WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 3, Number 2, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v3n2.40
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Deep Reinforcement Learning for Scan Recognition and Path Planning of Rescue UAVs

Minchao Ma

Mechanical engineering, The Hong Kong Polytechnic University, Hong Kong, China
21106075d@connect.polyu.hk

ABSTRACT

In recent years, rapid advances in communication and navigation technologies have led to the research of drones in disaster relief becoming a controversial topic of interest. These drones are capable of autonomously performing high-risk tasks under extreme and treacherous conditions, such as searching and locating disaster victims, effectively replacing human intervention. The purpose of this essay is to discuss the basic overview of rescue UAVs, to find how deep reinforcement learning techniques on the MATLAB platform can be used to improve the effectiveness of UAV recognition scanning and path planning. Subsequently discuss the future development path and potential applications of rescue UAV technology in a forward-looking manner.

KEYWORDS

Deep Reinforcement Learning; Object Recognition; Markov Decision Process; Path Planning

1. INTRODUCTION

With the rapid progress of society, the topic of disaster prevention and control is increasingly receiving widespread attention. In the Thirteenth Five-Year Plan for the Construction of the National Emergency Response System, UAVs have been incorporated into the emergency rescue system as professional equipment, which indicates that they will become an emerging player in emergency rescue. Nevertheless, there is still room for improvement in the effectiveness and accuracy of UAVs. It is crucial to improve the scanning algorithms of drones and incorporate deep and reinforcement learning techniques to enhance their ability to identify disaster hazards and locate affected people. The power of drones in rescue operations is clear, from searching and locating people in distress, to monitoring the scene in real time, and finally to performing post-disaster search and rescue assessments. For example, drones equipped with infrared thermal imagers can detect disaster hazards in low visibility environments and prevent potential disasters. Compared with traditional human means of investigation and rescue, drones show more significant advantages: high flexibility, rapid response capability, high data accuracy, and high collection efficiency. These advantages greatly reduce the safety risks and labour intensity of rescuers. This not only highlights the important role of scientific and technological innovation in enhancing the level of emergency management, but also provides strong support for building a more efficient and safe rescue system.

2. THE IMPACT OF DEEP LEARNING IN UAV TARGET RECOGNITION

The application of Artificial Intelligence (AI) technology in UAV target recognition is a remarkable trend in the development of science and technology in recent years. This emerging technology has brought revolutionary changes to UAVs, especially in the field of processing and analyzing image data. The excellence of AI deep learning in target recognition is analyzed next in several ways.

Firstly, after a UAV captures image data of a target area using on-board sensors during a mission, the AI system automatically extracts features from the image that contribute to target recognition through deep learning algorithms, particularly Convolutional Neural Networks (CNNs). These features may include the shape, texture, colour and even temperature distribution of the target. The power of AI deep learning is its ability to automatically learn and extract these features without human intervention or selection. After extracting the features, the AI model must be trained to learn how to recognize different targets. In this process, the model needs to be provided with a large amount of image data, through which it constantly adjusts its internal parameters to distinguish between different categories. This training process will determine the level of application of the AI model carried by the UAV in the field, which is directly related to its execution efficiency and accuracy [1].

In practice, a Pytorch-based deep learning framework can be built to integrate CNN and RNN models. The dataset is directly adopted from publicly available videos and images captured by drones, through which the models are trained to recognize and classify various targets. These data are usually subjected to a series of pre-processing steps, including image denoising, contrast enhancement, image cropping. Before they are fed into the AI model training, with the aim of improving the accuracy and efficiency of the subsequent processing. Model training should be considered using cross-entropy loss function and Adam optimizer for model training. During the training process, the focus is on monitoring the accuracy and loss, and periodically adjusting the learning rate and other hyperparameters to optimize the model performance. After the model training is completed, the performance of the UAV target recognition system before and after the improvement is compared in various environments.

3. UAV PATH OPTIMIZATION BASED ON MATLAB REINFORCEMENT LEARNING

With the rapid evolution of machine learning, people have begun to try to use methods based on deep learning and reinforcement learning to solve path planning problems. One of the most representative algorithms is the standard Q-learning algorithm based on the Markov decision process. As a supervised learning method, this algorithm can plan a better path for the UAV through a learning mechanism according to the changes in the environment.

In the current technological context, the autonomy and efficiency of UAVs in performing search and rescue missions is crucial. In order to improve the navigation ability of UAVs in complex environments, reinforcement learning provides an effective way to optimize flight paths. MATLAB, as a powerful numerical computation and simulation platform, provides a series of tools and function libraries, especially the Reinforcement Learning Toolbox [2]. Next, I will explore in detail how to use the Reinforcement Learning in MATLAB to optimize the flight path of the rescue UAV.

