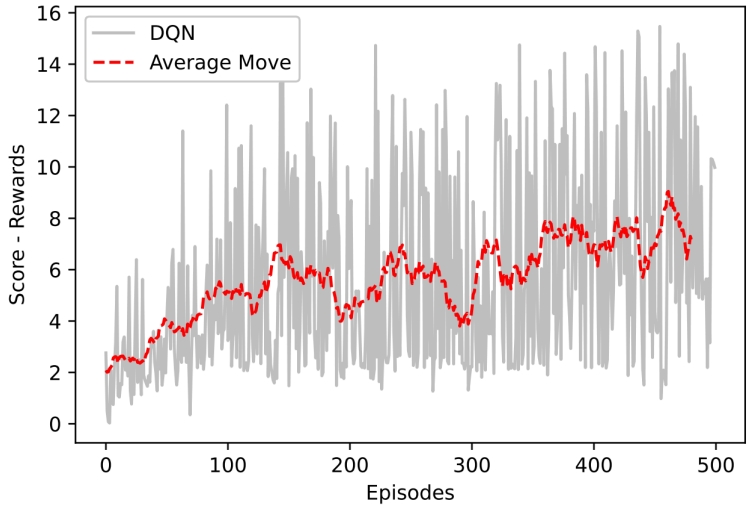

Figure 7. Rewards through the DQN Network.

It is possible to observe that the growth rate of the moving average is relatively low, reaching a maximum average of ≈8. This indicates that with the DQN method and exclusive use of the RGB-D visual sensor, the AMR learns few efficient actions regarding the proposed mission. In addition, it is possible to observe high volitivity in the gross rewards obtained per episode (gray line of the graph), an indication that the stability of the method is weak.

Figure 8 presents the result of implementing the DDQN through the graph of Rewards Obtained × Training Episodes.

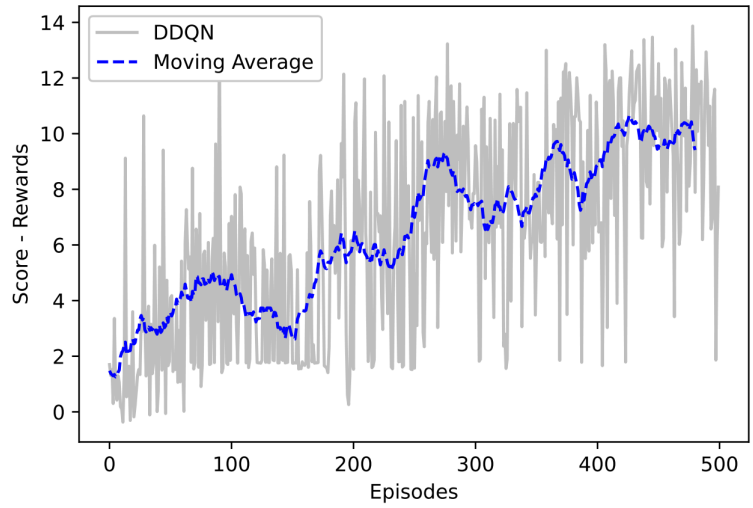

Figure 8. Rewards through the DDQN Network.

As it can be seen in Figures 7 and 8, DDQN tends to provide higher values of rewards than the DQN algorithm. The method also presents greater stability and reached an average

of ≈10. The difference between the computational cost of DQN and DDQN is practically imperceptible. These were indicative for the selection of the DDQN in the work in question.

It is worth mentioning that despite the improvements regarding the DDQN method, the exclusive use of the RGB-D sensor was not enough to complete the proposed mission. Perception by the RGB-D camera results in states sufficient enough to navigate without collision; this is due to the level of scene depth provided by the sensor. However, it is not sufficient enough to accurately represent the points needed to pass through each workstation accurately. Figure 9 shows this method deficiency. It is possible to observe that the trajectory (blue line) of the AMR passes through station 1 and 2; however, when passing through station 3, the AMR deviates from the route.

