Scientific Research Publishing

Advances in Artificial Intelligence and Robotics Research, 2026, 2(1), 17-41
https://www.scirp.org/journal/airr

Developing Intelligent Algorithm for Enhancing Robotic Factors Affecting Task Performance Strategical Methods

Zhenli Lu, Anas A. Nicola*

[1]School of Electrical Engineering and Automation, Suzhou University of Technology, Suzhou, China
[2]Faculty of Telecommunication, Engineering and Space Technology, Future University, Khartoum, Republic of the Sudan
Email: zhenlilu@szut.edu.cn, *anasnicola2018@gmail.com

How to cite this paper: Lu, Z.L. and Nicola, A.A. (2026) Developing Intelligent Algorithm for Enhancing Robotic Factors Affecting Task Performance Strategical Methods. *Advances in Artificial Intelligence and Robotics Research*, **2**, 17-41.

Received: December 21, 2025
Accepted: February 2, 2026
Published: February 5, 2026

Abstract

Effective task scheduling can significantly impact performance, productivity, and profitability in many real-world settings. Deep learning offers promising avenues in image recognition; robots can autonomously learn and optimize their actions, enhancing their performance in complex settings. Optimizing task scheduling is crucial. In particular, reinforcement learning is a promising task scheduling approach because it can learn from experience and adapt to changing conditions. Research analyzes the performance possibilities of task scheduling by using deep analysis; it allows for the selection of highly efficient environment models by using neural networks (CNNs) and recurrent neural networks (RNNs). Deep analysis allows for the selection of highly efficient environment models. Moreover, automatic selection based on optimization algorithms has been proposed. Furthermore, in this research, our finding aims to extend the applicability of deep learning in robotics, contributing to the design and implementation of future intelligent systems and improving robotic systems' environmental perception, decision-making, and actions. To understand non-collision robot movement in unknown dynamic environments, intelligent algorithms have been proposed, showing considerable improvement in failure rate and path length. To recognize objects, navigate in complex environments, and facilitate decision-making based on real-time data, deep learning (DL) can be used to enable robots to execute and perform various complex tasks while moving freely. Furthermore, patterns have been recognized in robotic movement paths. Therefore, the authors developed a novel hybrid intelligent approach that allows and provides superior efficiency regardless of environmental parameters.

Keywords

Deep Reinforcement Learning, RNN, Robot Navigation, Service Level Agreements, Task Scheduling

1. Introduction

As human-machine collaboration becomes an essential study for mobile robot navigation [1]. Another possibility to solve the task scheduling problem is the application of machine learning techniques [2]. Machine learning (ML) is a branch of Artificial Intelligence (AI) that allows applications to learn without being explicitly programmed, gradually improving their accuracy from various types of data. Reinforcement learning (RL) is the only category of ML that learns by interacting with the environment, not by provided datasets. Multi-intersection traffic signal control [3], humanoid robots [4], and many others. It has many different algorithms like Q-learning. In crowded environments, it is increasingly becoming an important aspect to consider. The algorithm is not recommended for relatively complex problems [5], and implementing such an algorithm is fairly easy. However, selecting suitable parameters for each application is not preferable. Unfortunately, the literature often lacks presentation of chosen parameters [6] [7]. The traditional navigation stack consists of positioning a global planner, which, given a global map, calculates an optimal path in highly dynamic environments [8]. On the other hand, unpredictably moving obstacles to avoid detection collisions that have been recorded in certain environments is difficult since the future motion of the obstacles is unpredictable [9]. Deep Reinforcement Learning (DRL) has emerged as an end-to-end method that has demonstrated superiority for obstacle avoidance in dynamic environments and for learning complex behavior [10] [11]. Mobile robots require an accurate understanding of their environment to act autonomously. Consistent simultaneous localization and mapping (SLAM) is a fundamental basis for this, which involves constructing a map of the environment from sensor data [12] [13]. In order to increase efficiency and added value in component production processes utilizing operations [14], AI (Artificial Intelligence), ML (Machine Learning), and DL (Deep Learning) applications have brought about significant advancements in the field of robotics [15]. Machine learning algorithms can help robots to predict when maintenance or repairs are required [16]. For further improved decision making: AI and ML algorithms can analyze large amounts of data and make informed decisions based on that data, allowing robots to consider and make better decisions and take appropriate actions in real time [17]. Better decision-making: AI, ML, and DL can enable robots to make better decisions based on data analysis and pattern recognition, leading to improved performance and outcomes [18]. Technologies can enable robots to adapt to changing environments and situations, making them more versatile and capable of handling a wider range of tasks [19]. Using AI, ML, and DL can enable robots to make better decisions based on data analysis and pattern recognition, leading to improved performance and outcomes applied in the system criteria [20]. Technologies can enable robots to adapt to changing environments and situations, with these techniques making them more versatile and capable of handling a wider range of tasks [21]. Robotics applications often require robots to operate in dynamic and changing environments, which need adaptability [22]. AI/ML/DL models must be designed to adapt to new situations and learn from experience, with the need for robots to be able to operate safely and effectively in a wide range of environments [23]. As robots become more autonomous and interact with humans, ensuring their safety becomes a critical challenge. AI/ML/DL algorithms are designed to avoid collisions with humans and other objects [24]. The frameworks can support the safe and effective use of these technologies. There are many potential applications of artificial intelligence (AI), machine learning (ML), and deep learning (DL) in advanced manufacturing robots. AI algorithms can identify defects in products and alert the production team to make necessary adjustments in real-time. This helps manufacturers to identify and eliminate bottlenecks, reduce waste, and increase productivity [25]-[28].

2. Background and Related Work

Redmon and Farhadi [29] and Joo *et al.* presented object recognition using YOLOv3, which can compensate for the low accuracy of ultrasonic sensor mobile robots. Wang Liguo [30] proposed an improved K-Medoids algorithm based on manifolds and applied it to the clustering practice of hyperspectral images. Hongpeng Fu [31] studied the evaluation method of improvement based on k-medoids clustering. Object detection recognition using deep learning neural networks with a massive amount of labeled data, enables the robot to identify and classify objects in their environment with higher accuracy [32]. Predictive maintenance of machine tool systems using artificial intelligence techniques applied to machine condition data [33]. Speech recognition are also important applications of AI and ML in robotics [34]. For example, robots like Pepper can recognize and respond to human gestures and speech. AI algorithms can be used to identify potential hazards or obstacles in a ship’s path and make course corrections to avoid them [35]. T. Le Nguyen and T. T. H. Do used artificial intelligence in healthcare [36]. M. El-Shamouty, K. Kleeberger [37] recognize objects, navigate complex environments, and even make decisions based on real-time data. DL can be used to enable robots to perform complex tasks. Kawaharazuka, K., Miki, A [38] stated that using deep learning, for instance, may be used to teach drones how to detect and avoid obstacles in real time. C. Chinag and C. Ding (2014) [39] presented a robot navigation method for a dynamic environment with static and dynamic obstacles. The robot avoids the static obstacles by utilizing a logic controller. These articles explain Methods for detecting people, tracing that use features such as color histograms obtained from RGB images [40], and those that use Kinect features [41], have been proposed. Guilherme and Avinash [42] refer to the construction of an environmental map, where we must detect the planar area in the image sequence observed by a camera mounted on a mobile robot. Since these methods are dependent on the environment around a robot, they involve difficulties, such as the necessity to predetermine. A deep reinforcement learning-based wireless body area [43], using mobile chargers to recharge the sensors, is becoming more common as a dependable method of powering the sensors using wireless power transfer technology. [44] refers to optimization approaches that necessitate lengthy procedures and iterations.

3. Research Methodology

Study categorized modern robots as endowed with rich, high-dimensional sensory systems, providing measurements of continuous environments. In this review, the authors cover some fundamental model-free RL algorithms and path-breaking function approximation-based deep RL (DRL) algorithms for complex uncertain tasks with continuous action and state spaces, making RL useful in various interdisciplinary fields. Therefore, reinforcement learning (RL) methods have shown promise for the automatic learning of robot emotion behavior. These methods, when applied to high-dimensional, continuous, high-diameter problems, still present a major challenge. Thus, dependence on human analysis of the robot, environment, and task is required to provide high-dimensional accuracy. Therefore, RL has emerged as an efficient technique for solving complicated sequential decision-making tasks. The survey presented in this paper provides an overall understanding of RL foundations. In this survey, the primary goal is to give new researchers comprehensive subject information about the RL approach. We conduct a comprehensive survey of the historical progress. Therefore, the process in their study mixed a heuristic approach of the artificial intelligence algorithms (RNNs) to find ways of optimization. The problem of enhancing the navigation of obstacle-avoiding robot models, along with a comprehensive analysis and comparison of model-free deep reinforcement learning (DRL) algorithms, is addressed. The survey gives a brief overview. Therefore, reinforcement learning (RL) methods have shown promise for the automatic learning of robot emotion behavior. In this survey's analysis of new algorithmic techniques for model suggestion, it is shown that robots can become attractive and can control speed through vision algorithms. The technique evaluates deep neural preference of objects based on object detection applications, attaining accuracies during obstacle detection. It is difficult to replicate these results for problems in every field. Each learning problem has its own set of challenges and constraints. When using (DRL) for optimization problems, there are plenty of practical issues. The outline of this paper is as follows: some basic concepts of RL are discussed in Section 2. Section 5 covers the model-free RL algorithms for discrete state and action spaces. Section 4 gives the idea of value function approximation in RL. Some function-approximation-based policy prediction and policy control algorithms are also discussed in this section. Section 5 covers some important model-free DRL algorithms and some model-based approaches. Section 6 describes the current research work and some important research fields in RL. Finally, Section 7 draws some conclusions.

