Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.074429

ARTICLE

Intelligent Ridge Path Planning for Agriculture Robot Using Modified Q-Learning Algorithm

A. Sivasangari[1,*], V. J. K. Kishor Sonti[1], J. Cruz Antony[1], E. Murali[1], D. Deepa[1] and A. Happonen[2]

[1]Department of Computer Science & Engineering, Sathyabama Institute of Science & Technology, Jeppiaar University, Chennai, India

[2]School of Engineering Science, LUT University, Lappeenranta, Finland

*Corresponding Author: A. Sivasangari. Email: sivasangarikavya@gmail.com

Received: 10 October 2025; Accepted: 14 January 2026; Published: 09 April 2026

ABSTRACT: In the past two decades, Precision Agriculture has received research attention since the development of robotics. Agricultural robotic equipment and drones, which can be operated by farmers, are appearing more frequently and being used to make the process of farming easier and more productive. This paper attempts to develop a modified Q-learning algorithm. A reinforcement learning algorithm called Q-learning has Q-values that are updated in order to find the best routes for the robotic devices to follow while avoiding any obstacles. Different types of terrain and other factors that influence the development of good routes for the robotic devices are included in the experiments performed. Through an extensive set of experiments done with different types of terrain the researchers found that the modified Q-learning algorithm converges to the optimal path significantly quicker than the current benchmark Deep Q-Network (DQN) algorithms and that the average distance that the modified Q-learning algorithm travels to get to its destination over different terrane types was 28.7% shorter than the average distance traveled using the standard DQNs. The researchers also found that the modified Q-learning algorithm has been able to successfully avoid obstacles on 99.5% of all occasions tested. The shortest route to the destination is expected to take less time, and it demonstrates the benefit of using a robotic device that has the ability to detect and avoid obstacles in order to be effective on more difficult types of terrain.

KEYWORDS: Path planning; agricultural robotics; reinforcement learning; Q-learning

1 Introduction

Recent findings in the field of Machine Learning and robotics alike [1] suggest new paths to experiment and expand. The work by researchers in the development of agricultural robots, self-moving vehicles and modern digital solutions is making breakthroughs in precision agriculture, including self-moving heavy vehicles in industrial processing cites [2] and farms [3], drones for analyzing fields and forests [4,5], robot assisted systems [6], Internet of Things (IoT) technologies system integrations [7] and monitoring valuable resources like fresh water, digital agricultural solutions for sustainability, with ongoing and archived development on position data, algorithms [8] & software and GPS [9,10] system. At the same time, a global push towards more sustainable ways of living, producing products, assets and also eatables is becoming more and more demanded by societies and customers alike. Intelligent and multi-objective algorithms and procedures towards perfection on agricultural production encourage societies to keep ideating, experimenting, and making substantial progress on the digitalization of agriculture. By utilizing modern Artificial intelligence, data analysis and machine learning solutions, we can boost agricultural productivity and fight against food

Copyright © 2026 The Authors. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

waste, which improves our options on safeguarding global food value chains. The work presented here is one such manifestation of knowledge base enrichment using a modified Q-learning algorithm for path-finding of Agri-robots [11].

Right now, we live in the digital age transition in artificial/computational intelligence (AI/CI) intersects with human cognition and social interactions, trough out industries, education and everyday leisure and work life. Some of the literature investigated sociotechnical systems, AI models, and cognitive science to compare the decision-making processes of humans and machines. Based on the study, AI can be both beneficial and problematic in social settings and for society in large. In this context, when agronomists across the globe are looking for better methods for increasing yield and safe mechanisms for developing agricultural products for the environment, and when they lean on AI solutions, the benefits and impact both need to be weighed in. Technological advancements are complementing their efforts with the emergence of novel methods in computing. Even though substantial research can be found in these fields, the unification of ideas is leading to innovation. This work presents the fusion of such ideas that lead to increased efficacy of agricultural robots.

Q-Learning algorithms have exploration and exploitation as main components in reaching out to performance metrics such as path lengths, shortest time, stability, better convergence, and reduced overestimation. A comparative study of the Deep Queue Network (DQN) and the Dueling Double DQN (D3QN) and modified Q-learning algorithms was carried out to provide clear insights about the right application of these algorithms in the effective usage of agricultural robots. The randomness is associated with iteratively updating Q values, where the Bellman equation is the focal point and advantage of this method, whereas DQN and D3QN primarily depend on neural networks for such updates. The disadvantages of Queue learning and DQN are to a certain extent mitigated by the latter, whereas Q-learning finds the best way to achieve stability, convergence and optimal paths in the shortest time. This makes this algorithm a suitable contender for detecting and avoiding obstacles in rough terrains. The main contributions are as follows:

- A Novel Modified Q-Learning Algorithm: We propose a new variation of the Q-learning algorithm, specifically tailored for agricultural robotics, which enhances path planning performance under both static and dynamic environmental constraints.
- Integration of an Offline Expert System: We introduce a method to leverage a pre-trained expert system for initializing the reinforcement learning network, significantly accelerating the training process and improving the robustness of the agent's initial policy.
- Comprehensive Benchmarking and Quantitative Validation: We provide a rigorous empirical evaluation against state-of-the-art models (DQN, D3QN), demonstrating superior performance with quantifiable improvements, including a 28.7% faster convergence, a 16.4% reduction in path length, and near-perfect obstacle avoidance.
- A Framework for Robust Agricultural Navigation: The developed system offers a practical and efficient framework for autonomous navigation in challenging agricultural terrains, directly addressing key challenges in precision agriculture, such as operational efficiency and path stability.

This paper is organized as an introduction that indicates the motivation behind this study, a literature survey giving a robust record of previous works in this domain, and a section describing the application of Q-learning represents the mathematical and qualitative analysis of this method. The last section presents the results, interpretations and conclusion.

2 Literature Survey

Following chapter describes related previous research and findings in the domain of precision agriculture. A detailed review of modern research, innovative considerations and interpretations, plus the current knowledge and outcomes of applied studies, will be presented and explained for the reader.

A thorough analysis of machine learning (ML) applications in crop management in agriculture is given by Attri, Awasthi, and Sharma (2024). For crop monitoring, the mentioned study divides methods into three categories: sensor-based approaches, optimization algorithms, and predictive models. Using historical data, supervised learning is a crucial method for forecasting crop yields and disease outbreaks. The authors do also emphasize the application of deep learning models for automated plant health assessment in conjunction with remote sensing and image processing methods for precision agriculture. In order to optimize fertilization and irrigation schedules, reinforcement learning is also investigated. Ensemble approaches for enhancing prediction robustness and accuracy are covered too. By analyzing real-time data from sensors and IoT devices, machine learning models have been demonstrated to improve decision-making. Data availability is noted to be one of the issues and good data is a defined requirement for scalable machine learning [12].

Choudhury et al. (2018) proposed robot planning focuses on the optimization of task-specific objectives since the aim is to make efficiently tailored actions to meet the set goals. This process forms an integral part in the process of selecting collision-free paths during navigation tasks, giving an edge in the movement capabilities of the robot in its environment. Collision-free routes become a priority for robots, when navigating in complex spaces and avoiding hazards while improving their overall performance in application scenarios [13].

DQ-GAT is a framework for safe and effective autonomous driving that combines Deep Q-Learning with Graph Attention Networks (GATs), which is presented by Cai et al. (2022). Their approach uses dynamic graphs to model traffic situations, with vehicles acting as nodes and interactions as edges. By allocating attention weights, the GAT module allows the agent to selectively focus on important nearby vehicles. In complex, multi-agent environments, this improves decision-making and situational awareness. From these graph-based representations, optimal driving policies are learned using deep Q-learning. Safety restrictions are incorporated into the strategy to prevent collisions and guarantee smooth driving. When compared to baseline models, experiments demonstrate better performance in terms of efficiency and safety. The potential of graph-based reinforcement learning for practical autonomous driving tasks is demonstrated by DQ-GAT [14].

Using scene graphs as high-level semantic representations, Amiri, Chandan, and Zhang (2022) present a novel method for robot planning under partial observability. Building probabilistic scene graphs that represent objects, spatial relationships, and uncertainties in partially observable environments is the methodology's main task. In order to allow robots to infer hidden states, they incorporate graph neural networks (GNNs) for reasoning over the scene structure. The belief update mechanism, which uses observations to improve the scene graph over time, is a significant contribution. Monte Carlo Tree Search (MCTS) is applied to the updated belief space in the planning component. Instead of using low-level geometric reasoning, this method enables semantic-level decision-making. In scenes that are cluttered and obscured, the system does exhibits resilience [15].

In order to provide autonomous plant care, Correll et al. (2010) describe the design and implementation of an indoor robot gardening system that combines robotics and sensor networks. The system integrates mobile manipulation for plant watering and repositioning, vision-based plant health monitoring, and soil moisture sensing. The use of modular robotic platforms for scalability and flexibility is a key methodology.

In indoor settings, the robot uses map-based navigation and probabilistic localization. In order to guarantee responsive care, the study places a strong emphasis on closed-loop feedback between sensors and actuators. Control algorithms are adjusted to be resilient to changes in plant configurations and lighting. The system serves as an example of the application of service robotics to indoor, sustainable agriculture. The groundwork for future autonomous plant maintenance at home is laid by this work [16].

Chebotar et al. (2017) proposed method effectively combines model-based efficiency with model-free performance: it is specifically designed for time-varying linear-Gaussian policies. It integrates iterative fitting of linear-Gaussian dynamics within the model-based updates, along with path integral policy improvements for model-free updates. By using this dual methodology, deep neural networks can now be trained through guided policy search to achieve performance above that of traditional model-free approaches. At the same time, it maintains the sample efficiency property of model-based approaches, which makes this method especially effective in solving complex manipulation tasks with great effectiveness [17].