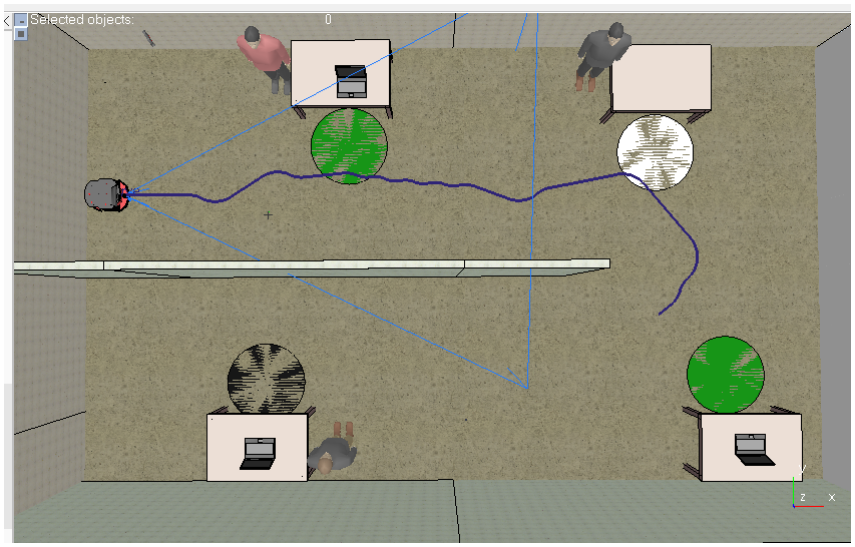

Figure 9. Mission trajectory with the Pure DDQN.

6.2. Application of Sensor Fusion Methods

After verifying that the DDQN method using only the RGB-D camera did not present sufficient results, the proposals for the fusion of visual and non-visual sensors were applied. The methods implemented were DDQN–Late Fusion and DDQN–Iterative Fusion. Figure 10 shows the moving average of the merger methods with the simple DDQN (visual sensor only). It is possible to visually notice the superiority in terms of the moving average in the Late Fusion method. However, it is also necessary to observe metrics that indicate the quality of navigation.

The success rate indicates the number of times the AMR completed the mission completely, i.e., passed through the four workstations accurately. Thus, this rate is calculated as the ratio between the amount of success and the number of training episodes. As the metric is evaluated during the learning process, we have the AMR exploration stage where it is not possible to have a 100% success rate due to the randomness of the actions applied in the exploration phase. Normally, this rate evolves during the exploitation phase, and therefore, the values presented are relatively low. The intention is to emphasize that with the proposed method, the success rate increased significantly during learning. Table 2 provides a summary of the navigation metrics during learning.

Table 2. Metrics measured in the learning process.

Proposal	Success Rate (%)	Global Reward Average
Late Fusion DDQN	28.2	7.38
Interactive DDQN	11.6	6.43
Pure DDQN	1.4	5.55

The DDQN–Late Fusion approach exhibited superior performance compared to the Interactive DDQN method. The learning curve for all proposed methods, illustrated in Figure 10, clearly demonstrates this advantage. Moreover, Figure 11 delineates the moving average over the concluding episodes of training, further highlighting the preeminence of the DDQN–Late Fusion technique.

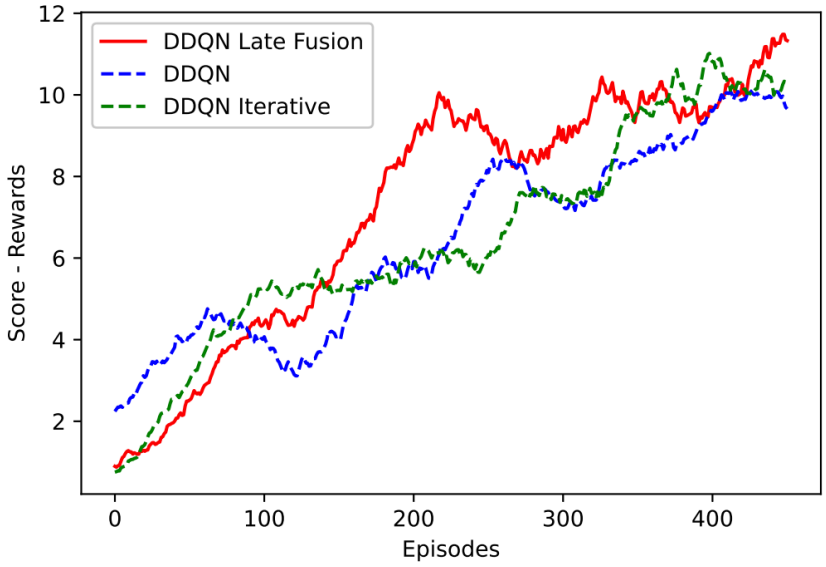

Figure 10. Comparison between the navigation methods with and without fusion.

