

Article

Not peer-reviewed version

---

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

---

[Carlos Daniel Bezerra](#)\*, Daniel Porto Carneiro, [Flávio Henrique Vieira](#)

Posted Date: 23 October 2023

doi: 10.20944/preprints202310.1391.v1

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



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

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

Carlos D. S. Bezerra <sup>1,\*</sup>, Flávio H. T. Vieira <sup>1,2</sup> and Daniel P. Q. Carneiro <sup>2</sup>

<sup>1</sup> Instituto de Informatica (INF), Federal University of Goiás (UFG), Goiás, Brazil

<sup>2</sup> Escola de Engenharia Eletrica, Mecânica e de Computação (EMC), Federal University of Goiás (UFG), Goiás, Brazil

\* Correspondence: carlosdbez@discente.ufg.br

**Abstract:** In this work, we propose an approach for the autonomous navigation of mobile robots using fusion 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  $\approx 27\%$  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 14% the success rate when compared to that of the autonomous navigation approach without the security module. Video was developed with robot navigation using the DDQN-Late Fusion method: <https://www.loom.com/share/684afa6a5b0148afadc9a200ab9f3483>.

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

## 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 the Industry 4.0 is the use of pillar technology for real-time monitoring and reduction of human intervention in production, consequently, obtaining greater precision in certain tasks and safety for the 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 [11]. 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 arise in function of three major areas of mobile robotics: perception, planning and action [4]. 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, joints, among others.

According to [5], 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 (*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 (*Reinforcement Learning - RL*).

RL is a powerful machine learning tool for AMR navigation. According to [8], 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 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 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 navigation of AMRs. Autonomous navigation approach is proposed that uses the *Deep Reinforcement Learning* algorithm combined with the following sensory fusion strategies: **Interactive-Model** [3] and **Late Fusion**. Sensory fusion of images is performed, 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 classification of objects in the scene, identification of routes, 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 originates in the medical area for classifying diseases.

The article is organized as follows: section 2 presents related works, 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 [5].

The authors in a [5] present a comprehensive literature review on 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. [2] 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, patient's gender and sex. The method described in [3] 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.

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. 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  $\pi$  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:

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

where  $r$  is the immediate obtained reward and  $\gamma$  a discount factor for future rewards,  $s$ ,  $s'$ ,  $a$ , and  $a'$  are the states and actions 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 ( $\theta$ ) 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 is dynamic and updated at each step of the algorithm's integration.

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

The DDQN algorithm, proposed by [12], 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 get the maximum value of  $Q(s', a')$  means getting 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 Double DQN we have two independent neural networks: the  $Q^{pi}$  and  $Q_{target}$  network. The modified TD Equation is described below:

$$Q^\pi(s, a) = r + \gamma (Q_{target}(s', \operatorname{argmax}_{a'} Q^\pi(s', a'))) \quad (3)$$

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  $Q^\pi(s', a')$ .

### 3.2. Deep Sensor Fusion

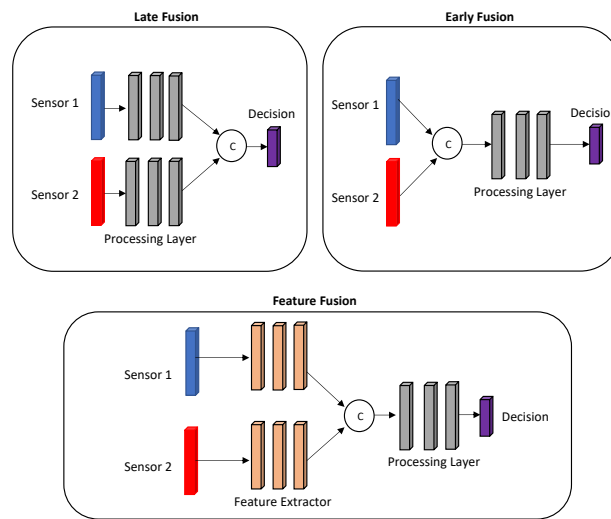
Data Fusion, or sensory information, is a skill constantly used by human beings in their day-to-day activities that mainly require the physical understanding of the objects around them. An example of data fusion is the union of the visual senses with touch (the ability to touch) for dexterity and object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

According to [5] 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 is 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 Bayesian inference and Kalman Filter.