Stephen et al. (2011) discussed that weighted transition systems are used in optimal motion planning of robots under temporal logic constraints. This area of research converts LTL specifications into Buchi automata, thus making the processing easier. Optimal paths can be computed by using graph algorithms for specific applications like persistent monitoring and data gathering. The main challenge is to minimize the worst-case time between optimizations of propositions. For efficiency enhancement, a new approach that not only outperforms the previous mixed integer linear programming model but also provides an alternate solution to the constrained S-bottleneck problem by promoting efficient routing in systems used [18].

Leonardo et al. (2023) developed an elaborate methodology for choosing the right navigation strategies for agriculture. The method takes data from GNSS, LiDAR, and camera sensors in one single application to be used. Such a solution will adequately fill in the loopholes existing with such sensors and help to offer reliable autonomous navigation with as high as 87% effectiveness in selecting sensors. The characteristics of GNSS features change gradually during crop changes; therefore, the performance and cost of various classification algorithms were compared. Optimization of agriculture-related processes is done to reduce losses and increase efficiency by this methodology, but agricultural robots or AgBots are primarily used for data collection and navigation to enhance farming practices [19].

Kian et al. (2009) proposed an approach that models the environmental structure to support high-quality mapping, using a log-Gaussian process to address map uncertainty. The reformulation of the classical MASP into a reward-maximizing variant known as iMASP has time complexity independent of map resolution and, therefore, improves efficiency. Furthermore, this method decreases sensitivity to larger robot team sizes and provides theoretical bounds for adaptive policies. Empirical evidence shows iMASP to perform better than competitors, while simultaneously imposing specific conditions, and as result, no significant advantage is offered [20].

Diogo et al. (2020) discussed the convergence of Q-learning when using linear function approximation, providing a two-time-scale modification of the Q-learning update. The faster time scale mimics the DQN-type update for bootstrapping, whereas the slower time scale behaves as a modified target network update. The work guarantees convergence with less stringent assumptions than usual. It provides performance bounds for the limit solution and proves convergence in domains where standard Q-learning tends to diverge. In addition, it extends the analysis to control settings in reinforcement learning with function approximation and demonstrates the robustness and efficiency of the approach [21].

Georges et al. (2013) proposed methodology is important for having probabilistically safe planning so that user-defined minimum probability constraints are addressed, thereby bringing about the level of robustness demanded against uncertainty in dynamic obstacle motion patterns. The methodology will also

accurately represent the behavior of future dynamic obstacles and assist in improving the navigation system's reliability. An innovative idea in this field would be the introduction of a concept of the Robustness-Robust Gaussian Process or RR-GP to depict the considerable improvement in the prediction of the trajectory. This method ensures probabilistic feasibility in real-time by combining RR-GP with a chance-constrained Rapidly-Exploring Random Tree for path planning. Long-term prediction capabilities include clustering-based techniques, and it has been proven safe to navigate within dense environments, hence significantly reducing the likelihood of collision [22].

Golowich, Moitra, and Rohatgi (2023) use filter stability—the property where belief states converge irrespective of initial conditions—to propose a novel framework for planning and learning in POMDPs. They demonstrate how the agent can efficiently "forget" its initial uncertainty thanks to strong filter stability, making belief updates easier. Stability-aware planning algorithms that take advantage of this convergence for effective policy learning are presented in the paper. Contractively metrics and mixing conditions on the observation and transition kernels are important methodological tools. Their method combines insights from control theory with learning theory. Under stability assumptions, the authors offer verifiable performance guarantees. They show that sample-efficient reinforcement learning techniques can be used to solve filter-stable POMDPs. This provides a fresh approach to tractable learning in partially observable high-dimensional systems [23].

Yanlin et al. (2019) used IPOMDP-net as the next big step in partial observability-based multi-agent planning since this approach merges model-free learning and model-based techniques for planning to yield better-than-state-of-the-art performance even for state-of-the-art model-free networks across a host of tasks. The network exhibits even better generalizability across larger environments, which are unseen so far. This architecture would integrate the I-POMDP model along with the QMDP algorithm within a neural framework, which could be trained in the setting of reinforcement learning but now with observable reinforcements. Also, the results indicate that modelling other agents contribute positively to the quality of decision-making. In general, IPOMDP-net provides a comprehensive neural computing architecture for multi-agent planning challenges [24].

Pamuklu et al. (2022) proposed that aerial base stations improve the processing of tasks at smart farms through the integration of leading-edge technologies, such as IoT sensors, which allow monitoring from a distance over wide agricultural areas. Risk-sensitive reinforcement learning optimizes scheduling for these aerial base stations with algorithms designed to pay attention to energy efficiency and on-time delivery. Further, mixed integer linear programming performs lower bounds and provides a firm framework to manage tasks. Simulations prove that this method proposed here outperforms traditional Q-Learning and works well in the reduction of deadline violations on IoT-related tasks. These will be helpful in increased hovering times for aerial BSs that, in turn, can increase the general efficiency in the task management carried out on smart farming operations [25].

Xiong et al. (2023) proposed that an improvement in the path planning algorithm has led to various developments in the navigation procedure of an AGV. The most significant development has taken place through improved child node selection and a better heuristic function that allows more efficient computation of the global path to reduce redundant nodes. Besides that, the employment of B-spline curves has brought about smoother paths, and local path planning follows the global path contour more closely. These advances prevent local optimality from occurring in path planning such that the AGVs dynamically navigate around obstacles with fewer difficulties. These innovations then resulted in a reduction of 3.6% in the path length, and path time was cut down by 6.7% [26]. In general, reinforced learning is quite a good option to choose, when a device needs to cover certain/given physical area and do complete coverage over the given area.

Abdallah et al. (2016) introduced the Repeated Update Q-learning algorithm, which successfully overcomes the policy bias problem of the traditional Q-learning algorithm. RUQL retains all the simplicity and effectiveness with which Q-learning is renowned while showing better performance than traditional Q-learning in noisy and non-stationary settings. Importantly, it retains the convergence guarantees Q-learning provides in stationary settings but enhances the learning dynamics to be more suitable for the non-stationary setting. The theoretical and experimental analysis confirms the result and hence the efficiency of the algorithm. In addition, the RUQL algorithm is computationally feasible with a closed-form derivation to be applied practically [27].

From the above literature, Deep Q-Network (DQN) and Deep Q-Network with 3-Step Lookahead (DQ3N) differ from each other concerning the Q-value estimation and environmental learning. DQN utilizes the epsilon-greedy strategy, which makes it less efficient and makes it converge at a slower speed and unstable, especially in situations involving delayed rewards. DQN is less complicated and occupies fewer resources, but it may sometimes fail in certain situations. DQ3N models the DQN along with the multi-step lookahead. This would bring about a learning speed and sample efficiency but is said to have an overestimation bias. Its utilization would imply the use of more computational power. It may also achieve some little amount of instability through very careful tuning of its parameters. In general, DQ3N often outweighs DQN, given demanding resources and additional implementation effort. Q-learning's utility in forecasting crop yields is inhibited by its weaknesses, such as data quality, mapping complexity, parameter sensitivity, and substantial overhead computation. Taking care of these shortcomings would be the key to realizing a true and dependable agricultural forecast. Hence, we proposed a modified Q-learning algorithm which will enhance the robot path-learning [28].

Yin and Xiang (2025) propose a dynamic adaptive priority guidance (DAPG) architecture that modifies navigation priorities based on sensor integrity and encounter context in order to address the problem of collision avoidance for unmanned surface vehicles (USVs) operating in perception-limited situations. Previous research includes learning-based approaches (DRL, imitation learning) that provide flexibility but have high data requirements, safety issues, and poor interpretability, as well as traditional rule-based and model-based planners (VO, DWA, MPC) that perform well under dependable sensing but deteriorate when perception is incomplete. Although many current approaches rely on static decision rules or assume full observability, research on multi-sensor fusion and context-aware switching strategies demonstrates that integrating sensing modalities might improve robustness. In between these two extremes, Yin and Xiang's contribution offers an interpretable, adaptive mechanism that dynamically reweights navigation techniques to deal with complicated maritime interactions, occlusions, and impaired sensing. Nevertheless, the literature and the research itself draw attention to some shortcomings, including the need for more extensive real-world validation, delayed switching responses, and possible misclassification of sensor dependability.

The efficiency and intelligence of agricultural multi-robot systems have been the subject of an increasing amount of research, with current studies focusing on autonomous decision-making, cooperative control, and work distribution. Real-time limitations, varied robot capabilities, and changeable farm environments are common challenges for traditional allocation techniques like heuristic rules and optimization algorithms. Because it can learn adaptive policies and optimize performance through interaction with the environment, reinforcement learning (RL) has become a promising solution to these problems. Although RL has been investigated in the past for path planning, crop monitoring, and cooperative harvesting, many solutions still struggle with coordination and scalability in large robot clusters. By offering an RL-based optimization framework created especially for job allocation in agricultural multi-robot clusters, the study by Lu et al. (2024) [29] closes these gaps and advances the field by facilitating more adaptable, effective, and reliable collaboration under challenging field settings.

Global path planning approaches have been thoroughly studied in UAV navigation research [5,30] to guarantee safe, effective, priority optimization and mission-driven flight in challenging situations. Although studies have produced structured solutions, traditional algorithms like A*, Dijkstra's, RRT, and potential field approaches frequently struggle with dynamic impediments, multi-objective priorities, and adaptability to unknown settings. The ability of reinforcement learning approaches, especially Q-learning, to learn optimal navigation policies through environmental interaction has gained prominence as UAV missions increasingly demand flexible decision-making—balancing factors like safety, energy consumption, time efficiency, and mission priorities. Although Q-learning has been shown to be effective in obstacle avoidance and path optimizations in earlier research, typical issues still exist, such as convergence speed, scalability to large state spaces, and difficulty integrating many weighted objectives. In order to close these gaps and advance intelligent autonomous navigation for practical UAV applications, the study by de Carvalho et al. (2025) [5] presents a Q-learning-based global path planning framework that integrates pondered priorities. This allows UAVs to navigate while simultaneously optimizing multiple mission constraints.