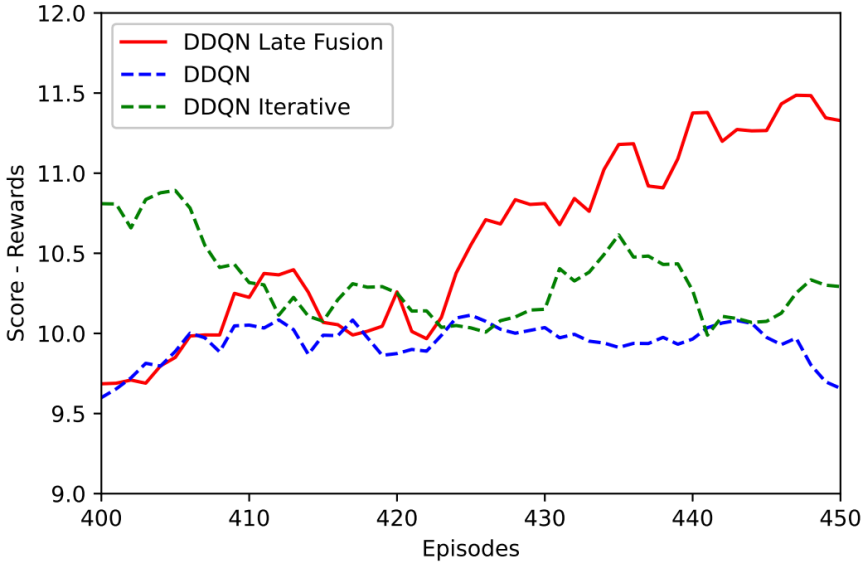

Figure 11. Emphasis on the final epochs of training.

Upon closer inspection of the learning curves, it is evident that between episodes 150 and 250, the DDQN–Late Fusion method consistently achieved a higher average reward, hovering around 10 points, in stark contrast to the other methods, which averaged around 6 points. While the rewards for both approaches plateaued near the 10-point mark from episodes 250 to 400, a detailed analysis of the span from episodes 400 to 450 reveals a marginal yet noteworthy advantage for the DDQN–Late Fusion.

It is possible to observe that for the Pure DDQN, the success rate is extremely low when compared to fusion methods. Looking at the average of all rewards obtained (Global Reward Average), the difference is not that big. As previously mentioned, the Pure DDQN starts to fail from station 3.

It was observed that the integration of non-visual sensors is essential to increase the success rate. Figure 12 illustrates the trajectory of the DDQN–Late Fusion method.

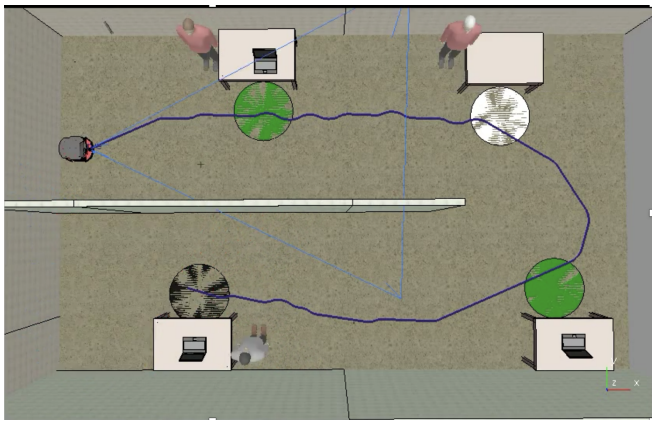

Figure 12. Mission trajectory with the Late Fusion–DDQN method.

To complement the findings presented in this paper, a video has been prepared showcasing the navigation of an Autonomous Mobile Robot (AMR) utilizing the DDQN–Late Fusion method. This video serves as supplementary material and offers a visual representation of the method's efficacy in real-world navigation scenarios. The link to this additional content is provided in Supplementary Materials.

6.3. Fusion Performance with Security Module

The training of the ResNet50 network (security module backbone) is based on a fine-tuning process. That is, the weights trained in ImageNet [28] were used and tuned along the training epochs in all layers of the network. After this process, the final ResNet50 configuration was applied as a binary output person detector. The ResNet50 network was trained for 500 epochs using the cross-binary entropy cost function. The network accuracy to detect people in the scenery in the test set was 93.96%.

Figure 13 (Percentage Accuracy × Epochs) and Figure 14 (Loss value × Epochs) show the results referring to training via supervised learning of the security module: the result referring to the evolution of the accuracy curve in the training set and the reduction of the cost function, respectively.