In [2], 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 [2,5]. 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.



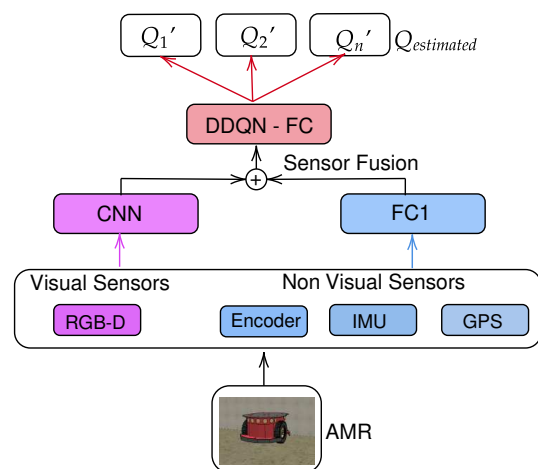
**Figure 1.** Representation of Current Sensory Fusion Methods.

#### 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.

##### 4.1. 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 System (Interactive DDQN). The sensors used in AMR navigation are: i) RGB-D Camera, ii) Global Positioning System (GPS), iii) Inertial Measurement Unit, 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, Pioneer 3-DX model. Computer simulations were performed using Coppelia software. 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.

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 [3] 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, 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 as 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 [3] 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 [3]. 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 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 guide the robot's pose, as well as the direction it should follow.

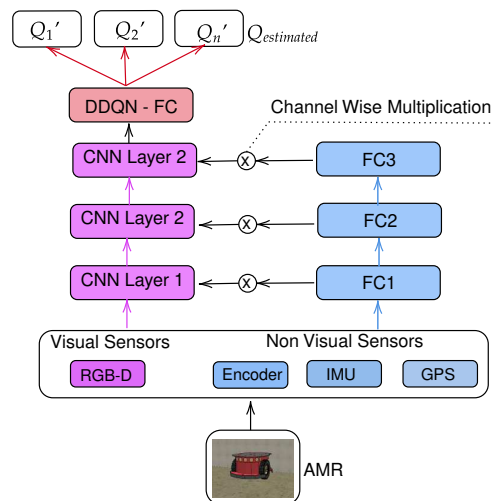


Figure 3. Interactive Fusion DDQN.

## 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 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 [7] 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 dataset customized.

The considered dataset contains more than 3,000 RGB-D frames acquired in a Universität Freiburg's hall from three vertically mounted Kinect sensors. The data mainly consists 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 [9].

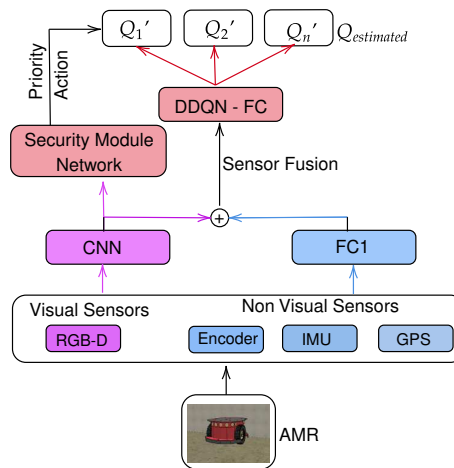


Figure 4. Images of Synthetic People on the Scene.



Figure 5. Images of Real People [9].

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 detection of people close to the robot.



**Figure 6.** Proposal for Insertion of the Security Module in the Fusion Control System.

### 5.1. 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 Figure 9. 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.