4. Problem Formulation

One of the biggest challenges for mobile robots is visual perception and image classification alongside interaction in the dynamic environment in the real world [45]. Object detection is a crucial robot overcome prevailing complication. Robot vision systems train to accomplish performance of complex tasks. Therefore, robots in the dynamic area need to detect any dynamic obstacles in real time. For the robot action to function properly, observation and integration using the navigation system are required. Autonomous mobile robot (AMR) technology, additionally, more accurate sensors may affect the quality of information received by AMR [46] [47]. Perspective algorithms for robotics with object detection, pedestrian detection, and obstacles detection. The framework can detect this problem using deep learning for two stages: convolution neural network (CNN), or faster region convolutional neural network (Faster R-CNN). The prediction results satisfy the search object. The path planning algorithm proposed by this paper takes the next step in this specific research field by presenting a new solution based on a heuristic and neural networks that can be regarded as a "self-learning" paradigm in solving real-parameter optimization challenges. However, the effectiveness of a single algorithm varies across different problem instances, necessitating considerable effort in algorithm selection or configuration. The reward for a collision-free trajectory is given if the mobile robot reaches the goal. In other words, the proposed solution samples each state, action, and result from the workspace as an underlying probability distribution, which helps in calculating the reward parameter [48].

Therefore, **Figure 1** represents a model of the robot navigation methodology. It enables the movement of the path selected from the location prediction sensors; accuracy depends on the sensor region information. However, the robot can predict movement in the stage of image object after processing; see **Figure 2**. The figure explains human estimation and action recognition; this enables robotic trajectory movement obstacle recognition [49]. On the other side of this study, the robot can recognize face and voice physiological recognition and movement control.

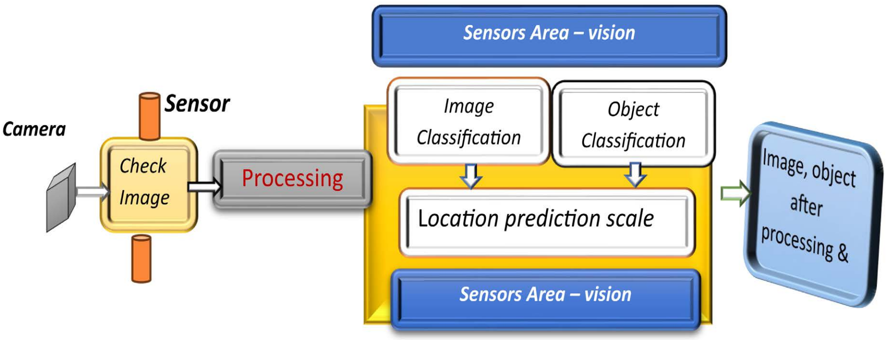


Figure 1. The standard process of object detection.

Figure 2. Analysis of the research conception.

Robot action recognition involves identifying and classifying specific trajectories, movements of any obstacles, or behaviors performed by a person. In this way, robot understanding concerns the capability to comprehend natural human voice input. Due to this fact, effective communication and collaboration between environmentally free-moving entities are essential, and the simulation models focus on developing techniques that can recognize images, interpret human emotion, and enhance the ability of trajectory movement to avoid closed areas through intelligent interaction controls [50] [51]. In simulation design, the velocities are used to calculate the known point of the robot's direction. The model has learned to navigate toward the target. Our objective is to show how the robot can trace and reach the goal when there are no large obstacles. The main objective in this work is for the robot to learn to avoid obstacles and collisions during the simulation activity, as calculated using the techniques shown in **Figure 3**. Meanwhile, we employ a sophisticated deep neural network model to infer the optimal action, ensuring informed algorithm selections. Additionally, an algorithm context restoration mechanism is embedded to facilitate smooth switching among different algorithms.

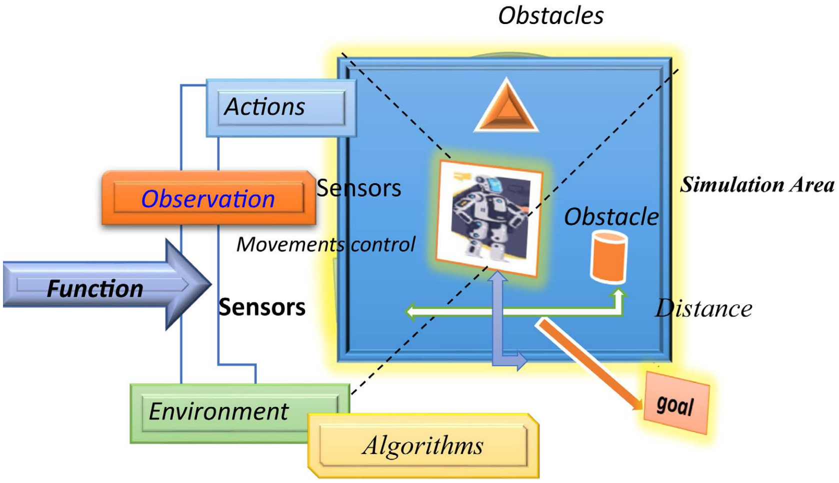


Figure 3. Hybrid model action movement evaluation.

5. System Design and Training Procedure

In Figure 4, the robot in the simulation area has two types of obstacle detection classified as dynamic and static objects. The process of robot movement depends on the information sent through the area of sensor location, which makes the moving path calculated within the path distance, and the number of obstacles calculated as the time for the robot to reach a goal. The algorithm classifies the objective work in the area of simulation. Furthermore, the area of the simulation is fixed at 200 × 200 cells divided equally into 4 blocks, with known and additionally unknown static obstacles. The stages were introduced to increase the difficulty of the environments as the robot's training performance increases. The robot is trained in the beginning in an uncomplicated environment to learn basic movements more easily. Later, the robot trains in a challenging environment to learn more complex behavior. These mechanisms together enable our framework to seamlessly select and switch algorithms in a dynamic online fashion. Notably, the proposed framework is simple and generic, offering potential improvements across a broad spectrum of evolutionary algorithms.

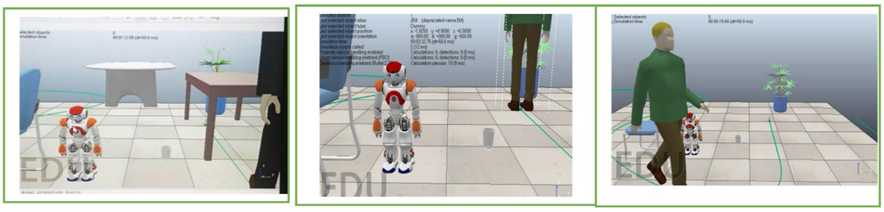


Figure 4. Robot training recognition.

In the area of the control system for the movement and function of various parts of the robot to execute specific functions from the set mentioned, the level of interaction adapts to the robot's existing control system implementation in robotics. Furthermore, radio-frequency identification (RFID) enables the robot control system to implement complex integrations of the active RFID system. Furthermore, the Kinect camera is utilized to capture human movement; it is a type of control connection motion framework remotely. Its process of recognition allows each gesture to correspond to different robot commands [52] [53]. In Figure 5, the camera connection is shown; as a proof-of-principle study, we apply this framework to a group of differential evolution algorithms.

Figure 5 shows the model of the robot motion classification recognition, feature extraction output, and image processing. The figure is connected through a vision camera for checking and verifying patterns in the area of the robot simulation. However, in this article, the research focuses on the classification of different aspects of deep learning (DL) techniques, modeling and learning capabilities of DL techniques, such as supervised and unsupervised tasks. The table below provides a summary of deep learning tasks and methods in several popular real-world application areas. The summary of deep learning tasks is provided in Table 1, which gives a general structure of the transfer learning process.

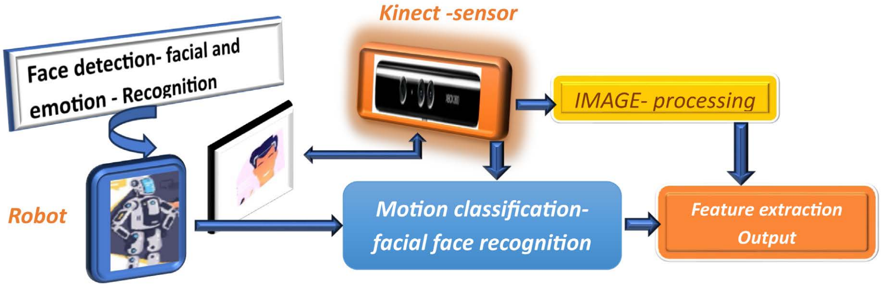


Figure 5. IMGE based robot control system.

Table 1. A summary of deep learning tasks and methods.

Application area	Tasks	Methods	Reference
Speech recognition	Distance speech recognition Speech emotion classification Emotion recognition from the speech	Attention-based LSTM Transfer learning based CNN and LSTM based	Zhang *et al.* [52] Latif *et al.* [53]
Object Detection and Recognition	- Object detection in X-ray images. - Object detection for disaster response - Medicine recognition system -Face recognition in IoT-cloud environment - Food recognition system - Affect recognition system - Facial expression analysis	CNN-based CNN-based CNN-based CNN-based CNN-based CNN-based	Gu *et al.* [54] Pi *et al.* [55] Chang *et al.* [56] Masud *et al.* [57] Liu *et al.* [58] Kawde *et al.* [59] Li *et al.* [60]

The following **Figure 6** represents a general structure of the transfer learning process, with contents including a dataset for providing learning tasks, pre-training, dependence on the DNN, robot position, velocity, and acceleration.