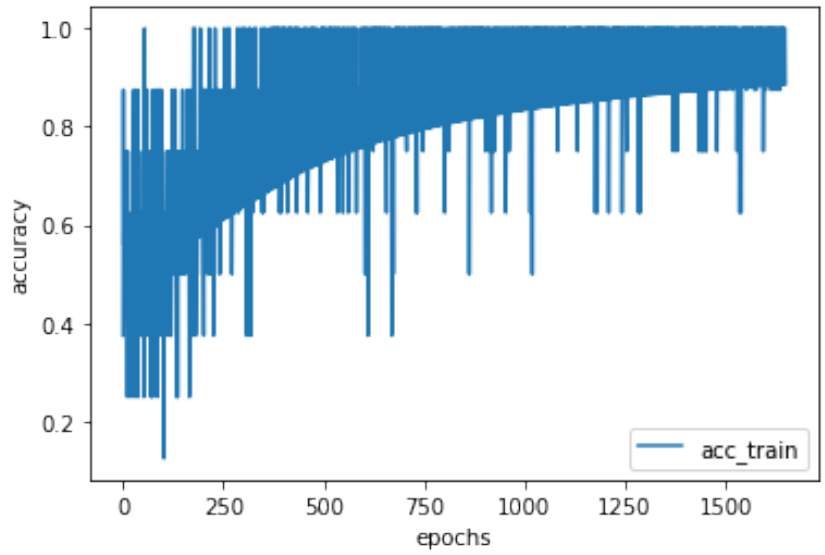

Figure 13. Evolution of the accuracy in the training set.

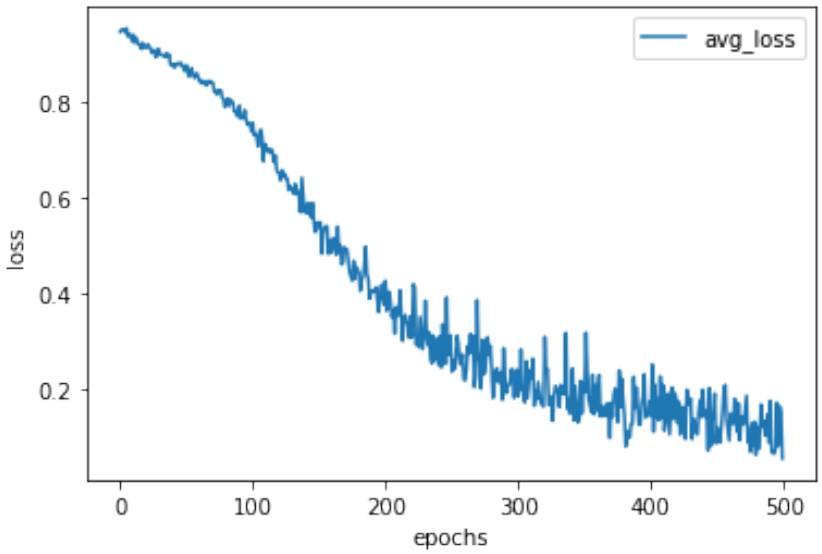

Figure 14. Minimization of the cost function in training.

The accuracy in Figure 13 is presented in terms of proportion, where the value 1 represents a perfect 100% accuracy. The loss value in Figure 14 represents the direct value calculated by the dimensionless cross-entropy function.

After completing the supervised learning training, the safe detection algorithm was inserted into the mobile robot and started to work together with the sensor fusion method in a separate module, or complementary. Table 3 presents the results referring to the tests developed in the simulation environment. The tests were performed with the robotic agent exclusively navigating with the fusion algorithm and with the help of the security module. To increase the complexity of the environment, people who move in the scene were also inserted.

Table 3. Metrics measured in the deployment process in a dynamic environment.

Test Number	Late Fusion (LF)		LF + Security Module	
First	Reward Average	6.22	Reward Average	8.83
	Success Rate	40%	Success Rate	40%
Second	Reward Average	7.61	Reward Average	9.08
	Success Rate	40%	Success Rate	60%
Third	Reward Average	5.72	Reward Average	10.19
	Success Rate	20%	Success Rate	40%
Final Average	6.51		9.37	
Success Rate on Tests	33.3%		46.7%	

Three sets of tests were included with the robot in the dynamic environment, where three people were inserted who moved randomly. The average success rate in the proposed mission was 46.7%. Without the security module, the success rate was 33.3%. It is evident that the security module avoided the robot's collision at certain times and increased the average of final rewards: 6.51% without the security module and 9.37% with the security module. Collisions continued to occur because people in the scene could collide with the side or rear of the robot, out of its field of vision, due to the randomness and unconsciousness of their movements.

6.4. Practical Implementation

To evaluate the practical applicability of our proposed method in real-world environments, we conducted experiments using a robotic platform equipped with an NVIDIA Jetson Nano and an RGB camera. This robot is illustrated in Figure 15.