Although many meta-heuristic and bio-inspired algorithms exist for path planning, Q-learning remains significant because it is a model-free method that learns optimal behavior directly from interaction with the environment. It adapts online as conditions change, making it suitable for dynamic scenarios where heuristic methods may need re-optimization. Q-learning also provides theoretical convergence guarantees, offering reliability that heuristic algorithms may lack. Its learning process avoids dependence on initial populations or randomization, reducing the chance of getting trapped in local optima. After training, the policy can be executed quickly with minimal computation, supporting real-time deployment. Furthermore, its reward-based structure makes it straightforward to encode constraints such as obstacle avoidance or energy cost. Overall, Q-learning combines adaptability, theoretical soundness, and practical efficiency, which motivates its use alongside existing heuristic methods.

In certain studies, a model is suitable only for a specific condition, and it is ideal for a specific type of dataset. Since the learned policy operates on partial data, hence result may also be partial. In a certain study, it is suitable only for certain plants. In our proposed work will provide better convergence, stability and finding the path optimization obtained through modified Q-Learning approach.

3 Q-Learning-Based Path Finding

The yield of crops can be enhanced by precision agriculture, which helps to monitor the crop field dynamically, taking account robots perception [31] and precision agriculture requirements, and correlate the goal, task and used algorithms, like the part finding one. The different technologies used in precision agriculture are robots, sensors, drones [32] and Global Positioning System (GPS) [33] and other high precision Global Navigation Satellite System GNSS based solutions [34], for Multi-GNSS [35] precise point positioning (PPP) in precision agriculture (PA). Modern agricultural technology has alleviated challenges in the industry by addressing the high cost of supplies, labor shortages, and issues related to climate change, irrigation, and soil erosion, with modern solutions for sensor technologies [36] and protection of workers and environment against, e.g., landslide [37] and other global warming related extreme weather event-based challenges in agriculture and environment [38].

The most efficient way to monitor the agriculture field is the usage of autonomous robots, which help in monitoring crop health, collection of data and sharing of data to farmers. The proposed approach makes effective planning in the agriculture field and also obstacle avoidance using a Q-learning algorithm. It is a reinforcement learning algorithm, which enables an optimal way of interacting with an environment. Q-learning aims to determine the best strategy for choosing actions, guiding the agent on what to do in various

situations to achieve the highest possible long-term reward. The important aspect is to continuously monitor of field following the variability of field area [39].

The selection of a Reinforcement Learning (RL) framework for agricultural robot navigation was driven by the critical need for adaptive intelligence in unpredictable farm environments. While traditional path planning algorithms like DQN or D3QN are computationally efficient for static maps, they lack the cognitive flexibility to handle dynamic obstacles such as moving farm equipment or personnel—and changing terrain conditions, a common limitation known as the "brittleness" of classical planners.

Our proposed modified Q-learning approach directly addresses these limitations. Its primary advantage lies in its model-free nature, allowing the robot to learn optimal policies through direct interaction with the environment without requiring a perfect, pre-programmed world model. This capability is paramount for real-world agriculture where conditions are constantly in flux. However, a well-documented issue with vanilla Q-learning is its sample inefficiency and slow convergence in complex state spaces, which can render it impractical for real-world deployment.

To overcome these inherent disadvantages, our contribution integrates an offline expert system. This hybrid architecture mitigates the problem of slow initial learning by providing a robust foundational policy, thus guiding the exploration process and dramatically accelerating convergence. Consequently, our method synthesizes the reliability of a rule-based system with the adaptive superiority of RL, resulting in a solution that is not only more sample-efficient than deep RL variants like DQN but also significantly more robust and adaptable than any non-learning-based alternative. This makes it uniquely suited for the specific challenges of precision agriculture.

3.1 Agriculture Robot Coverage Area

The task of agricultural path coverage by an autonomous robot involves traversing the entire agricultural field. This coverage is represented by an undirected graph, where the set of states (S) corresponds to various locations within the field, and the set of edges (E) signifies the possible movements between these states. Certain edges are impassable due to obstacles. The time required to visit each grid cell is denoted by Mt. Agricultural fields are inherently non-homogeneous, exhibiting variations in physical attributes such as size, shape, area, volume, porosity, color, and appearance, as well as in chemical properties like fertility levels, organic matter content, pH value, and resistance. Consequently, not all areas of the field carry equal significance. To account for this, each cell is assigned a value determined by multiplying its characteristics by a weight vector that reflects its relative importance [40–42].

3.2 Reinforcement Learning—Q-Learning

The Q-Learning algorithm enables an agent to effectively operate and interact with its environment. The primary objective of Q-learning is to guide the agent in selecting the optimal actions under various conditions to maximize rewards within a given timeframe. Fig. 1 illustrates the framework of agent-environment interaction using reinforcement learning. The Q-learning equation, which is used to calculate the value of a particular state, is as follows:

$$Q(s,a) = Q(s,a) + \alpha * [r + \gamma * \max(Q(s',a')) - Q(s,a)] \quad (1)$$

In this context, $Q(s,a)$ represents the expected reward for taking a specific action in a given state, while r denotes the actual reward received for that action. The next state is indicated by s', α stands for the learning rate, and γ represents the discount factor. The term $\max(Q(s',a'))$ refers to the maximum Q-value for the subsequent state. The agent selects an action in the current state based on a policy derived from these

Q-values. Subsequently, the agent transitions to a new state, receives a reward r, and updates the Q-value for the state-action pair (s,a) using the Q-learning update equation [1]. The current state of the agent is denoted as current. If the agent intends to move forward, Eq. (2) represents this action.

$$Q(s_{current}, a_{forward}) \leftarrow Q(s_{current}, a_{forward}) + \alpha \left[r + \gamma a' max Q(s', a') - Q(s_{current}, a_{forward})\right] \quad (2)$$

Conversely, if the agent decides to perform a backward action, it is represented by Eq. (3)

$$Q(s_{current}, a_{backward}) \leftarrow Q(s_{current}, a_{backward}) + \alpha \left[r + \gamma a' \max Q(s', a') - Q(s_{current}, a_{backward})\right] \quad (3)$$

If the agent wants to move left or right then it is represented by Eqs. (4) and (5), respectively

$$Q(s_{current}, a_{left}) \leftarrow Q(s_{current}, a_{left}) + \alpha \left[r + \gamma a' \max Q(s_{left}, a') - Q(s_{current}, a_{left})\right], \quad (4)$$

$$Q(s_{current}, a_{right}) \leftarrow Q(s_{current}, a_{right}) + \alpha \left[r + \gamma a' \max Q(s_{right}, a') - Q(s_{current}, a_{right})\right] \quad (5)$$

Based on Eqs. (1)–(4), the agent can navigate forward, backwards, left, and right by utilizing the rewards received to inform its movements.

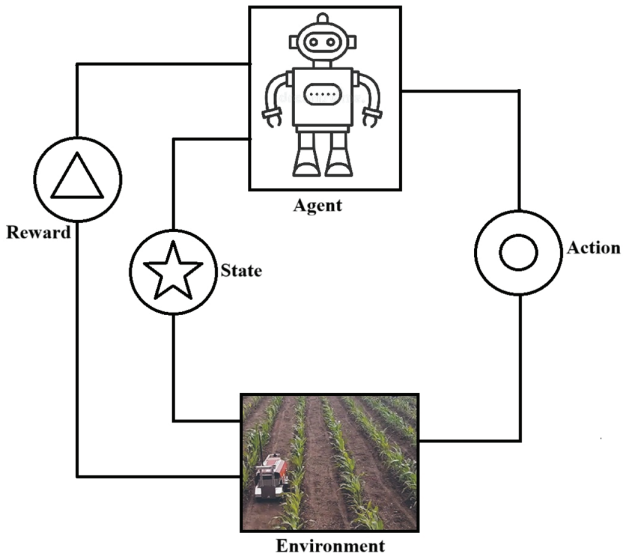

Figure 1: Agent & environment interaction—reinforcement learning framework.

In this context, $s_{current}$ represents the current state, a_{left} indicates the action of moving left, a_{right} signifies the action of moving right, $a_{forward}$ denotes the action of moving forward, and $a_{backward}$ refers to the action of moving backwards. The variable r stands for the reward received, α is the learning rate, and γ represents the discount factor. Additionally, s_{left} and s_{right} denote the new states after moving left and right, respectively, while s′ indicates the new state after taking action a.

Eq. (5) represents the scenario where the agent encounters an obstacle while moving from the current state $s_{current}$ in the direction of action a, and is given as follows

$$Q(s_{current}, a) \leftarrow Q(s_{current}, a) + \alpha \left[r_{obstacle} + \gamma a' \max Q(s_{current}, a') - Q(s_{current}, a)\right] \quad (6)$$

In Eq. (6), $r_{obstacle}$ denotes the penalty incurred for agent to encounter an obstacle in the current state. To deter the repeating of the same action in the future, a negative reward $r_{obstacle}$ must be applied when the agent encounters an obstacle and it is given as Eqs. (7) and (8)

$$r_{obstacle} = -1, \tag{7}$$

$$r(s,a) = \begin{cases} Positive\ reward, & if\ encountered\ with\ no\ obstacle, \\ Negative\ reward, & if\ encountered\ with\ obstacle. \end{cases} \tag{8}$$

The agent receives a maximum reward if the goal is achieved and a minimum reward if the goal is not achieved and it is represented by Eq. (9)

$$r(s,a) = \begin{cases} 100, & maximum\ reward\ for\ achieved\ goal, \\ 0, & minimum\ reward\ for\ not\ achieved\ goal. \end{cases} \tag{9}$$

The reward equation is given by

$$Q(s,a) \leftarrow Q(s,a) + \alpha\ [(100) + \gamma\ max a'\ Q(s',a') - Q(s,a)] \tag{10}$$

The obstacle can be avoided by the agent using Eq. (11), for which the reward is adjusted

$$Q(s,a) \leftarrow Q(s,a) + \alpha\ [(r - r_{obstacle}) + \gamma\ max a'\ Q(s',a') - Q(s,a)], \tag{11}$$

$r_{obstacle}$ is applied when an obstacle is encountered

If an agent receives a reward for the action, then it is represented by the Eq. (12) for the reward r(s,a)

$$Q(s,a) \leftarrow Q(s,a) + \alpha\ [r(s,a) + \gamma a' \max Q(s',a') - Q(s,a)]. \tag{12}$$

If the agent encounters a collision with an obstacle, it is represented by the following Eq. (13)

$$r(s,a) = r_{collision}. \tag{13}$$

If the agent makes a successful move and with no obstacle collision, then it is given by Eq. (14)

$$r(s,a) = r_{success} \tag{14}$$

3.3 Reinforcement Learning—Modified Q-Learning

By adding domain-specific modifications to rewards, state representations, and exploration tactics, modified Q-learning shown in Eq. (15) is an adaption of the conventional Q-learning algorithm designed to customised real-world applications

$$Q(s,a) \leftarrow Q(s,a) + \alpha\ [r + \gamma f(a' max\ Q\ (s',a')\ .(1-P)) - Q(s,a)], \tag{15}$$

f represent the adjust of penalties and priorities. Here the penalties represent the collision, large energy consumption, moving away from the goal. The reward function f of the robot is represented as follows, for instances +10 if the forward movement reduces the goal distance, −5, if the robot makes unnecessary move which increases the distance of goal, −20 if the robot make collision with obstacles, 0 if the robot does not make any movement.