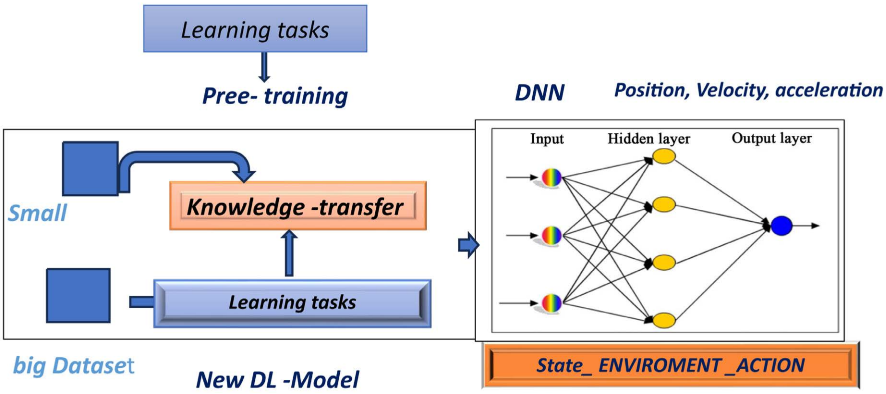


Figure 6. A general structure of the transfer learning process.

Based on this study, the present conceptual design of an autonomous robot system is proposed. The structure of operation control uses artificial technology, high techniques, and IoT control system to implement a smart and intelligent robotic system [54]. The control of the robotic system comes from the microcontroller to the servo motor for navigation. The microcontroller runs based on fuzzy logic control algorithms for navigation. In addition, the algorithms navigate image processing to check and determine the action frame of the environment, and the flowchart proposed for the system architecture depends on the robot movement, object detection calculation, and pattern recognition. Presented in Figure 7, controlled by a microcontroller [55] [56]. This controller serves as the central processing unit of the system.

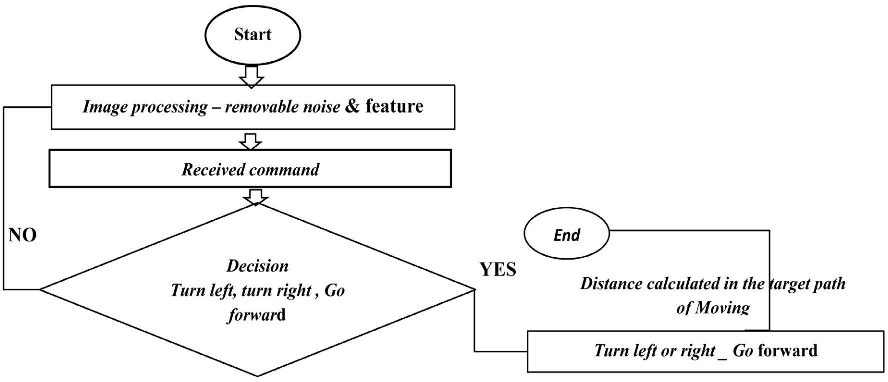


Figure 7. Steps of the operation of the robotic system.

The model in Figure 8 creation process, transition recognition using Principal Component Analysis, presented the area of the point of the gathering position of the target follow diagram by using pattern recognition.

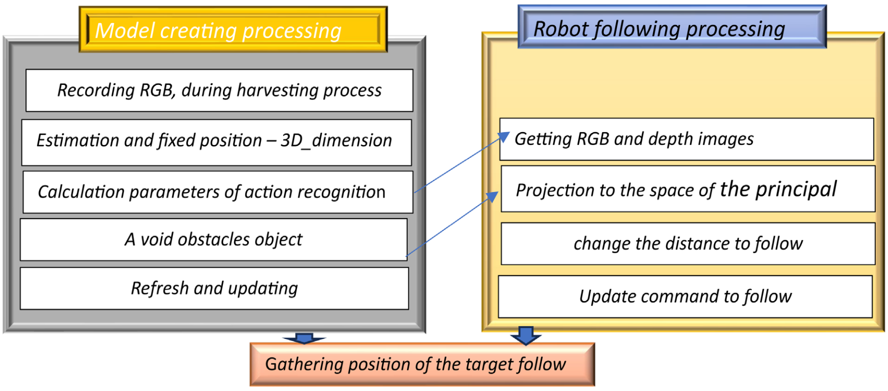


Figure 8. Flow diagram of the follow system using action recognition.

The research objective is to provide a solution for a complex environment, enabling the robot to make optimal decisions for avoiding static and dynamic obstacles, based on the velocity vector of the obstacles. It is capable of planning navigation in a complicated dynamic environment [57]. Sensor-based planning tasks make the suggested approach more effective and intelligent. Furthermore, intelligent algorithms using a fuzzy logic controller assess the prediction procedure and decide and choose the next step based on several criteria. These criteria are prepared by the rules and have been transformed into a fuzzy details planner [58]. See the scenario system situation in Figure 9 and Figure 10.

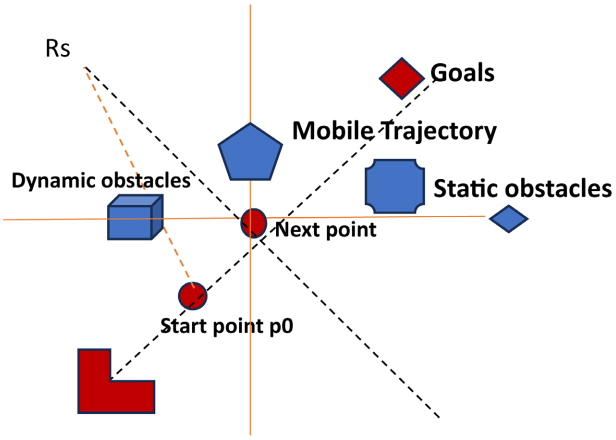
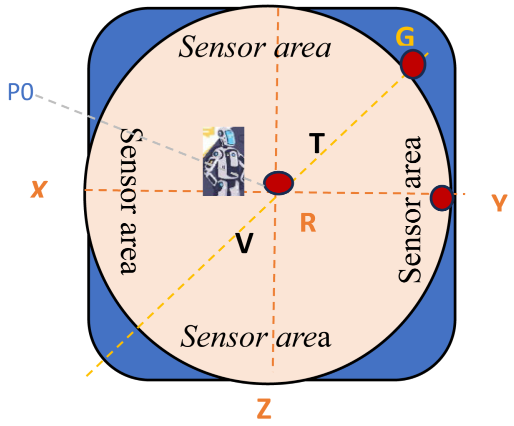


Figure 9. Robot navigation issue in complex dynamic environments.

Based on a good localization technique, the robot can determine precisely its position with respect to the changing environment (usually, a configuration known by space start point—x, is employed to describe the next point x + 1, all possible configurations of the robot; presuming the navigation takes place in a 2D, 3D workspace, a fixed point is divided into 4 sectors. See the scenario system situation in **Figure 10**.



X, Y, Z, sector direction 180, 90 – R = Radius – T = Trajectory – V = Velocity, T = time estimation, w = width, G = goal, po = start point. Where X, Y, and Z represent the direction constraint in our model.

Figure 10. Simulation Area with static and dynamic objects.

However, in the figures represented **Figure 9** and **Figure 10**, a collision has either a lower or higher risk and an unobstructed direction. The robot or a Wheeled Mobile Robot (WMR) is derived through sensor and camera scanning, shaped from the center as shown in **Figure 10**, (r_x, r_y), front wheel $r_c(t) = (r_x(t), r_y(t), r_\phi(t))$, shows the simulation configuration square that is autonomously driven. Moreover, the robot at time t, $r_x(t)$, $r_y(t)$ determines the coordinates of the robot's center in the robot simulation area. It becomes rotates around the origin. Moreover, $r_\phi(t)$ shows the orientation of the robot, as measured by its angle in relation to the positive X-axis.

$$X = v\cos\phi \tag{1}$$

$$Y = v\sin\phi \tag{2}$$

$$\Phi = W$$

$$p_0 = \left(x_0, y_0, \phi_0\right) \tag{3}$$

Robot trajectory has been selecting a short path length; it is presumed of the velocity robot moving calculate in the equation below

$$v_x\left(t\right) = v_y\left(t\right) = 0 \tag{4}$$

$$\omega\left(t\right) = \phi\left(t\right) = 0 \tag{5}$$

The goal position, perspective, and distance:

$$\phi = \tan^{-1}\left|y_2 - y_1\right| / \left|x_2 - x_1\right| \tag{6}$$

$$d = \frac{\sqrt{X^2 - X^1}}{Y^2 - Y^1} \tag{7}$$

$$\Delta V = \left|v_{xi} - v_{xj}\right| + \left|v_{yi} - v_{yj}\right| \tag{8}$$

$$\text{The state of intersection points of circle} = \left(x + y\right)^4 = \sum_{p=0}^{4} \binom{n}{p} \tag{9}$$

The distance of the robot to the destination should be decreased when the robot keeps moving forward in the navigation. The point position near should be (N_1) and should be compared to the other location of the robot (N_2), expressed in the equation constraint as:

$$\text{Next position}\left(N_i\right) = \text{Current position}\left(C_{i-1}\right) + \text{Step} \times \left[\cos\phi_i, \sin\phi_i\right]^{\mathrm{T}} \tag{10}$$

While the current position plus the next position equals the target distance G plus the future obstacle trajectory.