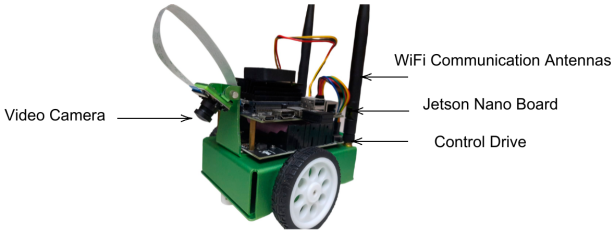

Figure 15. Mobile robot with a Jetson Nano development board.

The main objective of these experiments was to test the performance of the proposed CNN and reward function in real-world conditions, providing a practical validation of our navigation and perception method. To achieve this, the robot was assigned a mission: navigate a track using only the visual part (DDQN + CNN) with the proposed reward function. The robot will not be able to collide with any obstacle outside the track or even leave the track's limitations. This test environment (track) is illustrated in Figure 16. The tests also validate the methodology for transferring weights from neural networks trained from the Coppelia VREP Simulator to a real programmable robot.

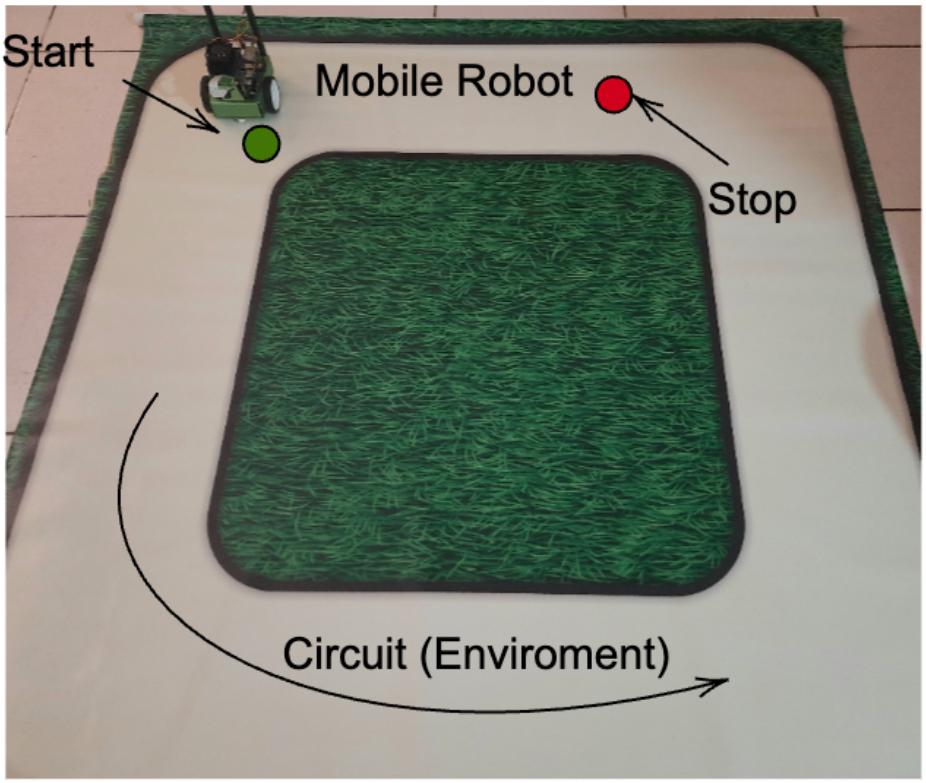

Figure 16. Mission assigned to the robot: starting point, stopping, and proposed circuit.

The hardware used for testing consists of an NVIDIA Jetson Nano, a powerful and efficient computing module capable of running multiple machine learning models in parallel and processing sensor data simultaneously. The Jetson Nano was chosen due to its ability to provide sufficient computational power for the real-time execution of convolutional neural networks (CNNs), making it an ideal choice for autonomous robotic applications.

One of the key steps in this experiment was the transfer of the CNN weights trained in a simulation environment to the real hardware. This process, known as Sim-to-Real [30], is crucial in validating whether models trained in simulations are capable of performing adequately in real-world settings. The transfer was successfully completed, allowing the proposed CNN to be tested directly on the robotic platform.