3.1. Mission Setting for Rescue UAVs

In rescue missions, UAVs may need to fly in unknown environments, search for missing persons, or assess disaster situations. These tasks require the UAV to be able to plan its flight path autonomously, avoid various obstacles, quickly reach the target area to perform the task, and return to the starting point. Therefore, the flight path optimization problem for rescue UAVs can be modelled as a Markov decision process, where:

State: the current position, speed, and direction of the UAV, as well as information about the environment (e.g., the location of obstacles).

Action: control commands for the UAV, such as accelerate, decelerate, ascend, descend, turn left, turn right.

Reward: a function that evaluates the effect of the drone's current action. For example, the UAV obtains a positive reward when it moves towards the target area and a negative reward when it collides with an obstacle or deviates from the path.

3.2. Reinforcement learning using MATLAB

The MATLAB software has a toolbox called Reinforcement Learning Toolbox which contains a range of tools for designing, simulating and training reinforcement learning intelligences. The following are the steps to perform reinforcement learning using MATLAB:

3.2.1. Modelling the environment

Firstly a simulation environment needs to be created which mirrors the scenario of a UAV in the real world. For this simulation environment I have used a simplified abstract model. In the Reinforcement Learning Toolbox we can use custom environments. The custom environment requires us to define states and actions to implement a step function that calculates the next state and reward based on the current state and action.

3.2.2. Intelligent Body Design

The intelligent body is the decision maker in reinforcement learning. In MATLAB, we can choose many types of intelligent bodies, such as Q-Learning, Deep Q-Network (DQN) and so on [3]. Each type of intelligence has its specific uses and advantages. For example, DQN is suitable for problems with discrete action spaces. The design of an intelligent body also includes choosing an appropriate neural network architecture and setting hyperparameters such as learning rate, discount factor, etc.

3.2.3. Training and Evaluation

Training is the core process of reinforcement learning. In MATLAB, we can use the train function to train the intelligent body. The training process involves the intelligent body continuously trying different actions in a simulated environment and updating its strategy based on the rewards. The training can continue for multiple rounds until the intelligence reaches the desired level of performance.

3.2.4. Deployment and practical application

Once the drone training is complete and has reached the desired level in the simulated environment, we can deploy it to an actual drone. This usually involves converting the trained neural network model into code that can be executed by the drone and ensuring that the model can run on the drone's hardware. A number of field tests are also required before actual deployment to ensure that the drone can work in real-world complex environments [4].

3.3. Code Simulation

In order to make the UAV reinforcement learning achieve the desired effect, I used MATLAB’s toolbox to build out a simulation environment for reinforcement learning and show a part of the code as shown below.

```
>> % Setting the initial conditions
numStates = [10, 10, 10]; % Size of the state space
numActions = 6; % Size of the action space
initialState = [1, 1, 1]; % initial state
goalState = [10, 10, 10]; % target state
obstacles = [5, 5, 5; 6, 6, 6; 7, 7, 7]; % List of obstacles
% Initializing the Q-table
Q = zeros(prod(numStates), numActions); % Simplifying Q-Tables with One-Dimensional Indexes

% Setting training parameters
numEpisodes = 1000; % Total number of training rounds
maxSteps = 100; % Maximum number of steps per round
learningRate = 0.1; % learning rate
discountFactor = 0.9; % discount factor
epsilon = 0.1; % exploring factor
```

```
function [nextState, reward] = executeAction(state, action, obstacles, numStates)
    % Get action vector based on action number
    action_vector = getActionVector(action);

    % Get the next state based on the current state and action
    proposedState = getNextState(state, action_vector, numStates);

    % Determine if the boundary is exceeded
    isOutOfBounds = any(proposedState < 1) || any(proposedState > numStates);
    if isOutOfBounds
        nextState = state; % If the boundary is exceeded, the current state remains unchanged
        reward = -10; % Giving negative incentives
    else
        nextState = proposedState;
        % Determine whether the target state has been reached
        goalState = [10, 10, 10]; % Replace with the actual target state value
        if isequal(nextState, goalState)
            reward = 100; % High rewards if targets are met
        else
            reward = -1; % Otherwise negative incentives
        end

        % Determine if an obstacle has been encountered
        if any(ismember(obstacles, nextState, 'rows'))
            reward = -10; % Gives greater negative bonus if obstacles are encountered
            nextState = state;
        end
    end
```

Figure 1. the part of simulation code to demonstrate the environmental parameters and reward mechanisms

These images show how I used MATLAB code to establish the environmental parameters and reward mechanisms for reinforcement learning.