Figure 9, 180˚ and −90˚ and fourth −90˚ and 0. The robot should record the intersection points between the positions of the obstacle and these circles. Moreover, determines which one of these regions has the intersection points [59] [60]. The sensor reading and motley closed area and fixed records, one for any position inside the obstacle and zero for any position outside the obstacle, record to estimate the future trajectory of the moving obstacle. Then the system stores the result in a visibility matrix, which includes the obstacle points position and evaluates the speed vector of each process as follows; see Table 2, as an example. Therefore, pseudo code algorithm for estimation and target position is provided in the table below.

Table 2. Accuracy time estimation.

Time	Static obstacle	Dynamic obstacle	Right obstacle	Left obstacle	Front obstacle
$T = t + 1$	0	1	0	0	1
$T = t - 1$	0	1	1	1	1

T = 1: present robot on the dynamic movement; T = 0: robot on the static movement; t + 1: forward movement; t -1: present backward movement.

Pseudo code algorithm

```
Input: initial position and target position
Output: calculate the path from the start point to the end point
While (the robot does not have clear navigation);
Navigating and scanning, the sensor collects clear information.
The sensor detects objects and obstacles. The sensor keeps monitoring.
If there is no obstacle around the robot, then
Goal seeking and other objects;
End if
If an obstacle is detected by the sensor, then
Record obstacle's position;
Record positions of the obstacle and fix all points.
If the number of intersection points does not alter, then
The obstacle is static.
Determine the obstacle's position by determining which one of these regions has the intersection points;
If the position of the obstacle crosses its path
Do the static avoidance.
Find the best next position depending upon the priority limitations and nearest;
End if
End if
Else if the number of intersection points alters , then
The obstacle is dynamic.
If the number of intersection points increases , then
The direction of the moving obstacle is toward the robot.
If the future trajectory of the obstacle crosses its path, then
Perform dynamic avoidance behavior using the predicted velocity vector of the moving obstacle.
End if
Else if the number decreases , then
The obstacle is moving away from the robot.
Find the best next position far from the dangerous region depending upon velocity vector rules;
End if
End if
End if
End if
End while
```

And therefore, if there are some specific conditions mentioned from the previously mentioned equation, step 2, suppose that $X \to |0,1|$. X is a variable, $x_{i,j}$ are parameters.

$V \xrightarrow{volicty} vehicles$

$X(x_{i,j}) = \frac{\Delta d}{\Delta t} = N$, N= distance. $X \in x_{i,j}$

Then

$\forall x_{i,j} \in X, \exists! X_{i,j} \neq 0$.

1 if $(v_n, x_{i,j}) \leq v_{\theta_{low}}$, then $v_n(t+\Delta t) = v_n(t) + d(t) + 1$;

2 If $v_n(t+\Delta t) \geq v_{\theta_{high}}$, then $v_n(t+\Delta t) = \Delta t(i+1) - (i) - 1$;

3 if $v_{n,x_{i,j}}(t+\Delta t) \leq 1$ and $\theta \neq 0$, then $v_n x_{i,j} = (t+\Delta t)$, $d(i-1)$,

$X = i+1, i-1, i++$; then/the probability $p\left(v_{x_{i,j}}, \Delta t\right) = i++$, $i=0$, $i \neq 0$, $i \leq 0$, $i \geq 0 = v_i(t+\Delta t) - i$; v_i, $x_{i,j} = \frac{\Delta t}{\Delta d} = N$;

Where (i), defined the cell by vehicle i, N= distance

$$v_{i+1}x_{i,j}(t) > v_i x_{i,j} + \Delta$$

$$v_i x_{i,j}(t+\Delta t) = v_{i+1}(t) + \frac{\Delta x_{i,j}(t) - v_{i+1}(t)\delta(t)}{\varepsilon(t)\{v_i(t) + v_{i+1}(t)\}/\Delta d\pi}$$

6. Algorithm Proposed

In the demand for optimal searching of robot navigation services, heuristic research algorithms in a graph are identified from path location at the point where the robot is located, as shown in this target searching planning, see Figure 11 and Figure 12, if we suppose that the current point of the robot is located at track node A. The track position of movement is presented in the graph location by the name of the character A. However, path location can be calculated through algorithm path distance and priority navigation.

6.1. Search Strategy

The analysis process of the search strategy using specific intelligent knowledge is informed by a heuristic search technique, with the algorithms working by combining different algorithms such as best first search (BFS) and depth first search (DFS). At each step, an appropriate heuristic function is used.

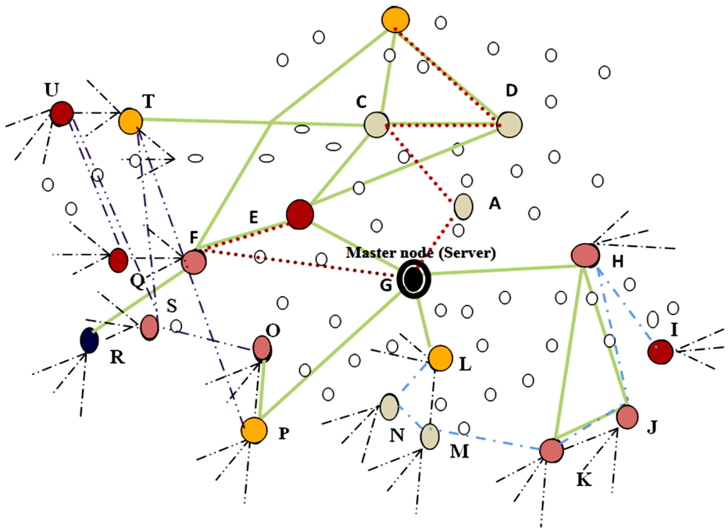


Figure 11. Network priority connection selection—greedy tree.

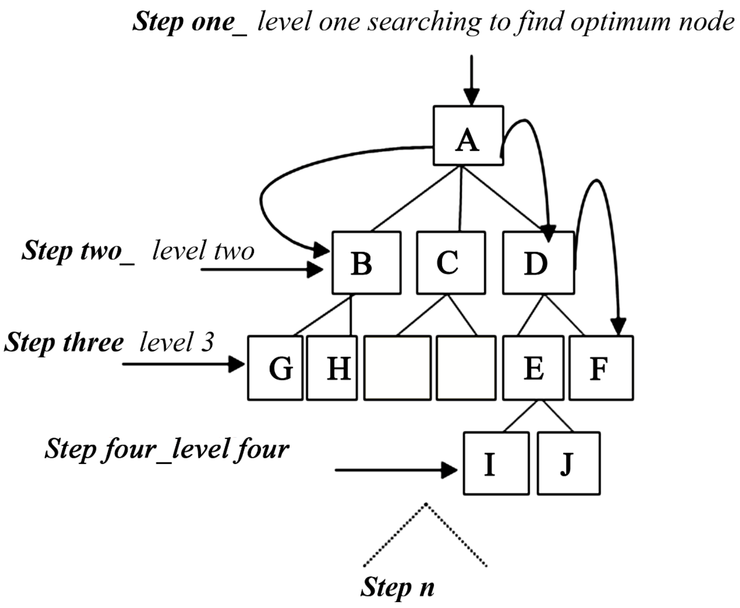


Figure 12. Represent heuristic deep research techniques.

6.2. Procedure (Path Distance Priority Navigation)

```
Procedure dB_ search (n₀, ………., n_c(goal)): path, bound: int):
If goal (n_c) = 0, report path (n₀…………, n_c);
If bound > 0
For each neighbor n of n_c
dB_ search (n₀, ………. n_c, n: goal node n+1, n-1);
else if n_c has a neighbor, then target node: = true or false;
end procedure dB_search;
Procedure id_ search (node):
Integer: = 0, 1;
repeat
Connect = true and disconnect = false;
dB_search (node_goal);
end procedure id_search
```

The cost of a path can be presented by the following equation:

$$\mathrm{Cost}\left(n_0,\cdots,n_c\right)=\sum_{n=1}^{\infty}\sum_{n=0}^{\infty}\mathrm{cost}\left(n_{i+1},n_{i-1}\right)$$

The target here is to find a solution of the lowest cost from the main equation:

$$\mathrm{Cost}_{\mathrm{lowest}}\left(n_0,\cdots,n_c\right)=\sum_{n=\infty}^{n}\mathrm{cost}_n=\mathrm{target}_n$$

Where n = node;

The goal of heuristic search h(n) is to estimate the cost of the optimal (cheapest) path from node n to the goal node, as presented in the following Figure 13.

The priority queue is ordered by probability (p) = cost(p) + h(p). Furthermore, since the goal h(n) = 0, and the cost f(s) = cost(n) > $f_{\min}$, we assume the hypothesis for all paths is cost(n) > $f_{\min}$. The optimality of selecting nodes for all the paths p on the greedy network f(p) ≥ f(n) > f_{min}.