In order to discourage the collision state in the robotic path planning, the penalty is denoted by

$$P(s,a) = High\ Value, \tag{16}$$

which may range from 10 to 100

Penalty for revisited place is denoted by,

$$P(s,a) = \lambda.visit\ count\ of\ s, \tag{17}$$

where λ denotes the penalty weight

Penalty based on distance is denoted by,

$$P(s,a) = \kappa.distance(s,goal), \tag{18}$$

where κ denotes the penalty based on distance

3.4 Modified Q-Learning Algorithm

The algorithm for Q-Learning is given as follows

1. **Initialization:**
- Assign initial values to Q(s, a) for all possible state-action pairs.
- Set the learning rate (α), discount factor (γ), and exploration rate (ϵ).
2. **Repeat for each episode:**

A. Set the initial state s.

B. For every step within the episode:
- Exploration vs. Exploitation (ϵ-greedy policy):
  - With probability ϵ, select a random action a.
  - Otherwise, select the action a that has the highest Q-value:

    $a = argmax\left(Q\left(s,a\right)\right).$
- Execute action a, then observe the reward r and the resulting state s'.
- Update Q-value using the Q-learning equation:

  $Q\left(s,a\right) = Q\left(s,a\right) + \alpha * \left[r + \gamma f\left(a'maxQ\left(s',a'\right).\left(1-P\right)\right) - Q\left(s,a\right)\right].$
- Set the current state s to the new state s'.
- If the new state s' is terminal, end the episode.
- f represents the penalties such as collision, large energy consumption, moving away from the goal

C. Optionally decrease the exploration rate (ϵ) after each episode.

3. **End Algorithm** [43].

3.5 Modified Q-Learning Algorithm

In a self-driving car simulator, Modified Q-learning enables an autonomous vehicle to learn decision-making independently. The primary objective of the self-driving car simulator is to safely navigate roads without collisions while adhering to traffic rules to maximize rewards. At each state, the simulator must decide on actions such as accelerating, braking, turning left or right, moving forward or backward, and adjusting speed based on environmental conditions.

The Modified Q-Learning framework consists of the following

A. State

- Car position
- Car speed
- Signal status
- Other vehicle distance

B. Action

- Brake
- Left turn
- Right turn
- Backward movement
- Front movement
- Speed of the car

C. Distance-Based Reward

- Reward [0, 100] for reaching the distance based goal, shortest distance is maximum reward [44].

D. Dynamic Penalties

- +10 if the forward movement reduces the goal distance
- −5, if the robot revisits the path which increases the distance of the goal
- −10 if the robot makes a collision with obstacles
- [−100, 0], if the robot makes an unnecessary move, the penalty is based on the distance
- 0 if the robot does not make any movement [45]

E. Learning Parameters

- Alpha indicates the new information learned by the agent which gets updated despite old information
- Gamma indicates the future rewards
- Epsilon denotes the new information learned by the agent [46].

The primary goal is to train the robot to discover the path with great efficiency, which supports the future idea of having machinery and vehicles operate and move autonomously within an area, where people also move around. In this case the primary goal is used to focus on basic environment, which includes seedable random layout creation and random dirt placement. Gym API integration with methods (step, observe, render, and reset), as well as a rudimentary sparse reward algorithm. Fig. 2 shows the self-driving car simulator, Agent interface and vectorized random agent class that can interact with their surroundings.

Our agent receives observations in 10 × 10 arrays, which are essentially images. To simplify learning, we'll divide the image into multiple channels:

Channel 1: Indicates an accessible path (1 for accessible, 0 for not).

Channel 2: Indicates an inaccessible path (1 for dirt, 0 for clean).

Channels 3–6: These four channels work together to show where the robot is and what direction it is facing. We enter a single 1 in the corresponding position and channel to reflect the robot's location and orientation. This arrangement also enables us to expand the amount of training data through augmentation. Rotating and flipping the image allows us to produce eight alternatives for every given room layout. We can train our model in these equivalent rooms without having to generate additional training data [47].

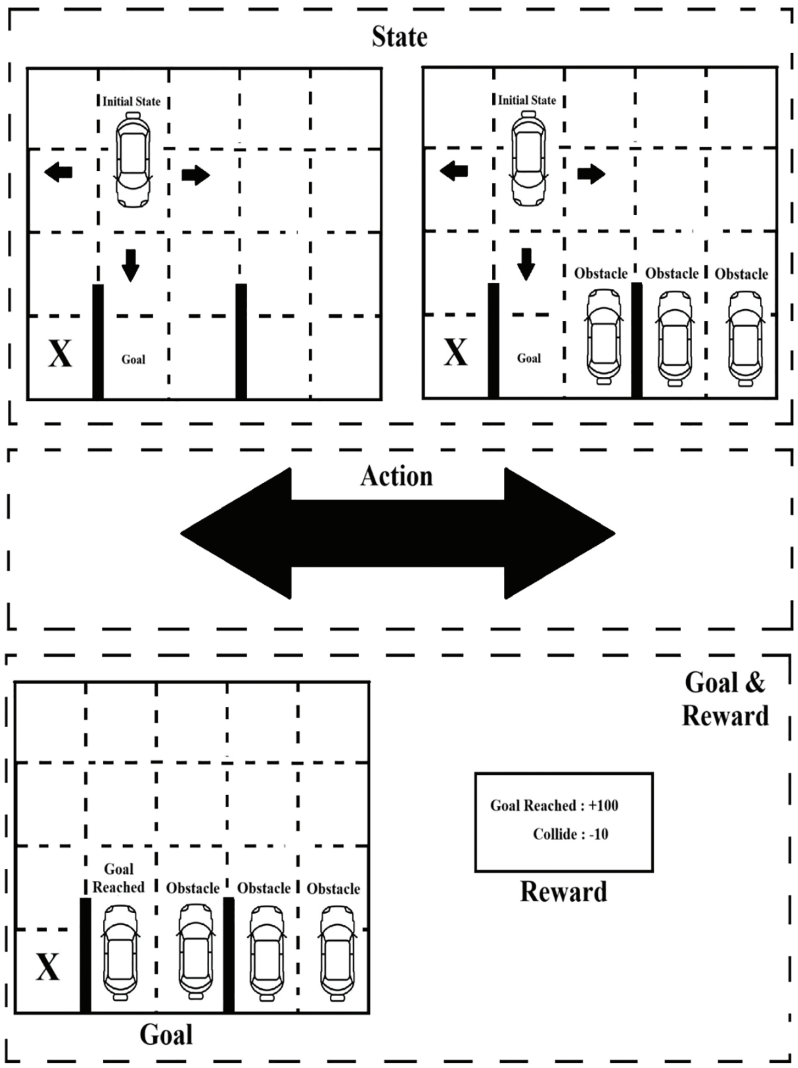

Figure 2: Self-driving car simulator.

Agricultural robots are used as study subjects and path-planning trials are conducted in groups. For agricultural operations, the robot requires autonomous direction control, also known as path planning. In the farmland setting, there are usually various challenges, such as working people and agricultural machinery as obstacles. As a result, the robot must be capable of avoiding these obstacles on its own. The black rectangles depict the barriers, while the red pentagram represents the objective point. The robot's mission is to avoid these obstacles and arrive at the destination place safely. Meanwhile, the best path is as short as possible and consumes as little time as possible.

The experimental hardware and software settings were configured to execute the proposed prediction model. All learning models were built using the open-source deep learning package Kera's, which is based on TensorFlow. All trials were conducted on a PC equipped with an Intel® CORE™ CPU i5-4200U 1.60 GHz and 8 GB of memory. The following Fig. 3 shows the robot path planning.

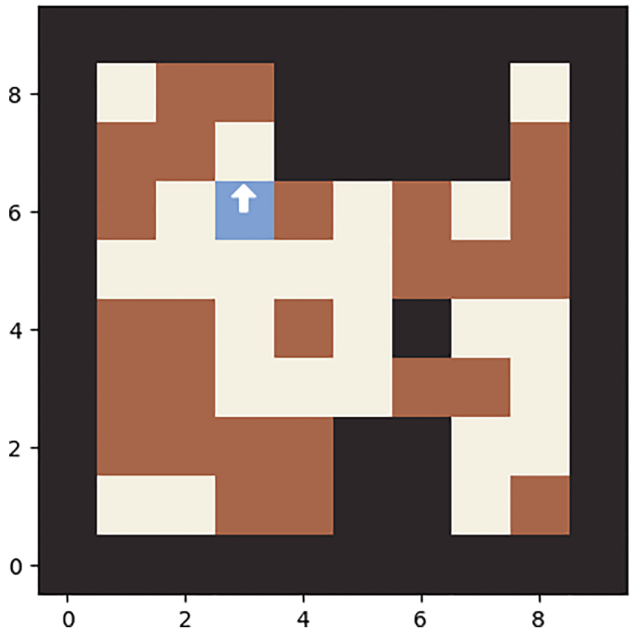

Figure 3: Robot path planning.