The CNN was configured in a real environment and is presented in Table 1. It is the same one used in a simulation environment. The reward function is given by Equation (4) and is the same one used in our simulations. The visual part of the navigation system as well as the navigation control algorithm are evaluated in the real implementation. Figure 6 shows the visual sensors (displayed in pink) as well as the DDQN-FC module in a real environment. Even being a simple environment, we believe that we can show that our methodology is applicable in a real environment, that is, the mobile robot learning carried out in Coppelia VEP simulations is effectively transferable to a real environment.

The reward function, on the other hand, was designed to encourage safe and efficient behaviors, penalizing collisions and rewarding progress toward the goal. The practical implementation of these components allowed us to evaluate their performance in real conditions, providing valuable insights for future improvements and adjustments.

The experiments conducted showed that the proposed CNN and reward function are capable of autonomously guiding the robot in different scenarios, avoiding obstacles, and seeking to reach the defined goal. Figure 17 illustrates one of the trajectories taken by the robot, highlighting the decision points and the actions taken in response to environmental perceptions. A video of the test performed can be found in the Supplementary Materials.

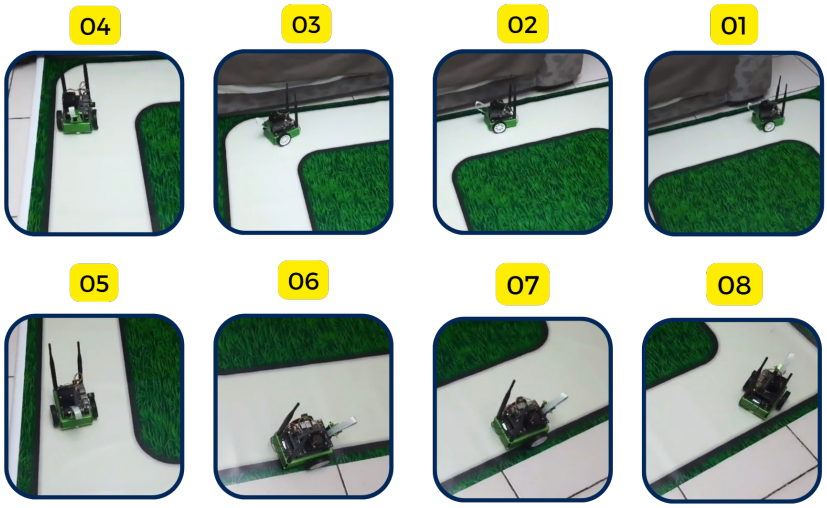

Figure 17. Robot movement frames.

Despite the promising results, it is important to highlight that the complete implementation of the sensor fusion system, including non-visual sensors, is a crucial step to increase the robustness and reliability of the system in more complex and dynamic environments. This step will be explored in future works, once the necessary sensors are available and integrated into the platform.

7. Conclusions

When evaluating certain missions involving autonomous navigation that require certain precision, it is concluded that it is necessary to merge sensory information to increase the success rate and, consequently, the AMR performance. Sensor fusion is still an incipient area in systems guided via reinforcement learning. With the evolution of Deep Learning, new fusion possibilities are emerging. Thus, the DDQN–Late Fusion methods and an adaptation of the Interactive Fusion method, called DDQN–Interactive Fusion, were proposed in this article.

It was observed that the DDQN–Late Fusion presented the best performance when compared with the other methods. Despite the authors in [7] stating in their research the superiority of the Iterative Fusion method in classification tasks, in the activity of joining sensors and applying it in reinforcement learning, Late Fusion was superior. This reinforces the thesis that each fusion method must be investigated in its respective applications. Late Fusion had a success rate of ≈28%, which was higher than Pure DDQN and Iterative DDQN, which had a success rate of ≈1% and ≈11%, respectively. It is worth noting that this success rate was calculated during the training process, and for this reason, its values are respectively low since the success rate only increases in the agent's exploitation stage.

Another point noted is the effectiveness of the proposed anti-collision safety module, integrated under the control architecture. This module is useful when you have dynamics in the environment, for example, people moving around. The security module has increased the success rate of completing the mission and the advanced rewards through the reinforcement method.

The fusion methods proposed in this article are part of a larger control architecture for autonomous navigation that is under development. This architecture also features local control and wireless communication capabilities. Future works intend to implement simultaneous mapping and localization (SLAM) techniques and also integrate them into the system. Developing other methods of sensor fusion is also intended.

The scope of this work has laid the groundwork for a comprehensive sensor fusion architecture tailored for autonomous robotic systems. Due to hardware limitations in our current laboratory setup, we have implemented in a real environment and validated only a segment of the proposed architecture in a practical hardware scenario. The full realization of the sensor fusion system on hardware, a critical step towards deploying this technology in real-world applications, is earmarked for future studies. These subsequent investigations will enable us to explore the complete potential of the architecture, addressing the challenges and harnessing the capabilities that were beyond the reach of the present work due to the constraints of the available resources. This planned extension will not only fortify the theoretical findings of this paper, but will also provide a robust platform for benchmarking against real-world complexities and performance criteria.