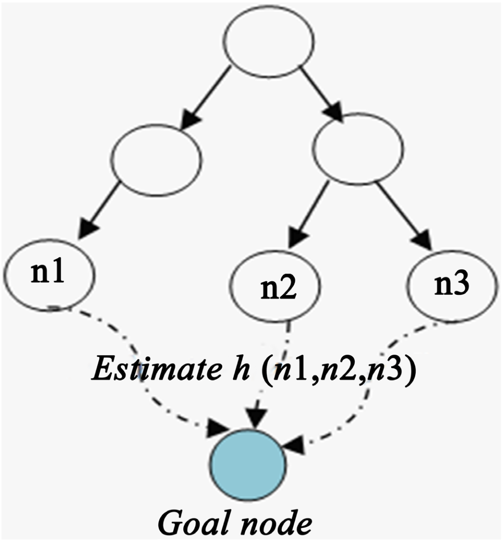


Figure 13. Represents heuristic research techniques for node priority.

Depth-bounded depth-first search looks for solutions at each depth (depth 1, then 2, then 3, etc., while ignoring longer paths). We propose a deep reinforcement learning-based dynamic algorithm selection framework to accomplish this task. Our approach models dynamic algorithm selection as a heuristic research algorithm process, training and has been classified in the search for the minimax in a policy gradient manner to select the most suitable algorithm according to the features observed during the optimization process. We propose a new path planning algorithm based on the use of artificial neural networks, see **Figure 14**.

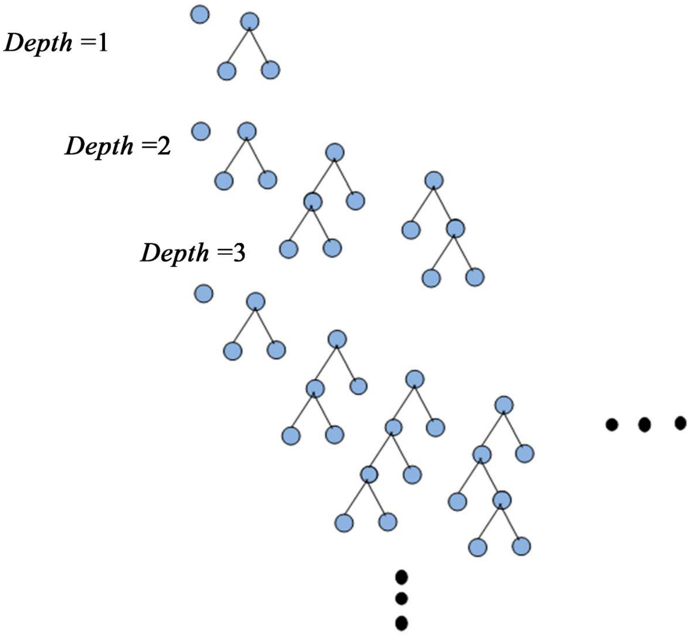
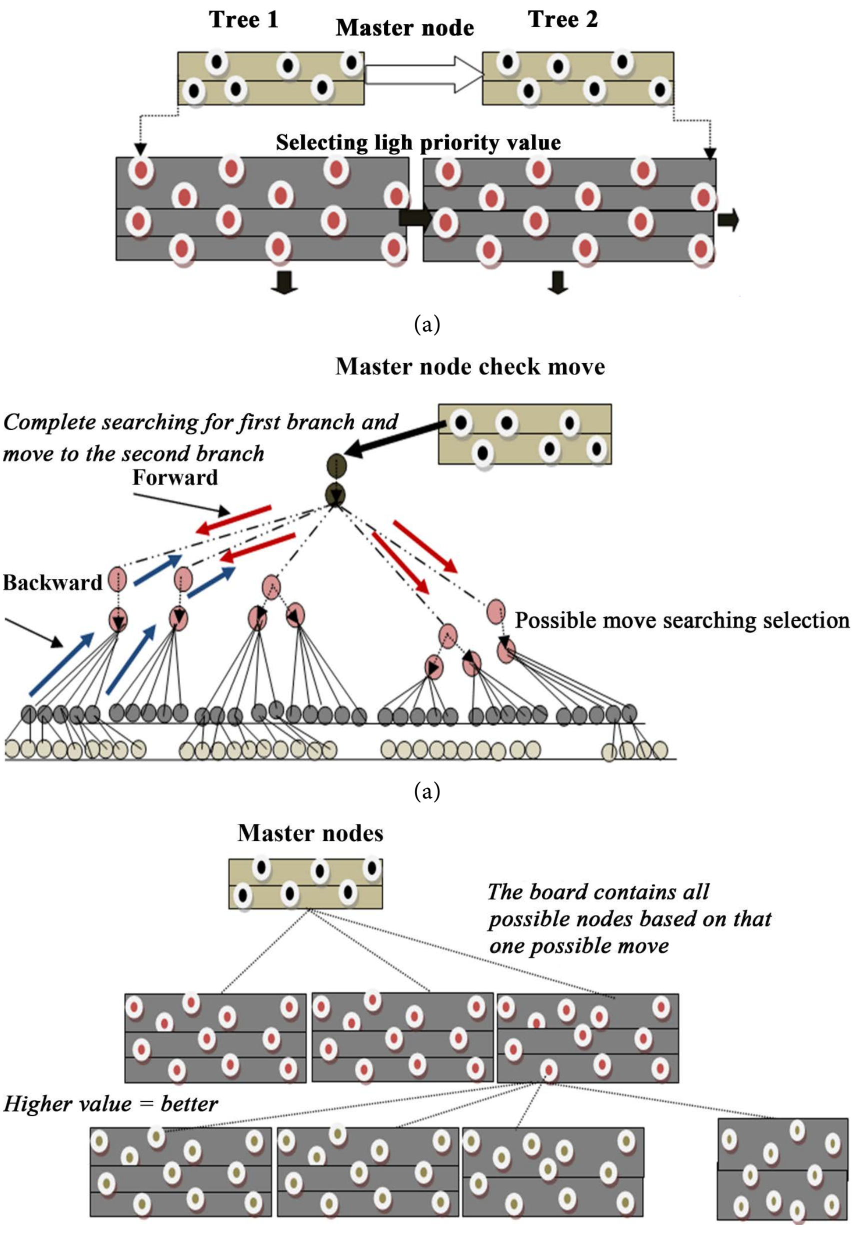


Figure 14. Represents heuristic hierarchical deep research levels.

Figure 14 has been classified in the search for the minimax, applied as shown below in Figure 15, as indicated in Figures 15(a)-(c).

6.3. Minimax Applied

The algorithms choose only one of the possible moves each turn and will have to consider all possible moves. In addition, any step of the algorithms starting from the current state (black node) is defined by the master node. The process of the search is defined as:



(a)

(a)

(c)

Figure 15. Example of [a] heuristic deep research technique: (a) Applied minimax technique, (b) Searching using forward and backward techniques, (c) Selecting higher probability values.

The algorithm works by following the steps for searching the navigation as in the following proposed searching algorithm.

6.4. Algorithm

```
1. Start searching from the master (node) current [state];
2. Select branch one from the tree.
3. Find the target node until the end of the branch [childes node];
4. Select high values, then continue deep search forward_[child nodes], until the end of the branch [∞];
5. backward [current state]_change the path
6. Repeat step no. 3, 4 and step no. 5;
7. Continue searching until the end of all nodes in the tree graph.
8. end;
9. Local variables: current, a node
10. neighbor, a node
11. current ← Make_Node (initial-state_target$_{goal}$)
12. Loop
13. Generate all the successors of all nodes
14. If any solution = solution, then select the best success.
15. neighbor ← a highest_valued successor of current
16. if value[neighbor]≤Value[current]then
17. return state[current];
18. search ← different paths
19. if value[neighbor] ≥ Value[current]then
20. Select state [a highest];
21. search ← current [state]
22. deep_search child nodes
23. If value [child nodes], exit state [highest value]
24. do select
25. If value [child nodes], do not exit state [highest value].
26. return state[current]
27search ← different paths
28. forward_search_[child nodes]
29. until ← the end of branch [ ∞ ]
30. backward ←initial [state nodes]
31. Search _neighbor nodes [∞]
32. Goal node ← highest value
33. forward←select _node success
34. backward ← select_node fail
35. current ← neighbor
36. end;
```

Furthermore, the following work depends on the previous Figure 14, algorithms can be selected for the goal in Figure 16. As an intelligent work follows techniques navigator.

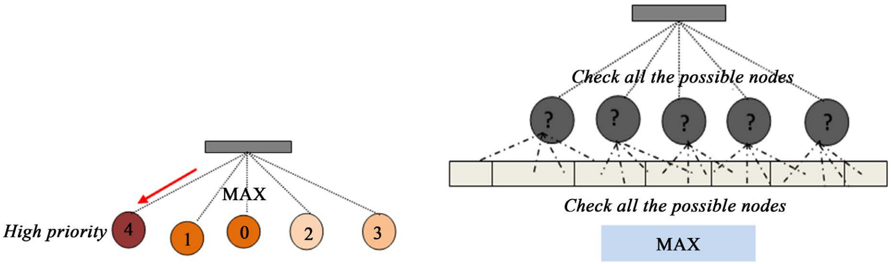


Figure 16. Example of intelligent techniques based on minmax (selection higher priority), for checking all possible paths connection, and selecting the goal of higher priority.

In this algorithm, the highest value is given the highest priority, while in the searching process the algorithm is finding the target goal. The Best First Search algorithm uses a list to maintain all states {open and close} such as:

Open: to keep track of the current search

Close: to record states already visited, according to the heuristic research estimation, and each iteration of the loop will be considered. The following figure describes the optimization of the networks.