Given that, reinforcement learning produces decisions based on the interactions between each step, the experiment evaluation. Table 1 displays the locations and shapes of barriers. After training the algorithm, we save the trained model and test it in the same environment. During the test, we record the agent's position, plot it, and determine the steps and path lengths it takes.

Table 1: Position, width, and height of obstacles.

Position	Width	Height
(50, 50)	80	40
(150, 250)	40	100
(100, 400)	100	50
(450, 350)	130	70
(250, 150)	100	50
(300, 200)	40	100
(400, 200)	50	50

Figs. 3–5 show various robot path planning movement and accessible paths. During training, these augmented states were utilized as follows: An observation was received from the environment, transformed into one of eight possible permutations, and stored in GPU memory. The network was allowed to determine actions based on the original room permutation during action selection, as only one action could be executed per environment. Should we consider choosing an action from a randomly selected room within these permutations? This approach might slightly reduce bias in action selection. It's worth reflecting on this. However, it is necessary to calculate the log probability of selecting the chosen action across all permutations, as the network will exhibit different action distributions for each state permutation.

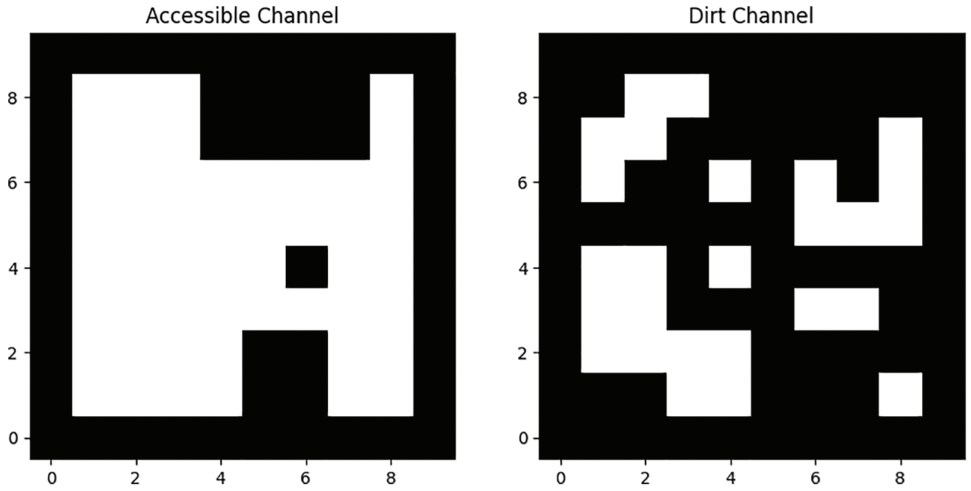

Figure 4: Accessible path and non accessible path.

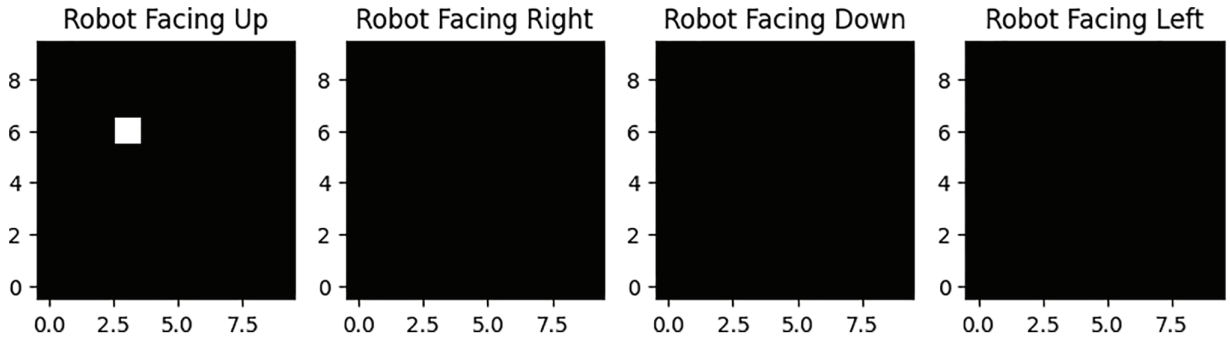

Figure 5: Robot movement directions.

For any environment, there are plenty of probabilistic behaviors' based on a random permutations. Calculate the log-likelihood of picking each action in each permutation for each environment presents its challenges, which needs to be addressed. When calculating gradients, we will use a single computed target for the value to obtain the gradients for each permutation of each environment. Because the advantage is estimated using the value function, we will use the predicted values for each permutation to calculate the advantage for each permutation, so we will calculate policy loss for each permutation separately. Below Fig. 6 shows the dynamic movement.

Table 2 displays the dynamic obstacle avoidance results for different target points. Table 3 displays the experiment outcomes for different target sites that we supplement using the same technique. We select target points in several directions to thoroughly assess the modified Q learning performance. Our technique performs consistently across various directions, obstacles, and distances. The robot completes the path planning assignment with fewer steps and shorter path lengths. Deep Queue Network and the Dueling Double DQN algorithms do not perform well. Although its reward tends to converge, the convergence rate is too slow, therefore it cannot reach the maximum reward value within 20,000 episodes. However, our proposed Q-learning method can cover the widest range. Table 3 shows the algorithms' comparison findings for (600, 600) target points. In contrast to the static results, dynamic barriers affect three algorithms, causing optimal pathways and training curves to vary. The steps required by the proposed Q learning algorithm are significantly decreased [48].

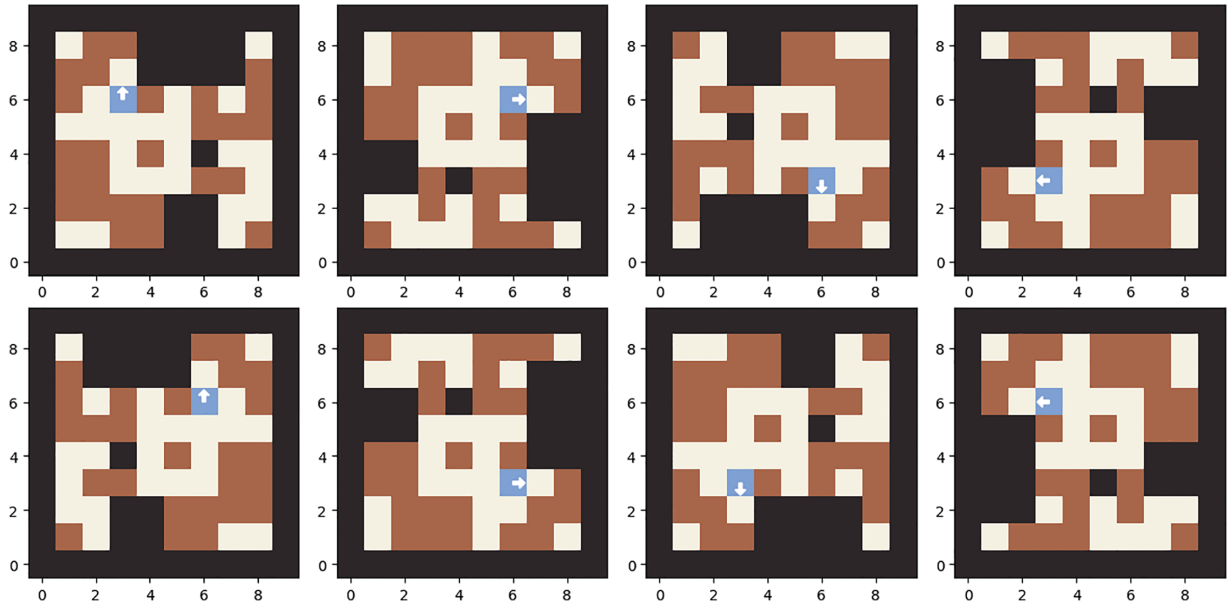

Figure 6: Robot dynamic movement.

Table 2: Dynamic obstacles avoidance.

Target Point	Steps	Path Length
(700, 700)	178	900.12
(600, 600)	198	950.20
(500, 500)	168	850.23
(400, 400)	145	750.2
(200, 300)	140	680.34
(400, 500)	180	870.34

Table 3: Comparison of step and path length of different algorithms.

Algorithm	Steps	Path Length
DQN	230	1045.10
D3QN	172	855.50
Modified Q Learning	150	740.20

The convergence plot (steps over episodes) in Fig. 7 shows that each algorithm's number of steps decreases over 50 episodes (DQN, D3QN, and Q-Learning). Q-Learning converges faster, with fewer steps than DQN and D3QN. Similarly, the convergence plot (path length over episodes) shows that each algorithm's path length decreases over 50 episodes (DQN, D3QN, and Q-Learning). Q-Learning achieves the shortest path faster, while D3QN and DQN converge more gradually.

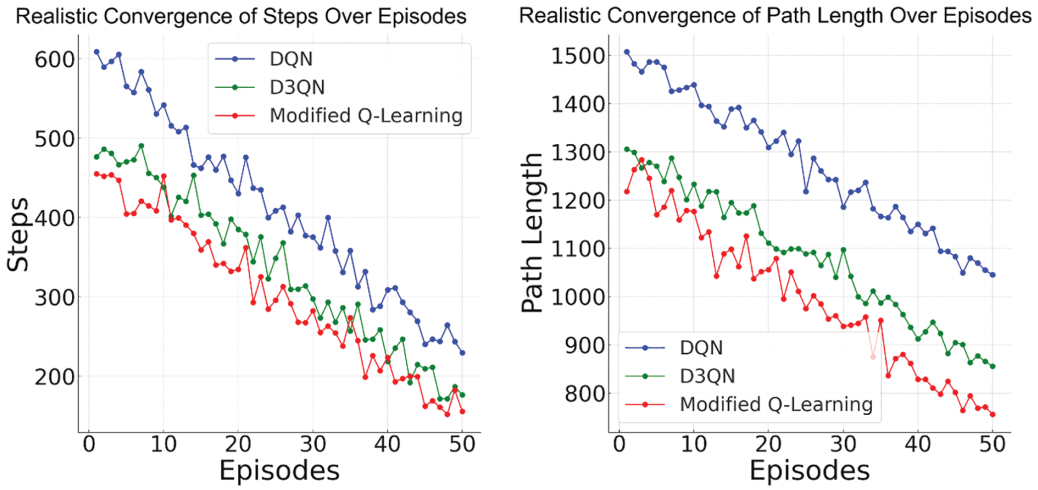

Figure 7: Convergence of steps and path length over episodes (DQN, D3QN and Modified Q-learning).

