Application of Deep Reinforcement Learning to Path Planning for Quadrotor UAVs

Haotian Peng

School of Aeronautics, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu Province, 210016, China

ukc@asu.edu.pl

**Abstract.** The quadrotor Unmanned Aerial Vehicle(UAV) navigation autonomy fundamentally depends on optimal path planning methodologies. Classical algorithms demonstrate notable limitations in unknown complex settings: insufficient real-time responsiveness and suboptimal integration of global spatial data. For the path planning problem of quadrotor UAV, this paper systematically analyzes the application and advantages of the deep reinforcement learning algorithm Q-learning, and deep learning object detection model Region-based convolutional neural networks (R-CNN) algorithm, and the You Only Look Once (YOLO) algorithm. The results show that the Q-learning algorithm has the ability of dynamic decisions, but is limited by the problem of local information and state space explosion. The R-CNN algorithm has high detection accuracy in complex scenes, but the multi-stage processing leads to insufficient real-time performance. The YOLO algorithm achieves high-speed detection by virtue of the single-stage architecture, but its small target recognition ability is weak. Furthermore, it is proposed to combine the global path optimization of the Q-learning algorithm with the real-time environment perception of the YOLO algorithm, and significantly improve the obstacle avoidance efficiency and path planning reliability of the UAV in the dynamic environment through dynamic obstacle detection and reinforcement learning strategy update, which provides theoretical support and a certain reference for the UAV path planning.

# Introduction

Unmanned aerial vehicles (UAVs) are unmanned aerial vehicles that are autonomously controlled or remotely controlled. At present, drones are widely used in both military and civilian aspects, and in civilian areas such as environmental monitoring, disaster relief, and agricultural and forestry protection [1]. In the military, it is mostly used as transport aircraft, reconnaissance aircraft, bombers, etc.

In practice, drones often need to navigate through a space full of obstacles to reach a series of predetermined points [2]. This need has created an urgent need for efficient autonomous navigation systems. Path planning technology plays a crucial role in these systems. To this end, researchers have developed a variety of traditional path planning algorithms, including search-based methods (e.g. A\*, Dijkstra), sampling-based methods (e.g. Rapidly-exploring Random Trees), and bio-inspired algorithms that mimic nature's intelligence (e.g. genetic algorithms).

In 1956, Edsger Dijkstra et al. proposed the Dijkstra algorithm, which is a single-source shortest path algorithm with a time complexity of O(n^2) to search for the optimal path by constantly updating the shortest path length from the node to the starting point [3].

Therefore, to reduce the number of nodes, Hart et al.[4] proposed the A\*(A Star) algorithm in 1968, which is a graph search algorithm with a time complexity of O((V + E)logV), which maintains a list of explored nodes and nodes to be explored, and selects the optimal path according to the heuristic evaluation function of the nodes, which has better performance than other global path algorithms, but has the problems of low computational efficiency and multiple turning points [5]. In 1973, Holland et al. were the first to use genetic algorithms to solve the problem of robot path planning. Genetic algorithms are based on the idea of biological evolution, which searches for the solution space by defining fitness functions and genetic manipulations to obtain the optimal path [6]. In 1986, Rafael Fierro et al. proposed an algorithm based on artificial potential fields by introducing the concept of potential fields and planning paths by calculating gravitational and repulsive forces. In 1998, Steven Lavalle et al. proposed the RRT algorithm for path planning in randomly sampled spaces. It constructs a tree structure by randomly selecting sampling points and extending new branches to search for potential paths [7].

This has stimulated the pursuit of more intelligent path planning solutions, and deep reinforcement learning (DRL) is a key technology that has shown great potential in this field [1].

This study conducts a methodological comparative analysis of Q-learning, R-CNN, and YOLO computational frameworks for quadcopter trajectory planning, evaluating their respective strengths and limitations to identify the optimal approach for drone path planning. It proposes to use the real-time obstacle detection results of the YOLO algorithm as the state input of the Q-learning algorithm to optimize the global path through the dynamic strategy update mechanism of reinforcement learning.

# Q-learning Algorithm

Reinforcement learning is a branch of machine learning that can be used to solve sequential decision optimization problems, with Markov Decision Process (MDP) theory as its mathematical foundation. MDP can be represented as a quadruple

(1)

In the formula: is the set of state spaces of the system; A ={a1, a2, …, a n} is the set of actions that agents in the system can take; s the state transition strategy of the agent; is the reward obtained by the agent after each state transition. The Q-learning algorithm is also based on MDP.

The basic algorithm steps of the Q-learning algorithm can be summarized as five stages: initializing the Q-table, selecting an action, executing the action, updating the Q-value, and repeating the iteration.

The application scenarios of the Q-learning algorithm in drone path planning include urban logistics distribution, agricultural crop protection operations, disaster rescue search, and military reconnaissance tasks.