6.5. Mathematical Explanation

Let $N = \{1, \cdots, N\}$ be the set of index classes of domain nodes, X is the target of class indexes, k is the set of dynamic or static objects generated from the simulation area of the robot position. Let x, y be parameters of coordinators $x_i, y_i \in N$, p represents the probability of point selection p_{ij}. $N(C)$ are the neighbors (setoff).

$$K_{select} = \sum_{x=1}^{n} \sum_{x=0}^{n} N(x, y)_{n_i} \to f(\pi) \approx \sum_{i=1}^{n} p_{ij} \cdot d\pi(i)\pi(j)$$

π represents the assignment of the facility location, and the element (i, j) represents the facility at the position of the location of robot movement.

Where their neighborhood N, it can be mathematically expressed as follows:

$$N_{networks} \in Map(Nodes, Sub(nodes))$$

$$N_{K \in Nodes} \approx \{X, Y \in Nodes \mid p_{ij}, x_i, y_i\}$$

$$N(C) \subset Nodes \approx \{X, Y \mid x_i, y_i (\exists x \in N(c))\}.$$

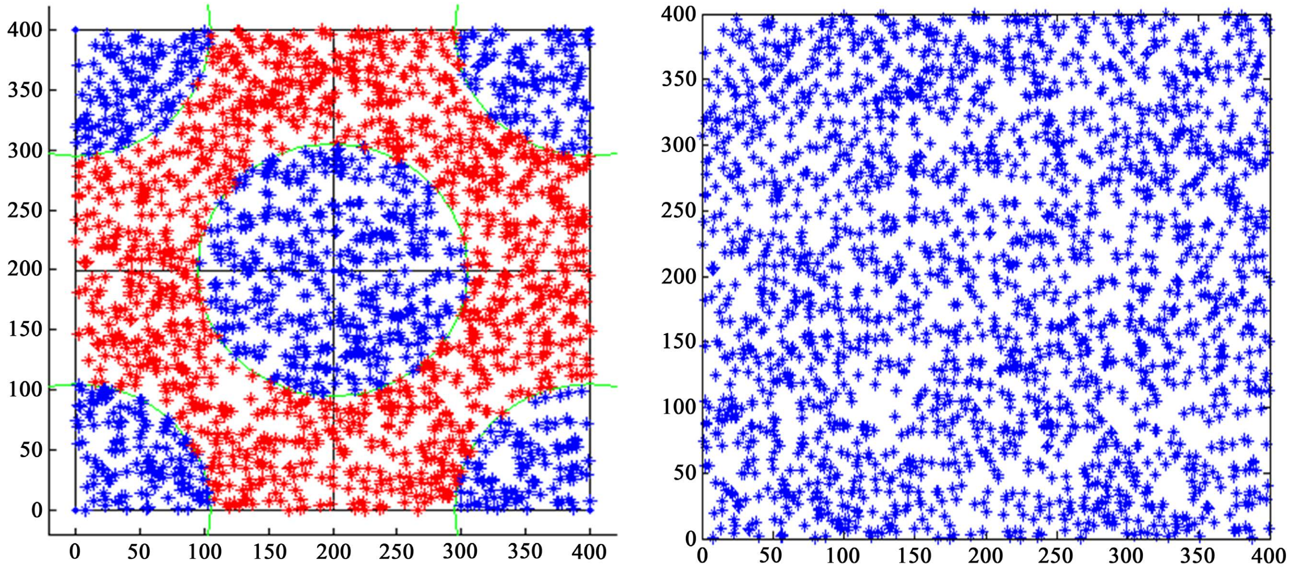
Where the neighbors (setoff) = $\{N_{x_i, y_i} \mid p_{ij \in 1,0, I \in 1,0}\}$

$$(\forall x_i \in 1,0)(N_i \neq 0) \nu \bigcup N_i = Nodes$$

$$\mid p \prod_{x,y}^{n} N \xrightarrow{nodes} \left(\exists^{\max_{i,j}} \in 1,0 \mid i \neq j\right)\left(N_i \cap N_j \neq 0, N(C) \in N \subset N(K)\right)$$

6.6. Simulation Results

As expected, the performance of the trajectory planner presented above depends on the configuration of the computer used for simulation. The experimental results showcase the remarkable effectiveness of the proposed framework, not only enhancing the overall optimization performance but also demonstrating favorable generalization ability across different problem classes, as described in the following Figure 17 and Figure 18.



(a)

(b)

Figure 17. (a and b) describes simulation objects: red and blue color. The red color represents static obstacles, and the blue color presents dynamic objects.

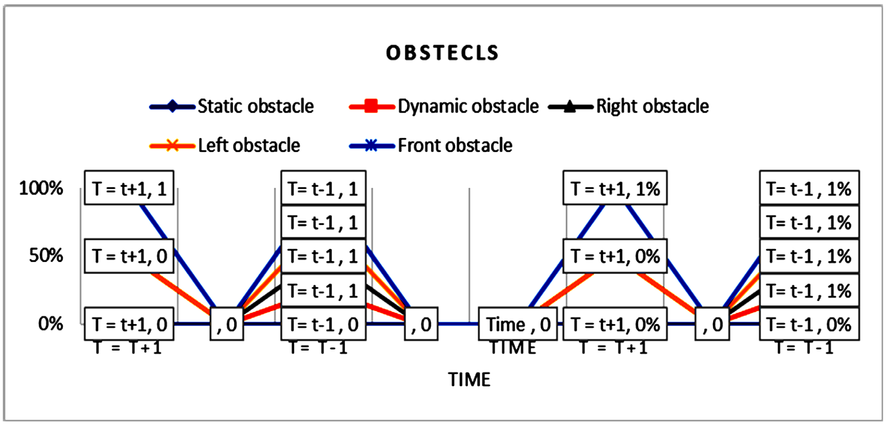


Figure 18. Describe the percentage of the objects, dynamics/static where the time presented in T = t + 1, T = t − 1, shows estimation time of the robot movement obstacles direction.

The first approach uses global knowledge of the environment, meaning that at each moment, the robot has complete information about its location, movement capabilities, obstacles, and target. This raises additional problems related to localization. In few-shot object detection (FSOD), image pyramids enrich object scales as data augmentation, and the sparse scale distribution can be theoretically solved. However, this multi-scale training strategy acts differently with the increasing number of improper negative samples. See Figure 16, red bounding colors, which present negative objects paththrough. However, samples in training may actually contain part of objects and may even be true positive samples in other contexts (as in the bottom right). These improper negative samples require sufficient context and clues to suppress. Figure 18, presented there, shows that accuracy performs scale normalization to detect scale-specific objects and adopts image pyramids for multi-scale detection. These studies generally aim to alleviate large size differences of objects. In addition, few-shot object detection suffers from scale variations in a more serious way, where a few samples are sparsely distributed in the scale space.

The task of determining the least or maximum value of a given function can be viewed as an optimization problem. For instance, if we consider a function $f(a) = 0,1$, we can determine that its minimum value, $f_{\min} = 0$, occurs at $a = 0$ in the entire domain of -infinity < a < infinity. However, for simpler functions, we can determine the potential solution by setting the first derivative, $f(a) = 0$, to zero. v, velocity changes depend on the robot's movement. In addition, the simulation in Figure 17. RL is based on trial-and-error learning through rewards and punishments. The ultimate goal of RL is to maximize cumulative reward.

Optimization

In the domain of optimization, a task that involves minimization or maximization can

Be expressed as a problem.

minimize $fx(1),\ldots, fx(0),\ldots, f(x)$, $a = (1, 0)$

subject to

$px(a) = 1, (a = 1, v = t + 1)$

$px(a) \leq 0, (x = 0, v = t - 1)$

Furthermore, Figure 18 presents and shows the percentages of the two sides of the object obstacle in the stages of the dynamic and static simulation area space without presenting error detection estimation.

6.7. Algorithm Highlight

Research and surveys in this paper propose an algorithm to make the robot more convenient and safe for tracking throughout the entire process of free movement. However, the heuristic algorithms applied and the analysis of the process of the search strategy use specific intelligent knowledge informed by a heuristic search technique, such as best-first search and depth-first search, including hierarchical trees combined with multiple greedy nodes; this provides better results, as shown in Figure 17. The figure presents simulated data for 1000 static and dynamic objects shown in the simulation, in the coverage area of the virtual target in the simulation of object detection recognition.

- The RL agent tries to learn the optimal value and policy functions.
- DNN-based function approximation is used to approximate the value and policy.