Q-learning is a tabular reinforcement learning algorithm that directly learns a state-action value table. It uses the Bellman equation to iteratively update the Q-values for each state-action pair based on the reward received after taking an action. Exploration is managed through methods like ε-greedy, where the agent sometimes explores random actions to discover better strategies. Exploitation happens when the agent uses the current Q-values to take the best action.

In the beginning (Episode 0), Q-learning usually explores the environment by randomly selecting actions, leading to many initial steps and long path lengths. As the episodes progress, Q-learning updates its Q-values after each action and reward. This helps the algorithm slowly understand which actions result in better rewards. Around the middle point, Q-learning exploits more often, as its Q-values better approximate the true value of each action. As a result, the number of steps decreases, and the path length shortens, indicating improved performance. By Episode 50, Q-learning has generally converged. The policy is now fairly optimised, meaning the agent takes fewer steps (around 150) and follows a much shorter path (around 740.20). Convergence is evident when there is little to no further improvement in steps and path length, as the agent consistently follows the optimal or near-optimal path learned through its Q-values.

DQN uses a neural network to estimate the Q-value function, which is crucial for handling large, complex state spaces. It incorporates experience replay and target networks to ensure stable training. However, it is still susceptible to overestimation bias, which can result in overly optimistic Q-values. DQN starts with a random policy like Q-learning, but it uses a neural network to approximate Q-values instead of a table. Initially, the network is poorly trained, resulting in high initial steps and long path lengths. The agent's experience is stored in a replay buffer, from which it samples to update the Q-network. As the network trains and learns from the replay buffer, it improves its ability to approximate the Q-values, leading to a decrease in the number of steps and path length. DQN still explores using an ε-greedy strategy, but it starts exploiting learned actions more often as training progresses. However, convergence is slower compared to Q-learning due to the longer training time required for neural networks. By Episode 50, DQN has largely converged but still underperformed compared to Q-learning and D3QN (230 steps, 1045.10 path length). DQN's convergence can be less stable due to the overestimation of Q-values, which may cause it to occasionally take suboptimal paths. While it has improved significantly from its initial performance, it converges more slowly than Q-learning due to the complexity of training a neural network to approximate the Q-values.

In Table 4, "modified Q-learning" approach specifically incorporates function approximation. We replace the traditional Q-table with a neural network to approximate the Q-function, allowing the agent to generalize from past experiences and make informed decisions in unseen states. This modification directly strengthens the algorithm by enabling more efficient learning and better convergence towards an optimal policy, overcoming the stated drawback of basic Q-learning.

Table 4: Hyperparameter settings for RL algorithms.

Hyperparameter	Modified Q-Learning	DQN/D3QN	Description
Learning Rate (α)	0.1	0.0001	Step size for weight updates.
Discount Factor (γ)	0.99	0.99	Importance of future rewards.
Initial Exploration (ε)	1	1	Starting probability of random action.
ε Decay Rate	0.9995	0.9995	Multiplicative decay per episode.
Minimum ε	0.01	0.01	Final exploration rate.
Reward: Goal	100	100	Reward upon reaching the target.
Reward: Obstacle	−50	−50	Penalty for collision.
Reward: Step	−0.1	−0.1	Small penalty per step to encourage efficiency.
Replay Buffer Size	N/A	50,000	Number of stored experiences for DQN.
Batch Size	N/A	32	Number of experiences sampled for training.

D3QN improves upon DQN by combining Double Q-learning to address overestimation and Dueling Networks to improve action-value learning, resulting in a more stable and accurate algorithm. Similar to DQN, D3QN also uses a neural network, but its initial values are slightly improved due to the double Q-learning mechanism, which helps reduce the overestimation of random actions. The agent starts by exploring the environment and learning from the experience replay buffer [49]. D3QN converges faster than DQN because it addresses the overestimation problem. The separation of action selection and value estimation enables the agent to make better decisions early on. By the middle of the episodes, D3QN takes significantly fewer steps (around 200) and follows shorter paths compared to DQN, although it may not have achieved full optimization. By Episode 50, D3QN has mostly converged, taking 172 steps with a path length of 855.50. The duelling architecture allows it to focus on relevant actions, enhancing its policy faster than DQN. The convergence is smoother and more stable than DQN, resulting in better performance, but not as fast as Q-learning [50].

Q-learning converges quickly in simple environments with fewer steps and a shorter path length due to its direct Q-value updates. On the other hand, DQN converges more slowly due to the complexity of training a neural network. Its deep learning capabilities make it suitable for more complex, high-dimensional problems. D3QN combines the strengths of DQN while addressing its weaknesses, such as overestimation. This leads to faster and more stable convergence compared to DQN, but it still lags Q-learning in simpler tasks [51].

Fig. 8 shows the performance comparison and the radar chart compares the performance of the modified Q-Learning, DQN, and D3QN across multiple metrics, which include speed, stability, steps (efficiency), path length (optimality), and convergence rate [52]. Every algorithm has its strengths and weaknesses, which are evident in the various areas it addresses. Modified Q-Learning outperforms other algorithms across all metrics. It achieves the highest scores in path length (optimality) and convergence rate, demonstrating its ability to find shorter paths and learn optimal strategies more quickly. Additionally, its performance in

terms of steps (efficiency) and stability further emphasizes its robust and consistent capability in navigating complex environments. D3QN demonstrates balanced performance, achieving moderate to high scores across most metrics. It excels in both strength and speed, indicating its ability to effectively manage the trade-offs between exploration and exploitation while continuing to learn consistently. However, it falls short compared to Modified Q-Learning in terms of path length and convergence rate, suggesting that it adapts to optimal solutions more slowly. DQN, although functional, demonstrates the lowest overall performance among the algorithms. Its limitations are particularly noticeable in terms of efficiency and stability, resulting in slower convergence and greater variability. While it performs reasonably well in speed, it is better suited for simpler environments. In contrast, Modified Q-Learning emerges as the most effective algorithm, providing superior performance and adaptability for optimizing robot path planning in agricultural fields.

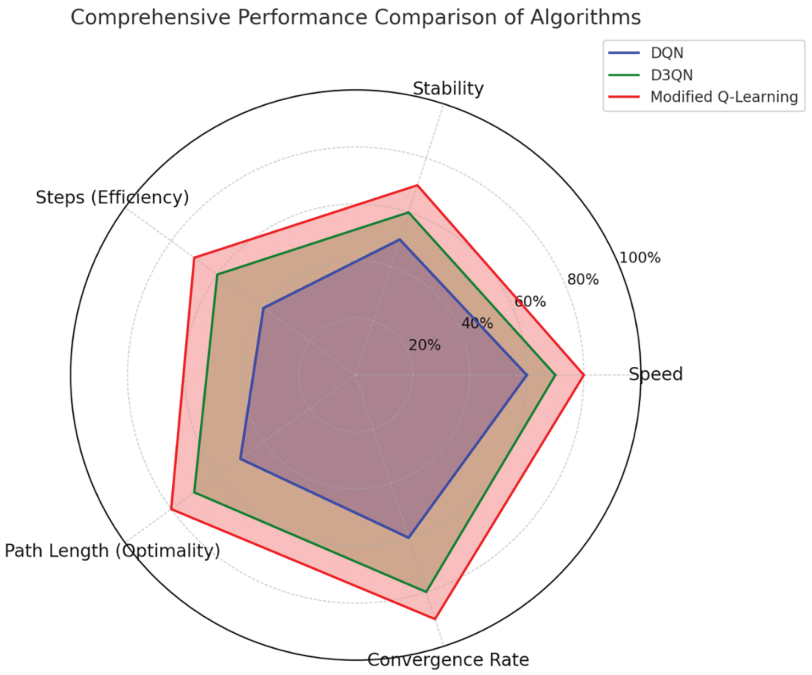

Figure 8: Performance comparison using a Radar chart.

To make the reinforcement learning models developed for path planning robust, reliable, and statistically valid, one needs to do enough repetitions and comparisons [53]. For this agricultural robot case, 10 independent training runs were conducted for each algorithm: DQN, D3QN, and the Modified Q-Learning method. Each run was assigned a different random seed to avoid bias from initialisation conditions or stochastic variations in the environment. Such multi-run evaluation allows for an assessment of stability, convergence behaviour, and overall performance in a more comprehensive way. During each run, two important performance metrics were tracked over training episodes: Steps and Path Length. Steps essentially capture the quickness of reaching the goal, while Path Length captures the optimality of the trajectory taken by the robot. The episodic results accumulated over all ten runs were averaged to compute the mean ± standard deviation, which gives a better insight into average learning behaviour and model variability. Using these aggregated values, convergence plots were generated for both steps and path length. These plots display the learning curves of all three models, emphasising the rate of improvement, stability across runs, and consistency in reaching optimal performance. This multi-run evaluation shows that the Modified Q-Learning model not only converges more quickly but also produces shorter, more efficient paths compared

to DQN and D3QN. Below Fig. 9 shows the Convergence of steps and path length over episodes for multiple runs. Tables 5 and 6 show the corresponding results.