Compared with neural network algorithms, the Q-learning algorithm has a simpler process, smaller computational complexity, and faster computation speed. But this algorithm also has many flaws. The exploration process of the Q-learning algorithm is only based on MDP, and in each state transition process, it is only based on current information, lacking the utilization of global information [4]. This will affect the accuracy of the path in UAV path planning problems.

# R-CNN Algorithm

Commonly called the region proposal-based method, this detection framework operates through a dual-phase mechanism. First, it generates candidate regions across the full input image; subsequently, it performs object recognition within these regions. This sequential processing inherently defines it as a two-stage algorithm. Firstly, it is necessary to train the RPN network, and in the first stage, the RPN network generates candidate boxes. The second stage extracts data features by inputting candidate box data into a convolutional neural network, and then the classifier classifies the content of the candidate boxes. Region-based convolutional neural networks (R-CNN) represent prominent implementations of the two-stage detection paradigm. The R-CNN algorithm is known as the pioneering work of applying convolutional neural networks to image object detection.

The traditional object detection method generates candidate boxes of different sizes by sliding windows on the original image, and then inputs them into a classifier for recognition [1]. Finally, redundant boxes of the same object are removed through nonmaximum suppression, and the final result of object detection is output. The R-CNN algorithm has changed the strategy of exhaustive search in traditional object detection, using selective search to generate candidate boxes instead of the sliding window method for object extraction in traditional algorithms. However, the entire algorithm has problems such as an excessive number of candidate boxes, which leads to a lot of repetitive calculations, cumbersome steps, long training time, and large storage space occupation. In addition, due to the fixed dimensionality of the input of convolutional neural networks, normalization processing such as cropping and adjustment of candidate regions is required before inputting them into FClayer or SVM classifiers. This operation may result in the loss of information used for classification in the candidate boxes.

The R-CNN algorithm is a milestone algorithm in the field of object detection, which, for the first time, combines deep convolutional neural networks (CNN) with region proposals, significantly improving the accuracy of object detection and laying the foundation for subsequent algorithms such as Fast R-CNN and Faster R-CNN.

R-CNN, as a classic object detection algorithm, provides critical support for dynamic path planning of unmanned aerial vehicles in terms of environmental perception and semantic understanding. In urban canyon navigation, it can solve the problem of GPS signal loss caused by dense high-rise buildings, identify small obstacles during agricultural plant protection and obstacle avoidance, and provide reasonable path planning.

# YOLO Algorithm

One stage, also known as a single-stage object detection method, differs from a two-stage algorithm mainly in that the former does not require the generation of candidate boxes. After the input image passes through a network structure, the category confidence and predicted coordinate values of the object can be obtained simultaneously. Single-stage detectors substantially outperform region-based methods in inference speed for image recognition tasks. Representative architectures of this paradigm include the You Only Look Once (YOLO) framework.

The YOLO algorithm creatively combines the two stages of object classification and object detection into one, and the input image can complete object detection after one network structure, hence, it is called a single-stage object detection algorithm [1]. The YOLO algorithm does not infer the candidate boxes that need to be input into the classification predictor and bounding box coordinate regressor through the network model, but instead segments the input image into grids as predefined candidate prediction regions. The YOLO algorithm allows for the prediction of two object detection boxes per grid, thus generating a total of candidate regions.

The YOLO algorithm greatly improves the efficiency of the detection model by dividing the input image into grids. In addition, the YOLO algorithm uses the Darknet network instead of the classic VGG network, which significantly reduces the parameter and computational complexity of the model. However, due to the fact that each grid in the YOLO algorithm can only detect two target objects, the algorithm's performance in detecting small objects and neighboring targets is not very good.

The fundamental principle of YOLO reformulates object detection as an end-to-end regression framework, performing joint prediction of target bounding boxes and class probabilities directly from complete images. Renowned for its computational efficiency, YOLO excels in time-sensitive applications like autonomous vehicles and surveillance systems, marking a significant departure from the region-based methodology characterizing R-CNN series detectors [8].

YOLO-based object detection has demonstrated significant capabilities across diverse domains, including robotic vision systems, autonomous vehicle navigation, and human activity recognition. Nevertheless, drone applications encounter operational constraints in complex environments where reliable data links with ground stations are essential, potentially restricting technological innovation in specific use cases. Furthermore, deploying YOLO architectures necessitates high-performance computing platforms synchronized with visual data streams, imposing critical hardware and environmental prerequisites for implementation [9].

# Discussion and Analysis

In the field of drone path planning, Q-learning algorithm, R-CNN algorithm, and YOLO algorithm each have their own characteristics, but their performance and adaptability to application scenarios need to be further explored in combination with dynamic environmental requirements. Table 1 summarizes the application scenarios, advantages, and disadvantages, and real-time performance of various algorithms for unmanned aerial vehicle path planning [10].