Supplementary Materials: This section presents some supplementary videos that illustrate the tests carried out in simulation and practical environments. (1) Navigation through the DDQN-Late Fusion Algorithm in a Simulation Environment: https://www.loom.com/share/684afa6a5b0148afadc9a200ab9f3483 (accessed on 6 November 2023); (2) Practical Implementation, Validation of the Proposed CNN Network and Reward Function: https://bit.ly/3IKlRb5 (accessed on 6 November 2023).

Author Contributions: The individual contributions of the authors to the manuscript are as follows: C.D.d.S.B. was involved in the conceptualization, methodology, software development, validation, formal analysis, and investigation. D.P.Q.C. contributed to the review and editing of the manuscript, as well as the visualization of data and results. F.H.T.V. contributed to the review and editing, visualization, supervision of the research project, project administration, and was responsible for the acquisition of funding. He also contributed to the methodology of the study. All authors have read and agreed to the published version of the manuscript.

Funding: We would like to thank the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for the financial support provided for the publication of this article.

Data Availability Statement: Restrictions apply to the availability of these data. Data was obtained from http://www2.informatik.uni-freiburg.de/~spinello/RGBD-dataset.html (accessed on 6 November 2023) and are available http://www2.informatik.uni-freiburg.de/~spinello/RGBD-dataset.html (accessed on 6 November 2023) with the permission of Luciano Spinello. Publicly available datasets were analyzed in this study. This data can be found here: https://drive.google.com/drive/folders/1vsT0PS8a_PlaXbDIDFN5rlejPbUt2c4Q?usp=sharing (accessed on 6 November 2023).

Conflicts of Interest: The authors have no relevant financial or non-financial interests to disclose. The authors have no competing interests to declare that are relevant to the content of this article. All authors certify that they have no affiliations with or involvement in any organization or entity

with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors have no financial or proprietary interests in any material discussed in this article.

