

Article

Not peer-reviewed version

Autonomous UAV Target Search Method Based on Lightweight YOLOv8n and Coverage Path Planning

[Haoyan Duan](#), [Zhenhua Wang](#)^{*}, Mengtong Li, Zhenbo He, Haoxuan Zhang

Posted Date: 13 April 2026

doi: 10.20944/preprints202604.0823.v1

Keywords: UAV; object detection; path planning; breadth-first search; greedy algorithm



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

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

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

Article

Autonomous UAV Target Search Method Based on Lightweight YOLOv8n and Coverage Path Planning

Haoyan Duan ¹, Zhenhua Wang ^{1,*}, Mengtong Li ², Zhenbo He ¹ and Haoxuan Zhang ¹

¹ School of Artificial Intelligence, China University of Geosciences, Beijing 100083

² School of Computer Science, Peking University, Beijing 100871

* Correspondence: wangzh@cugb.edu.cn; Tel.: +86 18510800885

Highlights

What are the main findings?

- A lightweight YOLOv8n model is proposed via channel pruning and INT8 quantization, reducing inference latency on Raspberry Pi 4B to 83 ms while maintaining 84.6% mAP@0.5.
- Grid-Greedy-BFS (GGB) hybrid coverage path planning algorithm achieves 98.7% area coverage with 83.8% time saving compared to pure BFS.

What are the implications of the main findings?

- The method enables real-time autonomous target search on resource-constrained UAV platforms.
- The hybrid planning strategy balances coverage completeness and search efficiency, supporting dynamic target replanning in unknown environments.

Abstract

Unmanned aerial vehicles (UAVs) have wide application prospects in disaster search and rescue, ecological monitoring and environmental inspection tasks, where target search is a key link to realize autonomous task execution. UAVs often face challenges related to limited onboard computational resources and inefficient environmental coverage when used for target search. To address these issues, this paper proposes an autonomous search method for UAVs based on combined lightweight target detection and coverage path planning. In this method, the target search task was decomposed into two core parts target recognition and path planning. Firstly, in terms of target recognition, the YOLOv8n model was subjected to channel pruning and INT8 quantization to reduce its computational complexity, while HSV space data augmentation was incorporated to enhance recognition robustness in complex environments. Secondly, path planning was formulated as a dual-layer task comprising "spatial coverage + target confirmation." A grid-based search environment model was constructed, and a coverage path planning strategy was put forward that integrated Breadth-First Search (BFS) with local greedy optimization to achieve efficient traversal of unknown areas. Simultaneously, the A* algorithm was employed for path backtracking to cover omitted regions. Finally, a simulation platform for UAV target search was built to validate the recognition performance and search efficiency of the proposed method. The experimental results demonstrated that the proposed method significantly improved the UAV target search efficiency and reduced the path redundancy while ensuring the recognition accuracy, thereby offering an effective solution for autonomous UAV search on resource-constrained embedded platforms.

Keywords: UAV; object detection; path planning; breadth-first search; greedy algorithm

1. Introduction

In recent years, unmanned aerial vehicles (UAVs) have been widely used in emergency rescue and other fields [1–4], and its autonomous ability is highly dependent on two core technologies [5–7]: target recognition and efficient path planning in complex environments. In the “14th Five-Year Plan” for civil aviation development, it is pointed out that it is necessary to innovate the UAV industry ecology [8], and conduct operation scenario-oriented research on operation theory, risk assessment and technical verification based on operation risk analysis. UAVs are mostly used as the camera module in daily life, and the existing algorithms are largely designed for structured environments and offer limited adaptability to dynamic target search making that UAVs cannot be an independent unit for solve problems. In addition, the target recognition accuracy of UAVs may be lowered by illumination mutation, color distortion and occlusion in real life, and it is difficult to achieve full coverage cruise and dynamically response to new tasks in an unknown environment, posing serious challenges for the research of UAV target search algorithms [9,10].

For target recognition UAV onboard platforms generally have limitations in computational resources and power consumption, which cause trouble during direct deployment of large-scale deep neural networks. The YOLO series models achieve a favorable balance between speed and accuracy through their end-to-end single-stage detection architecture [11], while the standard YOLOv8n model exhibits an inference latency of 125–200ms on embedded platforms such as Raspberry Pi 4B, thereby failing to meet real-time processing demands. Existing studies have improved the inference speed of the YOLO series on embedded platforms to some extent through model pruning and quantization techniques [12–14], though a critical challenge remains in minimizing accuracy degradation while maintaining model robustness against varying illumination and color interference during this process [15]. Moreover, current methods predominantly focus on model optimization in specific scenarios and do not sufficiently account for the communication reliability associated with transmitting perception results [16–18].