7. Conclusion

This paper presents a new decision support system, with in-depth analysis, which can provide insights for better exploration of a brand's image pattern recognition in multiple dimensions and improvement. This includes the possibility of considering robot movement. Our contribution suggests a new algorithmic framework by integrating the framework into the software for better robot interaction performance. We present this with case studies of robots in different situations in the simulation area, including static and dynamic objects, in order to find better evaluation and improvement. In addition, the results of the survey in this paper include path planning in an unknown environment; a new sensor-based algorithm was proposed. Another advantage of this approach is the ability to find short paths. The robot is equipped with a range sensor with 360˚ finite direction that obtains information from its surroundings by using high-tech mentoring navigation and location coordinates. Then, the robot will decide to select the next step depending on the future velocity vector of the obstacle and priority constraints. The robot departs from its current point along the original path trajectory while encountering obstacles on the shortest path. The survey recommends future approaches produced by the algorithm for robotic environmental reaction, avoidance, and object recognition through directional processing. Our contribution in this study is also extended by designing, in the next step of this work, a virtual robot that should handle complex environmental planners, for example, to avoid smart obstacles by creating steps to determine and select the best alternative solution's position and handle the status and position of each obstacle.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Marder-Eppstein, E., Berger, E., Foote, T., Gerkey, B. and Konolige, K. (2010) The Office Marathon: Robust Navigation in an Indoor Office Environment. 2010 *IEEE International Conference on Robotics and Automation*, Anchorage, 3-7 May 2010, 300-307. https://doi.org/10.1109/robot.2010.5509725

[2] Shyalika, C., Silva, T. and Karunananda, A. (2020) Reinforcement Learning in Dynamic Task Scheduling: A Review. *SN Computer Science*, **1**, Article No. 306. https://doi.org/10.1007/s42979-020-00326-5

[3] Haddad, T.A., Hedjazi, D. and Aouag, S. (2022) A Deep Reinforcement Learning-Based Cooperative Approach for Multi-Intersection Traffic Signal Control. *Engineering Applications of Artificial Intelligence*, **114**, Article ID: 105019. https://doi.org/10.1016/j.engappai.2022.105019

[4] García, J. and Shafie, D. (2020) Teaching a Humanoid Robot to Walk Faster through Safe Reinforcement Learning. *Engineering Applications of Artificial Intelligence*, **88**, Article ID: 103360. https://doi.org/10.1016/j.engappai.2019.103360

[5] Vázquez-Canteli, J.R. and Nagy, Z. (2019) Reinforcement Learning for Demand Response: A Review of Algorithms and Modeling Techniques. *Applied Energy*, **235**, 1072-1089. https://doi.org/10.1016/j.apenergy.2018.11.002

[6] Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R.H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., *et al.* (2019) Model-Based Reinforcement Learning for Atari. arXiv: 1903.00374.

[7] Szczepanski, R., Erwinski, K., Tejer, M., Bereit, A. and Tarczewski, T. (2022) Optimal Scheduling for Palletizing Task Using Robotic Arm and Artificial Bee Colony Algorithm. *Engineering Applications of Artificial Intelligence*, **113**, Article ID: 104976. https://doi.org/10.1016/j.engappai.2022.104976

[8] Kastner, L., Cox, J., Buiyan, T. and Lambrecht, J. (2022) All-in-One: A DRL-Based Control Switch Combining State-Of-The-Art Navigation Planners. 2022 *International Conference on Robotics and Automation* (*ICRA*), Philadelphia, 23-27 May 2022, 2861-2867. https://doi.org/10.1109/icra46639.2022.9811797

[9] Qian, K., Ma, X., Dai, X. and Fang, F. (2010) Socially Acceptable Pre-Collision Safety Strategies for Human-Compliant Navigation of Service Robots. *Advanced Robotics*, **24**, 1813-1840. https://doi.org/10.1163/016918610x527176

[10] Kastner, L., Buiyan, T., Jiao, L., Le, T.A., Zhao, X., Shen, Z., *et al.* (2021) Arena-Rosnav: Towards Deployment of Deep-Reinforcement-Learning-Based Obstacle Avoidance into Conventional Autonomous Navigation Systems. 2021 *IEEE*/*RSJ International Conference on Intelligent Robots and Systems* (*IROS*), Prague, 27 September-1 October 22021, 6456-6463. https://doi.org/10.1109/iros51168.2021.9636226

[11] Dugas, D., Nieto, J., Siegwart, R. and Chung, J.J. (2021) NavRep: Unsupervised Representations for Reinforcement Learning of Robot Navigation in Dynamic Human Environments. 2021 *IEEE International Conference on Robotics and Automation* (*ICRA*), Xi'an, 30 May-5 June 2021, 7829-7835. https://doi.org/10.1109/icra48506.2021.9560951

[12] Chen, C., Liu, Y., Kreiss, S. and Alahi, A. (2019) Crowd-Robot Interaction: Crowd-Aware Robot Navigation with Attention-Based Deep Reinforcement Learning. 2019 *International Conference on Robotics and Automation* (*ICRA*), Montreal, 20-24 May 2019, 6015-6022. https://doi.org/10.1109/icra.2019.8794134

[13] Xu, W., Cai, Y., He, D., Lin, J. and Zhang, F. (2022) FAST-LIO2: Fast Direct Lidar-Inertial Odometry. *IEEE Transactions on Robotics*, **38**, 2053-2073. https://doi.org/10.1109/tro.2022.3141876

[14] Marques, H.G., Jantsch, M., Wittmeier, S., Holland, O., Alessandro, C., Diamond, A., *et al.* (2010) ECCE1: The First of a Series of Anthropomimetic Musculoskeletal Upper Torsos. 2010 10*th IEEE-RAS International Conference on Humanoid Robots*, Nashville, 6-8 December 2010, 391-396. https://doi.org/10.1109/ichr.2010.5686344

[15] Osada, M., Izawa, T., Urata, J., Nakanishi, Y., Okada, K. and Inaba, M. (2011) Approach of "Planar Muscle" Suitable for Musculoskeletal Humanoids, Especially for Their Body Trunk with Spine Having Multiple Vertebral. 2011 11*th IEEE-RAS International Conference on Humanoid Robots*, Bled, 26-28 October 2011, 358-363. https://doi.org/10.1109/humanoids.2011.6100851

[16] Baduge, S.K., Thilakarathna, S., Perera, J.S., Arashpour, M., Sharafi, P., Teodosio, B., *et al.* (2022) Artificial Intelligence and Smart Vision for Building and Construction 4.0: Machine and Deep Learning Methods and Applications. *Automation in Construction*, **141**, Article ID: 104440. https://doi.org/10.1016/j.autcon.2022.104440

[17] Dalzochio, J., Kunst, R., Pignaton, E., Binotto, A., Sanyal, S., Favilla, J., *et al.* (2020) Machine Learning and Reasoning for Predictive Maintenance in Industry 4.0: Current Status and Challenges. *Computers in Industry*, **123**, Article ID: 103298. https://doi.org/10.1016/j.compind.2020.103298

[18] Abdar, M., Khosravi, A., Islam, S.M.S., Acharya, U.R. and Vasilakos, A.V. (2022) The Need for Quantification of Uncertainty in Artificial Intelligence for Clinical Data Analysis: Increasing the Level of Trust in the Decision-Making Process. *IEEE Systems*, *Man*, *and Cybernetics Magazine*, **8**, 28-40. https://doi.org/10.1109/msmc.2022.3150144

[19] Begoli, E., Bhattacharya, T. and Kusnezov, D. (2019) The Need for Uncertainty Quantification in Machine-Assisted Medical Decision Making. *Nature Machine Intelligence*, **1**, 20-23. https://doi.org/10.1038/s42256-018-0004-1

[20] Schmidhuber, J. (2015) Deep Learning in Neural Networks: An Overview. *Neural Networks*, **61**, 85-117. https://doi.org/10.1016/j.neunet.2014.09.003

[21] Dimiduk, D.M., Holm, E.A. and Niezgoda, S.R. (2018) Perspectives on the Impact of Machine Learning, Deep Learning, and Artificial Intelligence on Materials, Processes, and Structures Engineering. *Integrating Materials and Manufacturing Innovation*, **7**, 157-172. https://doi.org/10.1007/s40192-018-0117-8

[22] Zohuri, B., Moghaddam, M. and Mossavar-Rahmani, F. (2022) Business Resilience System Integrated Artificial Intelligence System. *International Journal of Theoretical & Computational Physics*, **3**, 1-7.

[23] Arents, J. and Greitans, M. (2022) Smart Industrial Robot Control Trends, Challenges and Opportunities within Manufacturing. *Applied Sciences*, **12**, Article 937. https://doi.org/10.3390/app12020937

[24] Linaza, M.T., Posada, J., Bund, J., Eisert, P., Quartulli, M., Döllner, J., *et al.* (2021) Data-Driven Artificial Intelligence Applications for Sustainable Precision Agriculture. *Agronomy*, **11**, Article 1227. https://doi.org/10.3390/agronomy11061227

[25] Garg, P.K. (2021) Overview of Artificial Intelligence. In: Sharma, L. and Garg, P.K., Eds., *Artificial Intelligence*, Chapman and Hall/CRC, 3-18. https://doi.org/10.1201/9781003140351-2

[26] Kapeller, A., Felzmann, H., Fosch-Villaronga, E. and Hughes, A. (2020) A Taxonomy of Ethical, Legal and Social Implications of Wearable Robots: An Expert Perspective. *Science and Engineering Ethics*, **26**, 3229-3247. https://doi.org/10.1007/s11948-020-00268-4

[27] Huang, Z., Shen, Y., Li, J., Fey, M. and Brecher, C. (2021) A Survey on AI-Driven Digital Twins in Industry 4.0: Smart Manufacturing and Advanced Robotics. *Sensors*, **21**, Article 6340. https://doi.org/10.3390/s21196340

[28] Sundaram, S. and Zeid, A. (2023) Artificial Intelligence-Based Smart Quality Inspection for Manufacturing. *Micromachines*, **14**, Article 570. https://doi.org/10.3390/mi14030570

[29] Redmon, J. and Farhadi, A. (2018) YOLOv3: An Incremental Improvement. arXiv: 1804.02767

[30] Wu, Z., Wang, Q., Jin, J. and Shen, Y. (2017) Structure Tensor Total Variation-Regularized Weighted Nuclear Norm Minimization for Hyperspectral Image Mixed Denoising. *Signal Processing*, **131**, 202-219. https://doi.org/10.1016/j.sigpro.2016.07.031