The above code is built in accordance with the previous code of reinforcement learning in part 3.2, and the result after ten times of path simulation deduction is shown in the figure (mapping 3D coordinates into 2D coordinates, the values of x, y, z axes are equal at the same time, 1 → (1, 1, 1) is the starting point, 10 → (10, 10, 10) is the end point, and 5, 6, and 7 are the coordinates of the obstacles).

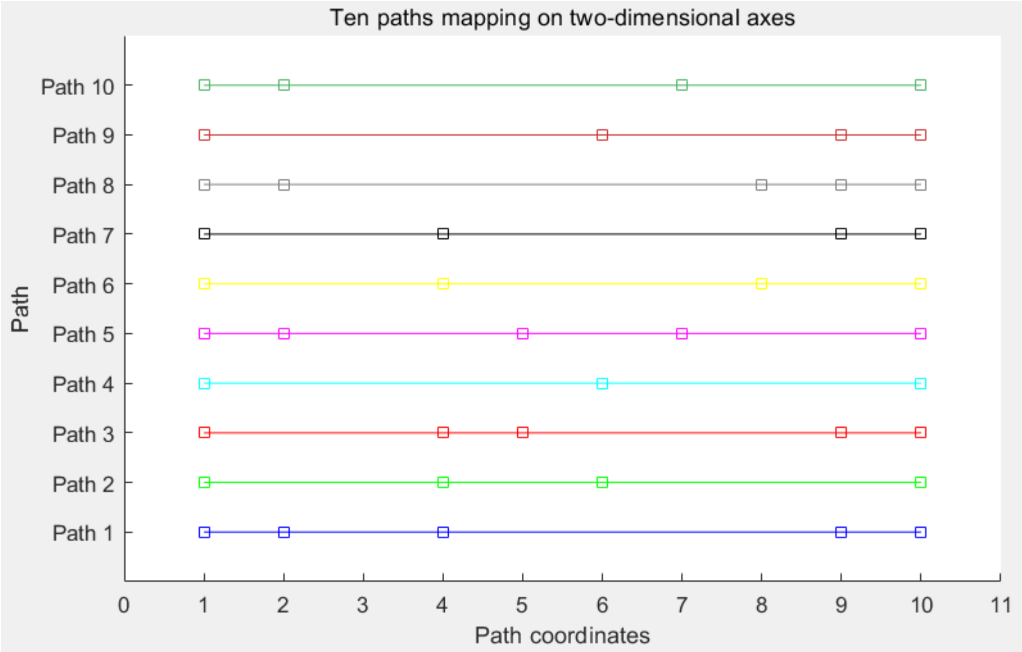

Figure 2. Path coordinates for ten times path simulation

In a total of 10 route-planning extrapolations, obstacle avoidance was successfully accomplished in only 4 of the first, sixth, seventh, and eighth route-planning sessions, reaching a not-so-ideal success rate of 40%. In order to study and improve the programme, I will address it later in the discussion section.

4. DISCUSSION

In UAV target recognition, the incorporation of AI deep learning techniques has demonstrated significant advantages. Firstly, the trained AI model can greatly improve the accuracy of recognition. By learning from a large amount of sample data, the model ensures high-precision target recognition even in complex and changing environments, which is especially critical for overcoming problems caused by target occlusion, changing light conditions, and background clutter. Second, with the continued development of computer hardware technology, advances in edge computing technology have greatly facilitated real-time data analysis of the built-in AI models of UAVs. This greatly shortens the time lag between data transmission and processing, which allows the UAV to respond instantly when performing tasks and improves the operational efficiency and flexibility of the UAV. In addition, deep learning algorithms have demonstrated strong generalization capabilities. After completing training on a large amount of heterogeneous data, the model is able to recognize new objects not included in the training samples because it grasps the essential properties of the target beyond its dependence on a specific situation. This is particularly important for object recognition tasks in UAVs, which must understand completely different environments and targets in real-world application scenarios. And over time, AI systems can improve their performance through an uninterrupted learning mechanism. With enough training data, learning models can quickly adapt and improve recognition accuracy when faced with unknown objects. This self-learning feature lays a solid foundation for the wide application of deep learning technology in the field of object recognition.