For path planning Coverage Path Planning (CPP) can be considered as the core part of the search task. The traditional Breadth-First Search (BFS) algorithms have advantages in traversal completeness while disadvantages in excessive path redundancy; conversely, greedy algorithms gain high efficiency with critical areas overlooked probably [19,20]. Although some scholars have proposed hybrid algorithm, there're rarely experimental verification algorithms combining target detection confidence for dynamic task planning [21,22]. In addition, UAVs have high vulnerability to battery limitations, network disruptions and unstable communication links, which elevates bit error rates, increases transmission delays and ultimately undermines the practicability of such systems [7,23–25].

In view of the above problems, this paper proposes a comprehensive algorithm framework for UAV target recognition and path planning in complex environments, which mainly includes: 1) implementing model lightweighting for YOLOv8n through channel pruning and INT8 quantization to accelerate inference speed on Raspberry Pi 4B, and enhancing model robustness by introducing HSV color space data augmentation so as to meet real-time processing requirements in complex color scenarios. 2) achieving rapid full coverage with high efficiency and strong dynamic target replanning capacity by integrating the local rapid optimization capability of greedy algorithms with the global coverage completeness of breadth-first search.

Finally, simulation and experimental verification were conducted within the preconfigured simulated field environment. The results show that the proposed algorithm performs well under a variety of different interference conditions, especially in the wild animal search scene, the proposed algorithm has a maximum mAP@0.5 of 84.6%, mAP@0.5:0.95 of 56.7% and a search coverage of 98.7%, significantly surpassing current methods.

2. Related Work

UAV target comprehensively covers issues of environment perception, path planning and decision control. Traditional research methods usually optimize target recognition and path planning as independent modules, lacking system-level collaborative design [9,26].

For object recognition, early studies mainly relied on methods combining hand-designed feature extractors (such as SIFT, HOG) with classical machine learning classifiers (such as SVM) [27]. These methods have achieved certain achievements in specific scenes, with obviously insufficient generalization ability and robustness in complex conditions frequently faced by UAV platforms, such as viewpoint changes, illumination interference and diverse target scales [28,29]. With the rise of deep learning technology, the object detection algorithm based on convolutional neural network has been paid more and more attention, among which the YOLO series achieves a good balance between detection speed and accuracy due to its end-to-end single-stage detection architecture, thereby especially suitable for resource-constrained UAV aboard platforms [11].

In terms of path planning, traditional methods are mainly divided into two categories: classical algorithms and heuristic algorithms. Classical algorithms include A* algorithm, Dijkstra algorithm, artificial potential field method, etc., which function steadily in static environments while have difficulties in dealing with dynamically changing task requirements. Heuristic algorithms such as genetic algorithm, particle swarm optimization algorithm, ant colony algorithm and so on are widely used in path optimization in complex environments, which have the problems in slow convergence speed and aptness to fall into local optimum [5,22,30]. As a core component of search tasks, full Coverage Path Planning is traditionally addressed by algorithms like BFS, which ensures exhaustive coverage at the cost of low efficiency, and greedy algorithms, which prioritize speed while often miss critical areas [20]. According to the research in IEEE journal, in uncertain environments, the communication constraints, target dynamics and task collaboration have to be taken into considerations during the coverage search of UAV swarm, and the traditional methods are difficult to meet these requirements at the same time [31,32].

To solve the above problems, this paper proposes a UAV target search algorithm framework based on two core theories: lightweight deep learning theory and full Coverage Path Planning theory.

Lightweight deep learning theory provides a technical path for deploying complex vision models on resource-constrained UAV platforms. By removing redundant channels or connections in the neural network, model pruning can significantly decrease the amount of computation while maintaining model performance. Model quantization transforms high-precision floating-point parameters into low-bit integer representations, which greatly lowers the model size and accelerates inference [12,13,33]. Research by Soumyalatha et al. demonstrates that the combination of pruning and INT8 quantization technology reduces the parameters of YOLOv3 model by 80.39%, and improves the inference speed by 22.72% [13]. For the selection of YOLO series models, YOLOv8n has an all-round improvement in performance compared with YOLOv5n. On the premise of ensuring a similar model size, the data of YOLOv8n model trained on COCO show that its mAP@0.5:0.95 can be improved by about 10% compared with YOLOv5n. The results above laying a theoretical foundation for airborne real-time target detection.