**TABLE 1.** Comparison of Q-learning algorithm, R-CNN algorithm, and YOLO algorithm

|  |  |  |  |
| --- | --- | --- | --- |
|  | Q-learning | R-CNN | YOLO |
| Application scenarios | Suitable for decision optimization in dynamic environments | Suitable for providing input information for path planning based on environmental perception, as well as high-precision but low real-time scenarios | Suitable for real-time environmental perception and dynamic obstacle avoidance scenarios that require high-speed processing |
| Advantages | The Q-learning algorithm has dynamic adaptability and can update strategies in real-time during the path planning process to cope with sudden environmental changes. At the same time, it can achieve long-term planning, optimize the global path through a reward mechanism, and avoid local optimal traps. | The R-CNN algorithm has high detection accuracy and is based on a region proposal mechanism, making it suitable for small object detection in complex scenes. Can also provide  Multi task support, capable of simultaneously classifying and locating obstacles. | The YOLO algorithm has strong real-time performance and is suitable for real-time path planning due to its single-stage detection architecture. This algorithm is an end-to-end optimization that can directly output detection results and simplify the system architecture. |

Continue TABLE 1

|  |  |  |  |
| --- | --- | --- | --- |
| Disadvantages | The Q-learning algorithm has high computational complexity, suffers from state space explosion, and takes a long time to train. At the same time, its real-time performance is poor, which may not meet the high-speed flight requirements of drones. | The R-CNN algorithm has a large computational load and high latency due to the multi-stage processing of region proposal, feature extraction, and classification. Like Q-learning algorithm, its real-time performance is poor, making it difficult to meet the real-time obstacle avoidance requirements of drones. Moreover, this algorithm has high hardware requirements and relies heavily on high-performance GPUs, making it difficult to deploy on lightweight drone platforms. | The YOLO algorithm has weak detection capabilities for small targets, and may miss detecting distant or dense small obstacles during the detection process. Compared to the R-CNN algorithm, this algorithm has slightly lower detection accuracy, especially in complex backgrounds. |
| Real-time | Low | Low | High |

# Conclusion

While conventional algorithms demonstrate robust performance in structured settings, they encounter limitations when navigating unfamiliar complex environments. In such scenarios, deep reinforcement learning (DRL) showcases superior adaptability and dynamic decision-making capabilities.

Researchers are making breakthroughs in the direction of multi-algorithm collaboration and lightweight deployment, providing a solid technical foundation for the widespread application of drones in high-risk tasks such as military reconnaissance and disaster rescue. The fusion framework of the YOLO algorithm and the Q-learning algorithm proposed in this article is based on the exploration and practice of this trend.

Li W et al. mixed A \* and Q-learning algorithms and added an adaptive exploration factor to the improved algorithm, solving the problem of 3D unmanned aerial vehicle path planning. Carnegie Mellon University in the United States proposed the Q-learning RRT hybrid framework, which combines the dynamic decision-making ability of Q-learning with the global search of RRT, significantly improving planning efficiency in dynamic environments.

Taking inspiration from the above research, this article proposes to combine the Q-learning algorithm with the YOLO algorithm to detect obstacles in real time through YOLO algorithm, output to Q-learning algorithm as state input, and generate obstacle avoidance paths, which can perceive and make dynamic decisions in real time and are suitable for dynamic environments.

This algorithm can be applied to high-risk scenarios such as disaster relief and military reconnaissance, improving the success rate of tasks. However, there are still some limitations to the research. For example, there are hardware dependencies that require high-performance processors to support YOLO real-time detection, which limits its application in low-cost drones. Future research can be conducted in areas such as algorithm fusion and lightweight DRL models .

# ReferenceS

1. T. Zhou, "Research on UAV Vision Technology Based on GPU-Accelerated CNN Model," Ph.D. thesis, Harbin Institute of Technology, 2022.
2. M. Lü, C. Ding, H. Han, Acta Automatica Sinica 1-15 (2024).
3. M. Sun, Shandong Industrial Technology (21), 164 (2016).
4. P. E. Hart, N. J. Nilsson, and B. Raphael, IEEE Trans. Syst. Sci. Cybern. 4, 100-107 (1968).
5. B. Mohajer, K. Kiani, E. Samiei et al., Math. Probl. Eng. 2013 (2013).
6. J. H. Holland, J. Comput. 2, 88-105 (1973).
7. Y. Xin, H. Liang, M. Du et al., Robot 36, 627-633 (2014).
8. P. Yao, X. Wei, L. Yu et al., Electro-Optics Control 30(05), 16-22 (2023).
9. W. Li, W. Yin, M. Wong, J. Jian, IEEE Access 10, 7664-7674 (2022).
10. Q. Cheng, W. He, Z. Jia, J. Guizhou Normal Univ. 1-10 (2024).