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, 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:

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

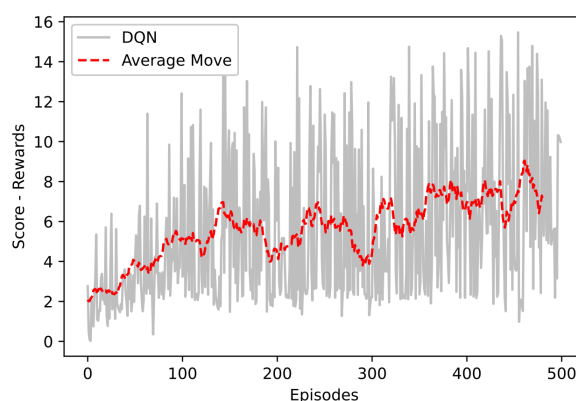
$dist_k$  and  $dist_{k-1}$  are respectively the distances from the robot to the target at the instant  $k$ . This calculation allows you to reward actions that increasingly reduce the distance to the target (workstations). If the robot collides with a wall, it suffers a -1 point penalty. The targets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

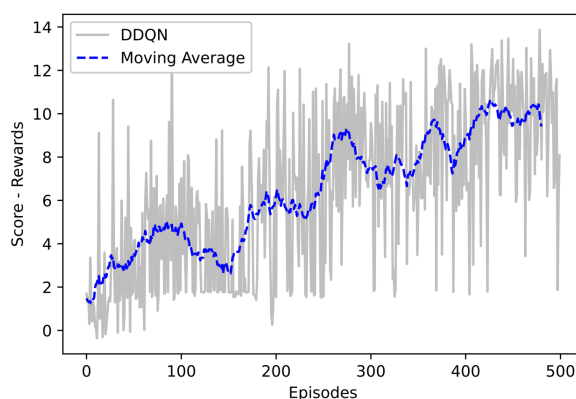
As a justification for the use of the DDQN method, before carrying out its implementation, a simpler alternative was tested, that is, a more elementary reinforcement learning algorithm, such as DQN. Initially, only one perception sensor, the RGB-D camera, was used. Thus, in the initial tests there was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



**Figure 7.** Rewards on DQN Network.

It is possible to observe that the growth rate of the moving average is relatively low, reaching a maximum average of  $\approx 8$ . 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 volatility in the gross rewards obtained per episode (gray line of the graph), an indication that the stability of the method is weak.

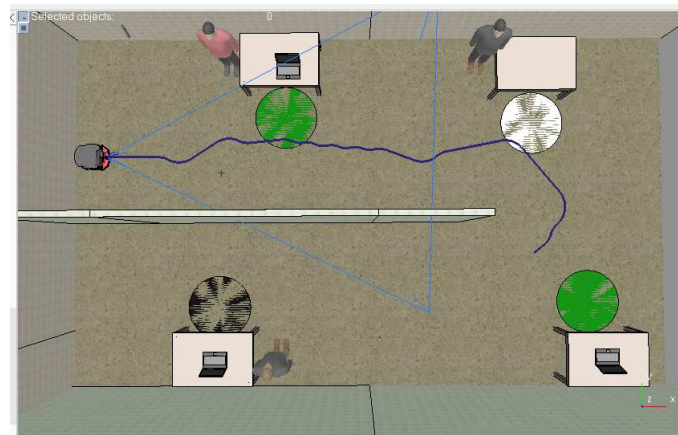
Figure 8 presents the result of implementing the DDQN through the graph of Rewards Obtained  $\times$  Training Episodes.



**Figure 8.** Rewards on DDQN Network.

As it can be seen by Figure 7 and Figure 8, DDQN tends to provide higher values of rewards than the DQN algorithm. The method also presents greater stability and reached an average of  $\approx 10$ . 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 to navigate without collision, this is due to the level of scene depth provided by the sensor, however it is not sufficient 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 Pure DDQN.