In addition, the system supports multi-threaded operation. This means that a single model can simultaneously handle multiple tasks such as object recognition, classification, and tracking. This integrated processing strategy improves resource utilization and time efficiency compared to conventional procedures. In terms of accuracy, the highly automated nature of AI systems reduces the number of human intervention steps and effectively reduces human error, which is particularly important for target recognition application scenarios that seek a high degree of objectivity and stability. Finally, emerging AI technologies can even effectively merge information from multiple sensors on the UAV, such as combining optical images with thermal radiation images, to achieve more comprehensive and accurate target identification. This cross-source fusion is particularly critical in scenarios where multiple sources of information need to be aggregated to make accurate judgements.

However, there are limitations to the application of deep learning in UAV target recognition. First, the performance of the model is highly dependent on the quality and quantity of the training data, and obtaining a large amount of high-quality labelled data is a resource-intensive task, which can raise the threshold for using deep learning and shut out most ordinary users. Second, deep learning models have poor interpretability. This is due to the fact that deep learning models typically contain a large number of layers and parameters that interact with each other in a complex, non-linear manner in a multi-layer structure. Such complexity makes it almost impossible for users to follow the decision paths within the model and difficult to understand the specific impact of individual parameters on the final output. This makes it difficult for users to trust the decisions of deep learning models.

To address these issues, visualization techniques can be used to show the features learned by the deep learning model at different levels, as a way to help users understand how the deep learning model gleans information from large amounts of raw data. Alternatively, AI frameworks that improve model interpretability on the web can be used, such as TensorBoard, which provides an interface and visualization capabilities for analyzing and interpreting model behaviour. Regarding reinforcement learning for UAV path planning, I simulated a Markov decision process using MATLAB code to solve the UAV path planning problem. The goal of the model is to avoid obstacles while having the UAV fly from the target starting point to the end point. Although the model is theoretically promising, the results in the simulation tests are not satisfactory.

Firstly, a key issue is the discretization of the state space. Whereas reality is the continuous physical world, the simulation divides the space into a finite number of discrete states. This can lead to loss of information, especially in 3D space, which is likely to have an impact on obstacle avoidance capabilities. In addition, the representation of obstacles is also limited by the state discretization, which may lead to less accurate representation of certain obstacles in the state space, thus increasing the risk of collision.

However, another limitation is the learning process itself. Q-Learning relies on a balance between exploration (by randomly selecting actions) and exploitation (by selecting the best-known action). An over-reliance on the best-known moves may cause the learning process to fall into a local optimal solution instead of a global optimal solution. This means that the path found by the UAV is not the safest or most efficient.

One way to improve this model is to refine the reward mechanism to reflect the complexity of the environment more accurately. For example, introducing rewards related to the distance to an obstacle encourages the UAV to take more careful decisions when approaching an obstacle. Consideration could also be given to using more advanced techniques to work with continuous state spaces, such as replacing the traditional Q-table by function approximation, which could help the model learn at a finer level.

5. CONCLUSION

This study has comprehensively explored the application of UAVs in disaster rescue, focusing on their technological advancements in target identification and path optimization. Through deep and reinforcement learning, the performance of UAVs has been enhanced, especially in performing precise search and rescue tasks in complex environments. However, the research equally reveals the challenges and limitations that current technologies face in practical applications. For example, deep learning models are highly dependent on the acquisition of large amounts of high-quality training data, and environment simulation for reinforcement learning is limited by the discretization of the state space, among others, and needs to be improved even further.

Despite the challenges, the integration of AI technology and UAS in disaster relief remains promising. Through continuous optimization and the combination of cross-disciplinary research, future UAVs will almost certainly be able to play a more effective role in disaster relief, providing greater protection and assistance to human society.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the theoretical support from Assistant Professor Lu Peng. The author's parents provided financial support to carry out this research.

REFERENCE

[1] WEI Mingxin, HUANG Hao, HU Yongming, WANG Dezhi, & LI Yuebin. (2020). Deep learning-based monocular visual target localization tracking method for multi-rotor UAVs. Computer Measurement and Control, 28(4), 156–160. https://doi.org/10.16526/j.cnki.11-4762/tp.2020.04.032

[2] VOLODYMYR M, KORAY K, DAVID S, et al. Human-level control through deep reinforcement learning [J]. Nature, 2019, 518(7540):29-33.

[3] LI R, FU L, WANG L, et al. Improved Q-learning based route planning method for UAVs in unknown environment[C/CD]//Proceedings in the 15th International Conference on Control and Automation (ICCA). 2019, Edinburgh, UK.

[4] ZHOU Bin, GUO Yan, LI Ning, et al. Path planning of UAV using guided enhancement Q-learning algorithm [J]. Acta Aeronautica et Astronautica Sinica, 2021, 42(9):506-513.