[31] Sprechmann, P., Bronstein, A.M. and Sapiro, G. (2015) Learning Efficient Sparse and Low Rank Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**, 1821-1833. https://doi.org/10.1109/tpami.2015.2392779

[32] Bai, Q., Li, S., Yang, J., Song, Q., Li, Z. and Zhang, X. (2020) Object Detection Recognition and Robot Grasping Based on Machine Learning: A Survey. *IEEE Access*, **8**, 181855-181879. https://doi.org/10.1109/access.2020.3028740

[33] Lee, W.J., Wu, H., Yun, H., Kim, H., Jun, M.B.G. and Sutherland, J.W. (2019) Predictive Maintenance of Machine Tool Systems Using Artificial Intelligence Techniques Applied to Machine Condition Data. *Procedia CIRP*, **80**, 506-511. https://doi.org/10.1016/j.procir.2018.12.019

[34] Jiang, X., Satapathy, S.C., Yang, L., Wang, S. and Zhang, Y. (2020) A Survey on Arti-

ficial Intelligence in Chinese Sign Language Recognition. *Arabian Journal for Science and Engineering*, **45**, 9859-9894. https://doi.org/10.1007/s13369-020-04758-2

[35] Yan, R., Wang, S., Zhen, L. and Laporte, G. (2021) Emerging Approaches Applied to Maritime Transport Research: Past and Future. *Communications in Transportation Research*, **1**, Article ID: 100011. https://doi.org/10.1016/j.commtr.2021.100011

[36] Adeniran, A.A. and Onebunne, A.P. (2024) Leveraging Artificial Intelligence for Intellectual Property Compliance and Global Regulatory Adherence. *International Journal of Engineering Technology Research & Management*, **8**.

[37] Deng, L. (2014) A Tutorial Survey of Architectures, Algorithms, and Applications for Deep Learning. *APSIPA Transactions on Signal and Information Processing*, **3**, e2. https://doi.org/10.1017/atsip.2013.9

[38] Kawaharazuka, K., Miki, A., Bando, M., Okada, K. and Inaba, M. (2022) Dynamic Cloth Manipulation Considering Variable Stiffness and Material Change Using Deep Predictive Model with Parametric Bias. *Frontiers in Neurorobotics*, **16**, Article 890695. https://doi.org/10.3389/fnbot.2022.890695

[39] Chinag, C. and Ding, C. (2014) Robot Navigation in Dynamic Environments Using Fuzzy Logic and Trajectory Prediction Table. 2014 *International Conference on Fuzzy Theory and Its Applications* (*iFUZZY*2014), Kaohsiung, 26-28 November 2014, 99-104. https://doi.org/10.1109/ifuzzy.2014.7091240

[40] Islam, M.M., Lam, A., Fukuda, H., Kobayashi, Y. and Kuno, Y. (2019) A Person-Following Shopping Support Robot Based on Human Pose Skeleton Data and Lidar Sensor. In: Huang, D.S., Huang, Z.K. and Hussain, A., Eds., *Intelligent Computing Methodologies*, Springer, 9-19. https://doi.org/10.1007/978-3-030-26766-7_2

[41] Yan, K., Gao, J. and Li, Y. (2023) Deep Reinforcement Learning Based Mobile Robot Navigation Using Sensor Fusion. 2023 42*nd Chinese Control Conference* (*CCC*), Tianjin, 24-26 July 2023, 4125-4130. https://doi.org/10.23919/ccc58697.2023.10240555

[42] Chen, Y., Han, S., Chen, G., Yin, J., Wang, K.N. and Cao, J. (2023) A Deep Reinforcement Learning-Based Wireless Body Area Network Offloading Optimization Strategy for Healthcare Services. *Health Information Science and Systems*, **11**, Article No. 8. https://doi.org/10.1007/s13755-023-00212-3

[43] Wang, T., Shen, X., Obaidat, M.S., Liu, X. and Wan, S. (2022) Edge-Learning-Based Hierarchical Prefetching for Collaborative Information Streaming in Social IoT Systems. *IEEE Transactions on Computational Social Systems*, **9**, 302-312. https://doi.org/10.1109/tcss.2020.3041171

[44] Kawaharazuka, K., Miki, A., Bando, M., Suzuki, T., Ribayashi, Y., Toshimitsu, Y., *et al.* (2022) Hardware Design and Learning-Based Software Architecture of Musculoskeletal Wheeled Robot Musashi-W for Real-World Applications. 2022 *IEEE-RAS* 21*st International Conference on Humanoid Robots* (*Humanoids*), Ginowan, 28-30 November 2022, 413-419. https://doi.org/10.1109/humanoids53995.2022.10000123

[45] Kästner, L., Marxand, C. and Lambrech, J. (2020) Deep-Reinforcement-Learning-Based Semantic Navigation of Mobile. arXiv:2008.00516v1 [cs.RO].

[46] Le Nguyen, T. and Do, T.T.H. (2019) Artificial Intelligence in Healthcare: A New Technology Benefit for Both Patients and Doctors. 2019 *Portland International Conference on Management of Engineering and Technology* (*PICMET*), Portland, 25-29 August 2019, 1-15. https://doi.org/10.23919/picmet.2019.8893884

[47] El-Shamouty, M., Kleeberger, K., Lämmle, A. and Huber, M. (2019) Simulation-Driven Machine Learning for Robotics and Automation. *TM—Technisches Messen*, **86**, 673-684. https://doi.org/10.1515/teme-2019-0072

[48] Moussaïd, M., Perozo, N., Garnier, S., Helbing, D. and Theraulaz, G. (2010) The

Walking Behaviour of Pedestrian Social Groups and Its Impact on Crowd Dynamics. *PLOS ONE*, **5**, e10047. https://doi.org/10.1371/journal.pone.0010047

[49] Shakya, A.K., Pillai, G. and Chakrabarty, S. (2023) Reinforcement Learning Algorithms: A Brief Survey. *Expert Systems with Applications*, **231**, Article ID: 120495. https://doi.org/10.1016/j.eswa.2023.120495

[50] Sasaki, M., Suhaimi, M.S.A.B., Matsushita, K., Ito, S. and Rusydi, M.I. (2015) Robot Control System Based on Electrooculography and Electromyogram. *Journal of Computer and Communications*, **3**, 113-120. https://doi.org/10.4236/jcc.2015.311018

[51] Wang, Y., Huang, M., Zhu, x. and Zhao, L. (2016) Attention-Based LSTM for Aspect-Level Sentiment Classification. *Proceedings of the* 2016 *Conference on Empirical Methods in Natural Language Processing*, Austin, 1-5 November 2016, 606-615. https://doi.org/10.18653/v1/d16-1058

[52] Zhang, Y., Zhang, P. and Yan, Y. (2017) Attention-Based LSTM with Multi-Task Learning for Distant Speech Recognition. *InterSpeech* 2017, Stockholm, 20-24 August 2017, 3857-3861. https://doi.org/10.21437/interspeech.2017-805

[53] Latif, S., Rana, R., Younis, S., Qadir, J. and Epps, J. (2018) Transfer Learning for Improving Speech Emotion Classification Accuracy. *InterSpeech* 2018, Hyderabad, 2-6 September 2018, 257-261. https://doi.org/10.21437/interspeech.2018-1625

[54] Gu, B., Ge, R., Chen, Y., Luo, L. and Coatrieux, G. (2021) Automatic and Robust Object Detection in X-Ray Baggage Inspection Using Deep Convolutional Neural Networks. *IEEE Transactions on Industrial Electronics*, **68**, 10248-10257. https://doi.org/10.1109/tie.2020.3026285

[55] Pi, Y., Nath, N.D. and Behzadan, A.H. (2020) Convolutional Neural Networks for Object Detection in Aerial Imagery for Disaster Response and Recovery. *Advanced Engineering Informatics*, **43**, Article ID: 101009. https://doi.org/10.1016/j.aei.2019.101009

[56] Chang, W., Chen, L., Hsu, C., Lin, C. and Yang, T. (2019) A Deep Learning-Based Intelligent Medicine Recognition System for Chronic Patients. *IEEE Access*, **7**, 44441-44458. https://doi.org/10.1109/access.2019.2908843

[57] Masud, M., Muhammad, G., Alhumyani, H., Alshamrani, S.S., Cheikhrouhou, O., Ibrahim, S., *et al.* (2020) Deep Learning-Based Intelligent Face Recognition in IoT-Cloud Environment. *Computer Communications*, **152**, 215-222. https://doi.org/10.1016/j.comcom.2020.01.050

[58] Liu, C., Cao, Y., Luo, Y., Chen, G., Vokkarane, V., Yunsheng, M., *et al.* (2018) A New Deep Learning-Based Food Recognition System for Dietary Assessment on an Edge Computing Service Infrastructure. *IEEE Transactions on Services Computing*, **11**, 249-261. https://doi.org/10.1109/tsc.2017.2662008

[59] Kawde, P. and Verma, G.K. (2017) Deep Belief Network Based Affect Recognition from Physiological Signals. 2017 4*th IEEE Uttar Pradesh Section International Conference on Electrical*, *Computer and Electronics* (*UPCON*), Mathura, 26-28 October 2017, 587-592. https://doi.org/10.1109/upcon.2017.8251115

[60] Li, T.S., Kuo, P., Tsai, T. and Luan, P. (2019) CNN and LSTM Based Facial Expression Analysis Model for a Humanoid Robot. *IEEE Access*, **7**, 93998-94011. https://doi.org/10.1109/access.2019.2928364