### 6.2. Application of Sensor Fusion Methods

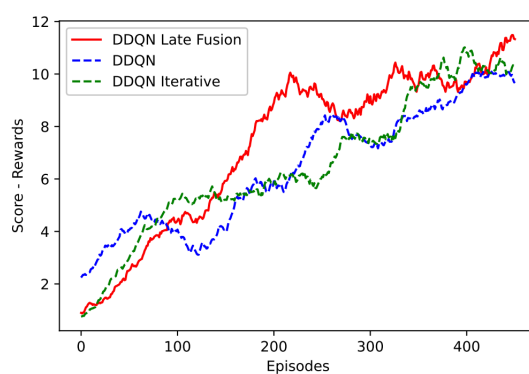
After verifying that the DDQN method using only the RGB-D camera did not present sufficient results, then the proposals for fusion of visual and non-visual sensors are applied. The methods implemented were DDQN-Late Fusion and DDQN-Iterative Fusion. Figure 10 shows the moving average of the merger methods 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 navigation metrics during learning.

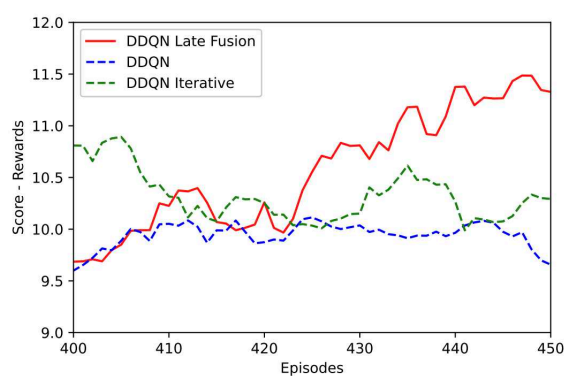
**Table 2.** Metrics measured in the Learning Process.

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

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.

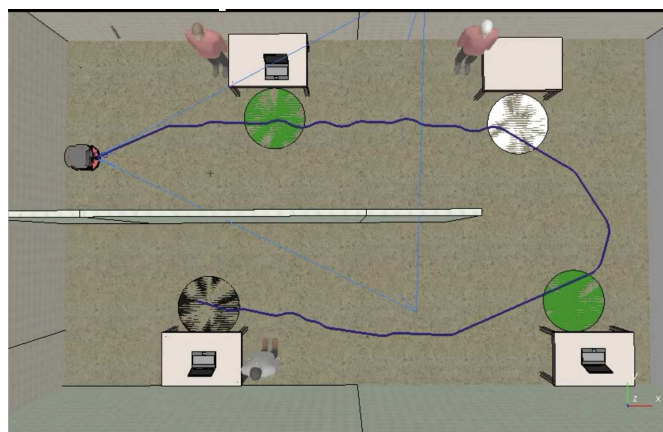


**Figure 10.** Comparison Between Navigation Methods with and Without Fusion.



**Figure 11.** Emphasis on the Final Epochs of Training.

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.



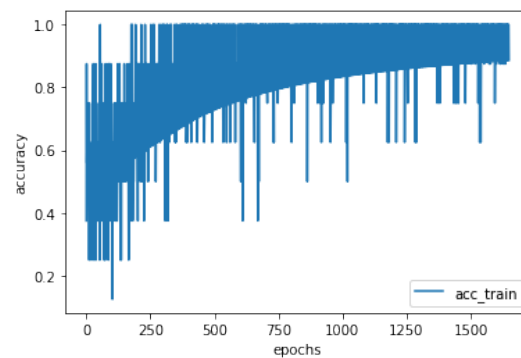
**Figure 12.** Mission Trajectory with Late Fusion DDQN.