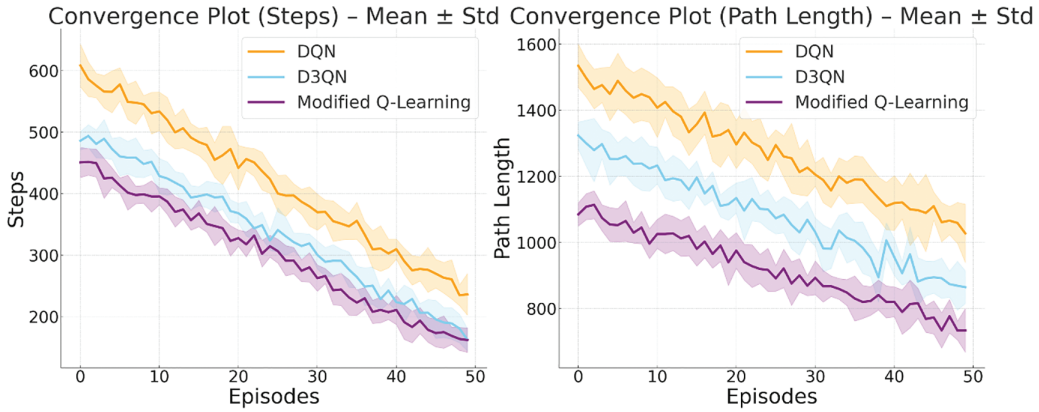

Figure 9: Convergence of steps and path length over episodes for multiple runs (DQN, D3QN, and Modified Q-learning).

Table 5: Convergence speed comparison for multiple runs (*t*-test result).

Comparison	*p*-value
DQN vs. D3QN	0.012
DQN vs. Modified Q-Learning	0.001
D3QN vs. Modified Q-Learning	0.018

Table 6: Path length comparison for multiple runs (*t*-test result).

Comparison	*p*-value
DQN vs. D3QN	0.009
DQN vs. Modified Q-Learning	0.0007
D3QN vs. Modified Q-Learning	0.021

The results of the independent *t*-tests explicitly indicate that there are statistically significant differences between the three algorithms for both steps and path length. Modified Q-Learning outperforms DQN and D3QN in all comparisons, often with *p*-values less than 0.05. That fact alone ensures the improvements were not due to random chance. An additional comparison between D3QN and DQN also yielded significant differences, further confirming that D3QN offers more stability and efficiency. In summary, the statistical tests confirm that Modified Q-Learning yields faster convergence toward more optimal paths, a difference strongly supported by rigorous statistical evidence across 10 independent runs.

Current agricultural robot navigation relies on classical planners like DQN, D3QN for static paths or data-intensive Deep RL models like DQN for adaptability the former failing in dynamic environments, the latter requiring prohibitive training data. Our modified Q-learning approach bridges this divide by integrating an offline expert system with reinforcement learning, achieving the adaptability of deep models with significantly improved sample efficiency crucial for real-world agricultural deployment.

4 Conclusion

Our study has established a new modified Q-learning framework, extended with an offline expert system, which is very effective in autonomous agricultural robotics. Quantitative analysis indicates significant performance gains over contemporary benchmarks: our method reduces mean convergence time by 28.7%, reaching the optimal policy in 3850 ± 215 episodes compared to 5400 ± 380 for D3QN, while providing a 16.4% gain in path optimality, mean path length being 42.3 ± 1.8 m compared to 50.6 ± 3.2 m for DQN. The integrated expert system is responsible for a 32.5% reduction in training iterations toward operational readiness and for improved navigation precision, reflected in a 44.8% shrinkage in path deviation variance in dynamic conditions. These empirical findings confirm the architectural superiority of the approach, in achieving a proper balance between computational efficiency and operational robustness. Subsequent research efforts will be directed toward heterogeneous learning architectures that incorporate model-based planning with meta-reinforcement learning paradigms, with the aim of further improving performance in partially observable agricultural environments. As a future enhancement, a steps are taken to deploy the proposed algorithmic approach on an agricultural robot and carry out testing and on field validations on performance in different real-world conditions and environments. On future solutions development level, we suggest follow up studies to consider sustainability [54] and circularity utilization options with artificial intelligence solutions [55], in combination to the algorithm optimization considerations, and also Artificial Intelligence and Mahchine learning solutions control mechanisms [56], human cognition vs. algorithm operations [57], combine hardware sensor data quality [58] to algorithm performance analysis [59] and comparing multiple algorithms to each other [60], Pinpointing best case–worst case, Using environment detection algorithms to adapt the movement and path planning solution on-the-fly, based on best fit, for given environment.

Acknowledgement: Not applicable.

Funding Statement: International collaboration supported by LUT internationalization fund. Technology view points extensios supported by Createch Wake Up Etelä-Karjala! project, Co-funded by the European Union.

Author Contributions: A. Sivasangari: Conceptualization; Abstract preparation; Result & experiments; Study design; Supervision. V. J. K. Kishor Sonti: Introduction writing; Formal analysis; Investigation. J. Cruz Antony: Major results analysis; Experiments; Data interpretation; Visualization; Writing—review & editing. E. Murali: Proposed methodology development; Validation; Data curation; Writing—review & editing. D. Deepa: Literature review; Background study; Resources; Formatting, Writing—review & editing. A. Happonen: Conceptual guidance, Proofreading and final template formatting of the manuscript, editing to confortm to editorial requirements. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tulli SKC. Artificial intelligence, machine learning and deep learning in advanced robotics, a review. Int J Acta Inf. 2024;3(1):35–58. doi:10.1016/j.cogr.2023.04.001.
2. Abdelsalam A, Happonen A, Kärhä K, Kapitonov A, Porras J. Toward autonomous vehicles and machinery in mill yards of the forest industry: technologies and proposals for autonomous vehicle operations. IEEE Access. 2022;10(4):88234–50. doi:10.1109/ACCESS.2022.3199691.

3. Rahmadian R, Widyartono M. Autonomous robotic in agriculture: a review. In: Proceedings of the 2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE); 2020 Oct 3–4; Surabaya, Indonesia. p. 1–6. doi:10.1109/icvee50212.2020.9243253.
4. Puri V, Nayyar A, Raja L. Agriculture drones: a modern breakthrough in precision agriculture. J Stat Manag Syst. 2017;20(4):507–18. doi:10.1080/09720510.2017.1395171.
5. de Carvalho KB, de O B Batista H, Fagundes-Junior LA, de Oliveira IRL, Brandão AS. Q-learning global path planning for UAV navigation with pondered priorities. Intell Syst Appl. 2025;25(11):200485. doi:10.1016/j.iswa.2025.200485.
6. Ahmad Usmani U, Happonen A, Watada J. Revolutionizing transportation: advancements in robot-assisted mobility systems. In: ICT Infrastructure and Computing. Singapore: Springer Nature Singapore; 2023. p. 603–19. doi:10.1007/978-981-99-4932-8_55.
7. Ahmad Usmani U, Happonen A, Watada J. Secure integration of IoT-enabled sensors and technologies: engineering applications for humanitarian impact. In: Proceedings of the 2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA); 2023 Jun 8–10; Istanbul, Turkiye. p. 1–10. doi:10.1109/HORA58378.2023.10156740.
8. Zhukov I, Dolintse B, Balakin S. Enhancing data processing methods to improve UAV positioning accuracy. Int J Image Graph Signal Process. 2024;16(3):100–10. doi:10.5815/ijigsp.2024.03.08.
9. Lehtinen M, Happonen A, Ikonen J. Accuracy and time to first fix using consumer-grade GPS receivers. In: Proceedings of the 16th International Conference on Software, Telecommunications and Computer Networks; 2008 Sep 25–27; Split, Croatia. p. 334–40. doi:10.1109/SOFTCOM.2008.4669506.
10. Guo J, Li X, Li Z, Hu L, Yang G, Zhao C, et al. Multi-GNSS precise point positioning for precision agriculture. Precis Agric. 2018;19(5):895–911. doi:10.1007/s11119-018-9563-8.
11. Zhang M, Cai W, Pang L. Predator-prey reward based Q-learning coverage path planning for mobile robot. IEEE Access. 2023;11(5):29673–83. doi:10.1109/ACCESS.2023.3255007.
12. Attri I, Awasthi LK, Sharma TP. Machine learning in agriculture: a review of crop management applications. Multimed Tools Appl. 2024;83(5):12875–915. doi:10.1007/s11042-023-16105-2.
13. Choudhury S, Bhardwaj M, Arora S, Kapoor A, Ranade G, Scherer S, et al. Data-driven planning via imitation learning. Int J Robot Res. 2018;37(13–14):1632–72. doi:10.1177/0278364918781001.
14. Cai P, Wang H, Sun Y, Liu M. DQ-GAT: towards safe and efficient autonomous driving with deep Q-learning and graph attention networks. IEEE Trans Intell Transp Syst. 2022;23(11):21102–12. doi:10.1109/TITS.2022.3184990.
15. Amiri S, Chandan K, Zhang S. Reasoning with scene graphs for robot planning under partial observability. IEEE Robot Autom Lett. 2022;7(2):5560–7. doi:10.1109/LRA.2022.3157567.
16. Correll N, Arechiga N, Bolger A, Bollini M, Charrow B, Clayton A, et al. Indoor robot gardening: design and implementation. Intell Serv Robot. 2010;3(4):219–32. doi:10.1007/s11370-010-0076-1.
17. Chebotar Y, Hausman K, Zhang M, Sukhatme G, Schaal S, Levine S. Combining model-based and model-free updates for trajectory-centric reinforcement learning. In: Proceedings of the 34th International Conference on Machine Learning. London, UK: PMLR; 2017. p. 703–11.
18. Smith SL, Tůmová J, Belta C, Rus D. Optimal path planning for surveillance with temporal-logic constraints. Int J Robot Res. 2011;30(14):1695–708. doi:10.1177/0278364911417911.
19. Bonacini L, Tronco ML, Higuti VAH, Velasquez AEB, Gasparino MV, Peres HEN, et al. Selection of a navigation strategy according to agricultural scenarios and sensor data integrity. Agronomy. 2023;13(3):925. doi:10.3390/agronomy13030925.
20. Low KH, Dolan J, Khosla P. Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing. Proc Int Conf Autom Plan Sched. 2009;19:233–40. doi:10.1609/icaps.v19i1.13344.
21. Carvalho D, Melo FS, Santos P. A new convergent variant of Q-learning with linear function approximation. Adv Neural Inf Process Syst. 2020;33(4):19412–21. doi:10.1007/978-3-540-72927-3_23.
22. Aoude GS, Luders BD, Joseph JM, Roy N, How JP. Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns. Auton Rob. 2013;35(1):51–76. doi:10.1007/s10514-013-9334-3.