The theory of full Coverage Path Planning provides a mathematical framework for the spatial traversal strategy of UAV search missions [5,34]. The full Coverage Path Planning problem can be modeled as finding an optimal path that can visit all passable grids in a given area. The BFS algorithm guarantees the completeness of the traversal from the perspective of graph theory, while the greedy algorithm pursues the local optimal solution from the perspective of optimization. The hybrid algorithm integrating the BFS and greedy algorithms enhances search efficiency while maintaining coverage [20]. At the same time, the target recognition module performs target comparison in real time during flights, so that the data of the two modules can be coordinated to ensure the dynamic replanning ability. In the review of Sheltami et al., the technical classification system of UAV path planning was systematically sorted out, and it was pointed out that about 30% of the studies were

involved in classical methods, 29% in metaheuristic methods, 18% in AI methods and 23% in hybrid methods [21]. Rahman et al. further emphasized that multi-UAV cooperative path planning posed requirements for designing intelligent algorithms with real-time re-planning abilities in complex scenarios [22].

3. Methodology

3.1. Task Modeling

In this paper, the UAV target search task is formalized as a dual-layer optimization problem: full space coverage and accurate target confirmation.

Firstly, the task environment was divided into a grid map of $M \times N$, and the size of each grid matched the field of view of the UAV, which was initially set to $1m \times 1m$. The map state could be represented as a matrix Z_{xy} , where the element m took the value 1 (visited), 0 (none visited) or -1 (obstacle) [20]. The goal of the full coverage layer was to plan a path, so that the UAV could traverse all the obstacle-free grids under the energy constraint to complete the initial search. Upon receiving high-confidence suspected target indications from the detection module, the target confirmation layer was activated, prompting the UAV to suspend its current coverage task and formulate a localized path for secondary detailed inspection of the designated area. This modeling method separates “wide-area search” from “fixed-point confirmation”, and takes into account both search breadth and mission reliability.

3.2. Model Architecture

Based on the ideas proposed by the above task modeling, this paper constructs a three-in-one algorithm architecture of “identification-transmission-planning”. The overall framework is shown in Figure 1.

Algorithm 1. Object Detection and Serial Transmission.

Require:

- serial: serial port
- detector: YOLO detector
- cam: camera
- disp: display

Ensure:

- Send target information packets

1: Initialize serial, detector, cam, disp

2: while not terminated do

3: img ← cam.read ()

4: objs ← detector.detect (img)

5: if objs is not empty then

```

6:     for each obj in objs do
7:         extract (x, y, w, h, label, score)
8:         data ← package (x, y, w, h, label, score)
9:         serial. send (data)
10:    end for
11: else
12:     data ← package (0, 0, 0, 0, 0, 0)
13:     serial. send (data)
14: end if
15: disp. show(img)
16: end while

```

The algorithm implements YOLO-based real-time object detection and sends the detection results to external devices through the serial port. Initially, the system initialized the serial communication, YOLO detection model, camera and display device. Then it entered the main loop: the image was obtained from the camera and input to the detector for object recognition. When the target was detected, the location information, category label and confidence of each target were extracted one by one, encapsulated as a data packet, and sent out through the serial port. In case no target was detected, an empty data packet was sent to indicate that there was no target at present.

Algorithm 2 Path Search and Compression

Require:

- start: starting point
- forbidden: no-fly zones
- directions: four directions

Ensure:

- Output a feasible compressed path

```

1: if not isReachable (start, forbidden) then return error
2: current ← start, path ← [start], visited[start] ← true
3: while there exist unvisited reachable nodes do
4:     next ← a valid neighboring node selected from directions
5:     if next exists then
6:         current ← next, visited[current] ← true, path ← path + current

```

```

7:   else
8:       break
9:   end if
10: end while
11: for each unvisited node p do
12:     path ← path + BFS(current, p)
13:     current ← p
14: end for
15: path ← compressPath (path)
16: path ← path + BFS (current, start)
17: return path