The DDQN-Late Fusion method also outperformed the Interactive DDQN method. Figure 10 presents the learning curve of all the proposed methods. Figure 11 emphasizes the moving average in the final training episodes. It is possible to verify the superiority of DDQN-Late Fusion. Video was developed with AMR navigation using the DDQN-Late Fusion method: <https://www.loom.com/share/684afa6a5b0148afadc9a200ab9f3483>

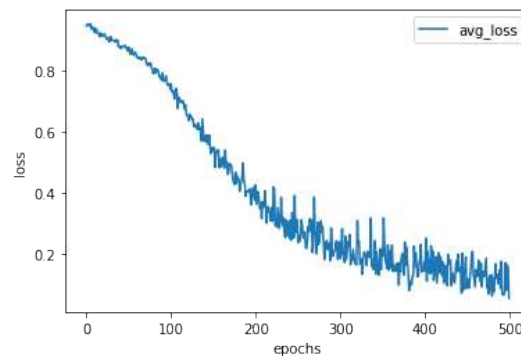
### 6.3. Fusion Performance with Security Module

The training of the ResNet50 network (security module backbone) is based on a fine-tuning process. That is, the weights trained in ImageNet [7] were used and tuned along the training epochs in all layers of the network. After this process, the final ResNet50 configuration was applied as 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%.

Figures 13 and 14 show the results referring to training via supervised learning of the security module, respectively 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 Accuracy in the Training Set.



**Figure 14.** Minimization of the Cost Function in Training.

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	
1st	Reward Average	6.22	Reward Average	8.83
	Success Rate	40%	Success Rate	40%
2nd	Reward Average	7.61	Reward Average	9.08
	Success Rate	40%	Success Rate	60%
3rd	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 move 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 security module and 9.37% with security module. Collisions persisted in occurring 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.

## 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, 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 to the other methods. Despite the authors in [3] 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  $\approx 28\%$ , higher than Pure DDQN and Iterative DDQN, which had a success rate of  $\approx 1\%$  and  $\approx 11\%$  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 by 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. It is also intended to develop other methods of sensor fusion.

**Funding:** The authors did not receive financial support for the preparation of the work.

**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. R. Andreja, *Industry 4.0 concept: Background and overview*, *International Journal of Interactive Mobile Technologies (ijIM)* **11** (2017).
2. M. Bednarek, P. Kicki, and W. Krzysztof, *Robustness of multi-modal fusion—robotics perspective*, *Electronics* **7** (2020), 120–124.
3. H. Duanmu et al., *Prediction of pathological complete response to neoadjuvant chemotherapy in breast cancer using deep learning with integrative imaging, molecular and demographic data*, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part II*, Springer-Verlag, 242–252, doi:10.1007/978-3-030-59713-9\_24.
4. G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics*, 2nd edn., Cambridge University Press, USA, 2010.
5. J. Fayyad et al., *Deep learning sensor fusion for autonomous vehicle perception and localization: A review*, *Sensors* **20** (2020).
6. A. Geron, (ed.) *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly Media, USA, 2019.
7. K. He et al., *Deep residual learning for image recognition*, 2015, doi:10.48550/ARXIV.1512.03385. URL <https://arxiv.org/abs/1512.03385>.
8. J. Krohn, G. Beyleveld, and A. Bassens, *Deep Learning Illustrated: A Visual, Interactive Guide to Artificial Intelligence*, The Addison-Wesley data & analytics series, Addison Wesley, USA, 2019.
9. M. Luber, L. Spinello, and K. Arras, *People tracking in rgb-d data with on-line boosted target models*, 3844–3849, doi:10.1109/IROS.2011.6048836.
10. Y. Michael, R. Bodo, and M. Vittorio, (eds.) *Multimodal Scene Understanding*, O'Reilly Media, USA, 2019.
11. J. Mohd et al., *Substantial capabilities of robotics in enhancing industry 4.0 implementation*, *Cognitive Robotics* **1** (2021).
12. H. van Hasselt, A. Guez, and D. Silver, *Deep reinforcement learning with double q-learning*, *Proceedings of the AAAI Conference on Artificial Intelligence* **30** (2016), no. 1.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.