References

1. Andreja, R. Industry 4.0 Concept: Background and Overview. *Int. J. Interact. Mob. Technol. (IJIM)* **2017**, *11*, 77–90. [CrossRef]
2. Mohd, J.; Abid, H.; Ravi, P.; Rajiv, S. Substantial Capabilities of Robotics in Enhancing Industry 4.0 Implementation. *Cognitive Robot.* **2021**, *1*, 58–75. [CrossRef]
3. Dudek, G.; Jenkin, M. *Computational Principles of Mobile Robotics*, 2nd ed.; Cambridge University Press: New York, NY, USA, 2010.
4. Fayyad, J.; Jaradat, M.A.; Dominique, G.; Homayoun, N. Deep Learning Sensor Fusion for Autonomous Vehicle Perception and Localization: A Review. *Sensors* **2020**, *20*, 4220. [CrossRef] [PubMed]
5. Krohn, J.; Beyleveld, G.; Bassens, A. *Deep Learning Illustrated: A Visual, Interactive Guide to Artificial Intelligence*; The Addison-Wesley Data & Analytics Series; Addison Wesley: Boston, MA, USA, 2019.
6. Geron, A. (Ed.) *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*; O'Reilly Media: Sebastopol, CA, USA, 2019.
7. Duanmu, H.; Huang, P.B.; Brahmavar, S.; Lin, S.; Ren, T.; Kong, J.; Wang, F.; Duong, T.Q. Prediction of Pathological Complete Response to Neoadjuvant Chemotherapy in Breast Cancer Using Deep Learning with Integrative Imaging, Molecular and Demographic Data. In Proceedings of the Medical Image Computing and Computer Assisted Intervention—MICCAI 2020: 23rd International Conference, Lima, Peru, 4–8 October 2020; Part II; Springer: Berlin/Heidelberg, Germany, 2020; pp. 242–252. [CrossRef]
8. Feng, D.; Haase-Schütz, C.; Rosenbaum, L.; Hertlein, H.; Gläser, C.; Timm, F.; Wiesbeck, W.; Dietmayer, K. Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 1341–1360. [CrossRef]
9. Brenner, M.; Reyes, N.H.; Susnjak, T.; Barczak, A.L.C. RGB-D and Thermal Sensor Fusion: A Systematic Literature Review. *IEEE Access* **2023**, *11*, 82410–82442. [CrossRef]
10. Yang, M.Y.; Rosenhahn, B.; Murino, V. *Multimodal Scene Understanding: Algorithms, Applications and Deep Learning*; Elsevier Science: Amsterdam, The Netherlands, 2019; ISBN 9780128173589.
11. Bednarek, M.; Kicki, P.; Krzysztof, W. Robustness of Multi-Modal Fusion—Robotics Perspective. *Electronics* **2020**, *9*, 1152. [CrossRef]
12. Patle, B.K.; Pandey, A.; Parhi, D.R.K.; Jagadeesh, A.J.D.T. A review: On path planning strategies for navigation of mobile robot. *Def. Technol.* **2019**, *15*, 582–606. [CrossRef]
13. Wang, J.; Chi, W.; Li, C.; Wang, C.; Meng, M.Q.H. Neural RRT*: Learning-based optimal path planning. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1748–1758. [CrossRef]
14. Ait Saadi, A.; Soukane, A.; Meraihi, Y.; Benmessaoud Gabis, A.; Mirjalili, S.; Ramdane-Cherif, A. UAV path planning using optimization approaches: A survey. *Arch. Comput. Methods Eng.* **2022**, *29*, 4233–4284. [CrossRef]
15. Guo, J.; Liu, Q.; Chen, E. A Deep Reinforcement Learning Method for Multimodal Data Fusion in Action Recognition. *IEEE Signal Process. Lett.* **2022**, *29*, 120–124. [CrossRef]
16. Sun, Y.; Zhang, L.; Ma, O. Force-Vision Sensor Fusion Improves Learning-Based Approach for Self-Closing Door Pulling. *IEEE Access* **2021**, *9*, 137188–137197. [CrossRef]
17. Karle, P.; Fent, F.; Huch, S.; Sauerbeck, F.; Lienkamp, M. Multi-Modal Sensor Fusion and Object Tracking for Autonomous Racing. *IEEE Trans. Intell. Veh.* **2023**, *8*, 3871–3883. [CrossRef]
18. Chi, W.; Ding, Z.; Wang, J.; Chen, G.; Sun, L. A Generalized Voronoi Diagram-Based Efficient Heuristic Path Planning Method for RRTs in Mobile Robots. *IEEE Trans. Ind. Electron.* **2022**, *69*, 4926–4937. [CrossRef]
19. Yan, C.; Xiang, X.; Wang, C. Towards Real-Time Path Planning through Deep Reinforcement Learning for a UAV in Dynamic Environments. *J. Intell. Robot. Syst.* **2020**, *98*, 297–309. [CrossRef]
20. Puente-Castro, A.; Rivero, D.; Pazos, A.; Fernandez-Blanco, E. A review of artificial intelligence applied to path planning in UAV swarms. *Neural Comput. Appl.* **2022**, *34*, 153–170. [CrossRef]
21. Jiang, S.; Yao, W.; Wong, M.S.; Hang, M.; Hong, Z.; Kim, E.J.; Joo, S.H.; Kuc, T.Y. Automatic Elevator Button Localization Using a Combined Detecting and Tracking Framework for Multi-Story Navigation. *IEEE Access* **2020**, *8*, 1118–1134. [CrossRef]
22. Pehlivanoglu, Y. Volkan and Pehlivanoglu, Perihan. An Enhanced Genetic Algorithm for Path Planning of Autonomous UAV in Target Coverage Problems. *Appl. Soft Comput.* **2021**, *112*, 107796. [CrossRef]
23. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998. (Also translated into Japanese and Russian)
24. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjel, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
25. Winder, P. *Reinforcement Learning*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2020; ISBN 9781098114831.
26. van Hasselt, H.; Guez, A.; Silver, D. Deep Reinforcement Learning with Double Q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16; AAAI Press: Phoenix, Arizona, 2016; pp. 2094–2100. [CrossRef]

27. Wang, Z.; Schaul, T.; Hessel, M.; van Hasselt, H.; Lanctot, M.; de Freitas, N. Dueling network architectures for deep reinforcement learning. In Proceedings of the 33rd International Conference on Machine Learning (ICML), New York, NY, USA, 19–24 June 2016; Volume 48, pp. 1995–2003.
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**. [CrossRef]
29. Luber, M.; Spinello, L.; Arras, K. People tracking in RGB-D Data with on-line boosted target models. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 3844–3849. [CrossRef]
30. Zhao, W.; Queralta, J.P.; Westerlund, T. Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: A Survey. In Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, ACT, Australia, 1–4 December 2020; pp. 737–744. [CrossRef]