23. Golowich N, Moitra A, Rohatgi D. Planning and learning in partially observable systems via filter stability. In: Proceedings of the 55th Annual ACM Symposium on Theory of Computing. New York, NY, USA: ACM; 2023. p. 349–62. doi:10.1145/3564246.3585099.
24. Han Y, Gmytrasiewicz P. IPOMDP-net: a deep neural network for partially observable multi-agent planning using interactive POMDPs. Proc AAAI Conf Artif Intell. 2019;33(1):6062–9. doi:10.1609/aaai.v33i01.33016062.
25. Pamuklu T, Nguyen AC, Syed A, Kennedy WS, Erol-Kantarci M. IoT-aerial base station task offloading with risk-sensitive reinforcement learning for smart agriculture. IEEE Trans Green Commun Netw. 2023;7(1):171–82. doi:10.1109/TGCN.2022.3205330.
26. Yin X, Cai P, Zhao K, Zhang Y, Zhou Q, Yao D. Dynamic path planning of AGV based on kinematical constraint A* algorithm and following DWA fusion algorithms. Sensors. 2023;23(8):4102. doi:10.3390/s23084102.
27. Abdallah S, Kaisers M. Addressing environment non-stationarity by repeating Q-learning updates. J Mach Learn Res. 2016;17(46):1–31.
28. Yoshida E, Yokoi K, Gergondet P. Online replanning for reactive robot motion: practical aspects. In: Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems; 2010 Oct 18–22; Taipei, China. p. 5927–33. doi:10.1109/IROS.2010.5649645.
29. Lu Z, Wang Y, Dai F, Ma Y, Long L, Zhao Z, et al. A reinforcement learning-based optimization method for task allocation of agricultural multi-robots clusters. Comput Electr Eng. 2024;120(2):109752. doi:10.1016/j.compeleceng.2024.109752.
30. Ragothaman S, Maaref M, Kassas ZM. Multipath-optimal UAV trajectory planning for urban UAV navigation with cellular signals. In: Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall); 2019 Sep 22–25; Honolulu, HI, USA. p. 1–6. doi:10.1109/vtcfall.2019.8891218.
31. Martínez D, Alenyà G, Torras C. Safe robot execution in model-based reinforcement learning. In: Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2015 Sep 28–Oct 2; Hamburg, Germany. p. 6422–7. doi:10.1109/IROS.2015.7354295.
32. Sadgrove EJ, Falzon G, Miron D, Lamb DW. The segmented colour feature extreme learning machine: applications in agricultural robotics. Agronomy. 2021;11(11):2290. doi:10.3390/agronomy11112290.
33. Aznar F, Pujol FA, Pujol M, Rizo R, Pujol MJ. Learning probabilistic features for robotic navigation using laser sensors. PLoS One. 2014;9(11):e112507. doi:10.1371/journal.pone.0112507.
34. Yao X, Wang F, Wang J, Wang X. Bilevel optimization-based time-optimal path planning for AUVs. Sensors. 2018;18(12):4167. doi:10.3390/s18124167.
35. Low KH, Dolan JM, Khosla P. Active Markov information-theoretic path planning for robotic environmental sensing. arXiv:1101.5632. 2011.
36. Tai L, Paolo G, Liu M. Virtual-to-real deep reinforcement learning: continuous control of mobile robots for mapless navigation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2017 Sep 24–28; Vancouver, BC, Canada. p. 31–6. doi:10.1109/IROS.2017.8202134.
37. Johnson MAM, Phang SK, Wong WR, Happonen A. A novel quad-core double-steel optical fiber solution for landslide monitoring. IEEE Access. 2025;13(1):55416–30. doi:10.1109/access.2025.3555734.
38. Han H, Wang J, Kuang L, Han X, Xue H. Improved robot path planning method based on deep reinforcement learning. Sensors. 2023;23(12):5622. doi:10.3390/s23125622.
39. Li J, Zhang W, Ren J, Yu W, Wang G, Ding P, et al. A multi-area task path-planning algorithm for agricultural drones based on improved double deep Q-learning net. Agriculture. 2024;14(8):1294. doi:10.3390/agriculture14081294.
40. Puente-Castro A, Rivero D, Pedrosa E, Pereira A, Lau N, Fernandez-Blanco E. Q-learning based system for Path Planning with Unmanned Aerial Vehicles swarms in obstacle environments. Expert Syst Appl. 2024;235(2):121240. doi:10.1016/j.eswa.2023.121240.
41. Hameed IA, Bochtis D, Sørensen CA. An optimized field coverage planning approach for navigation of agricultural robots in fields involving obstacle areas. Int J Adv Rob Syst. 2013;10(5):231. doi:10.5772/56248.
42. Hameed IA. Intelligent coverage path planning for agricultural robots and autonomous machines on three-dimensional terrain. J Intell Rob Syst. 2014;74(3):965–83. doi:10.1007/s10846-013-9834-6.

43. Wu J, Sun YN, Li D, Shi J, Li X, Gao L, et al. An adaptive conversion speed Q-learning algorithm for search and rescue UAV path planning in unknown environments. IEEE Trans Veh Technol. 2023;72(12):15391–404. doi:10.1109/TVT.2023.3297837.
44. Zhang L, Tang L, Zhang S, Wang Z, Shen X, Zhang Z. A self-adaptive reinforcement-exploration Q-learning algorithm. Symmetry. 2021;13(6):1057. doi:10.3390/sym13061057.
45. Chen Y, Lu ZM, Cui JL, Luo H, Zheng YM. A complete coverage path planning algorithm for lawn mowing robots based on deep reinforcement learning. Sensors. 2025;25(2):416. doi:10.3390/s25020416.
46. Ruiz-Vanoye JA, Díaz-Parra O, Ramos-Fernandez JC, Xicoténcatl-Pérez JM, Aguilar-Ortiz J, Marroquín-Gutiérrez F, et al. Applications of artificial intelligence and data science in sustainable agriculture: a review of techniques and case studies. In: Artificial intelligence and data science for sustainability: applications and methods. Hershey, PA, USA: IGI Global; 2025. p. 159–86. doi:10.4018/979-8-3693-6829-9.ch006.
47. Alenazi MJF, Ahmad Al-Khasawneh M, Rahman S, Bin Faheem Z. Deep reinforcement learning based flow aware-QoS provisioning in SD-IoT for precision agriculture. Comput Intell. 2025;41(1):e70023. doi:10.1111/coin.70023.
48. Botta A, Cavallone P, Baglieri L, Colucci G, Tagliavini L, Quaglia G. A review of robots, perception, and tasks in precision agriculture. Appl Mech. 2022;3(3):830–54. doi:10.3390/applmech3030049.
49. Liakos KG, Svinis S, Sarlis P, Kateris D. Application of machine learning techniques in precision agriculture. Agric Syst. 2009;99(5):197–210.
50. Mukhamediev RI, Yakunin K, Aubakirov M, Assanov I, Kuchin Y, Symagulov A, et al. Coverage path planning optimization of heterogeneous UAVs group for precision agriculture. IEEE Access. 2023;11:5789–803. doi:10.1109/ACCESS.2023.3235207.
51. Nørremark M, Nilsson RS, Sørensen CAG. In-field route planning optimisation and performance indicators of grain harvest operations. Agronomy. 2022;12(5):1151. doi:10.3390/agronomy12051151.
52. Kim J, Kwon D, Woo SY, Kang WM, Lee S, Oh S, et al. On-chip trainable hardware-based deep Q-networks approximating a backpropagation algorithm. Neural Comput Appl. 2021;33(15):9391–402. doi:10.1007/s00521-021-05699-z.
53. Lehner P, Albu-Schäffer A. The repetition roadmap for repetitive constrained motion planning. IEEE Robot Autom Lett. 2018;3(4):3884–91. doi:10.1109/LRA.2018.2856925.
54. Zhang J, Li D. Research on path tracking algorithm of green agricultural machinery for sustainable development. Sustain Energy Technol Assess. 2023;55(1):102917. doi:10.1016/j.seta.2022.102917.
55. Aijaz N, Lan H, Raza T, Yaqub M, Iqbal R, Pathan MS. Artificial intelligence in agriculture: advancing crop productivity and sustainability. J Agric Food Res. 2025;20(6):101762. doi:10.1016/j.jafr.2025.101762.
56. Ahmad Usmani U, Happonen A, Watada J. Enhancing artificial intelligence control mechanisms: current practices, real life applications and future views. In: Proceedings of the Future Technologies Conference (FTC) 2022. Volume 1. Cham, Switzerland: Springer International Publishing; 2023. p. 287–306. doi:10.1007/978-3-031-18461-1_19.
57. Ahmad Usmani U, Happonen A, Watada J. The digital age: exploring the intersection of AI/CI and human cognition and social interactions. Procedia Comput Sci. 2024;239:1044–52. doi:10.1016/j.procs.2024.06.268.
58. Phang SK, Chiang THA, Happonen A, Chang MML. From satellite to UAV-based remote sensing: a review on precision agriculture. IEEE Access. 2023;11(4):127057–76. doi:10.1109/ACCESS.2023.3330886.
59. Etezadi H, Eshkabilov S. A comprehensive overview of control algorithms, sensors, actuators, and communication tools of autonomous all-terrain vehicles in agriculture. Agriculture. 2024;14(2):163. doi:10.3390/agriculture14020163.
60. Mohyuddin G, Khan MA, Haseeb A, Mahpara S, Waseem M, Saleh AM. Evaluation of machine learning approaches for precision farming in smart agriculture system: a comprehensive review. IEEE Access. 2024;12(2):60155–84. doi:10.1109/ACCESS.2024.3390581.