```

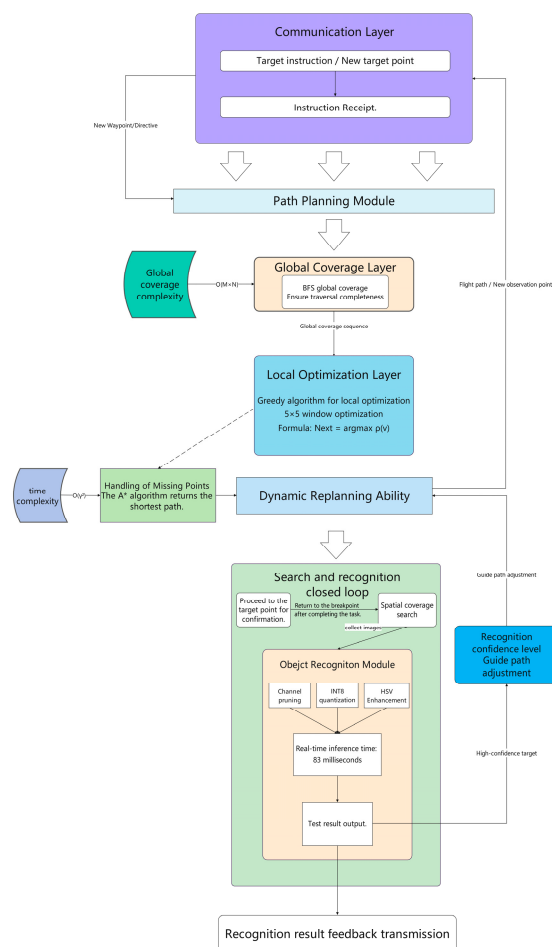


Figure 1.

This algorithm was designed to generate a feasible path in a grid environment with no-fly zone constraints, and the path went through compression subsequently. Firstly, the reachability test was used to determine whether all feasible areas could be covered from the starting point. An error was directly returned in case of any point unreachable. Then, starting from the initial point, the traversal strategy based on direction priority was adopted to gradually expand the path and mark the visited nodes until no further progress could be made. For any remaining unvisited nodes, the BFS algorithm was utilized to identify the shortest paths from the current position to these nodes, which were then integrated into the main path to ensure comprehensive coverage of all reachable regions. After the coverage was completed, the path was compressed to remove redundant intermediate points and only the turning points were retained to reduce the path length and execution cost. Finally, the algorithm connected the current position back to the starting point through one BFS to form a closed path.

3.2.1. Object Recognition Module

The target recognition module uses YOLOv8n as the basic model. Its core convolution formula is as follows.

$$Y(i, j, k) = \sum_{c=1}^C \sum_{m=1}^M \sum_{n=1}^N X(i+m, j+n, c) \cdot W(m, n, c, k) + b_k. \quad (1)$$

The goal of the YOLO loss function is to minimize the difference between each predicted box and the true box. The formula is as follows:

$$L = \lambda_{coord} \sum_{i=0}^{S^2} \mathbb{1}_{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2] c + \lambda_{noobj} \sum_{i=0}^{S^2} \mathbb{1}_{noobj} [(C_i - \hat{C}_i)^2] + \sum_{i=0}^{S^2} \mathbb{1}_{obj} [(p_i - \hat{p}_i)^2 + (C_i - \hat{C}_i)^2]. \quad (2)$$

In parallel, lightweight optimization was performed specifically for the Raspberry Pi 4B platform. Firstly, sparse training was carried out on the final C3 module to remove 40% redundant channels. After pruning, the mAP@0.5 loss of the model was controlled within 2%, which effectively reduced the computational complexity. Secondly, TensorRT INT8 quantization technology was used. The model size was compressed from 6 MB to 4.2 MB by combining the two methods of pruning and quantization, and the recognition speed of a single image was improved from 48ms/img to 26ms/img. The overall model inference latency reached 83ms, meeting the requirement for real-time onboard processing. Channel pruning and INT8 quantization were used to achieve lightweight compression. The loss of the pruned model is calculated as follows:

$$\Delta mAP = \frac{mAP_{after} - mAP_{before}}{mAP_{before}}, \quad (3)$$

the inference speed improvement formula after quantization is:

$$Speedup = \frac{T_{before}}{T_{after}}. \quad (4)$$

In addition, the model further incorporated an HSV color space data augmentation strategy to account for complex lighting conditions and color interference. Specifically, random adjustments were applied, including H-shift offsets within the range of $[-15^\circ, +15^\circ]$, S-scale factors between $[0.5, 1.5]$ and V-shift offsets within $[-20, +20]$. This approach emulated realistic illumination variations in practical scenarios and enhanced the model's adaptability to color changes. The proposed strategy made the mAP attenuation amplitude of the model less than 5% on the low-light and strong reflection test subsets, which verified the effectiveness of the proposed strategy.

Table 1. Common symbols and meanings.

Symbols	Meaning
X	Input feature map
Y	Output feature map
W	Convolution kernel
b_k	Bias for the KTH kernel
i, j	Outputs the spatial position coordinates in the feature map
m, n	Internal coordinates of the convolution kernel
c	Input channel index
k	Output channel (kernel number)
C	Number of channels to input the feature map
M, N	The size of the convolution kernel
K	Number of convolution kernels (number of output channels)
S	Size of the mesh
x_i, y_i	The center position of the target box (predicted value)
\hat{x}_i, \hat{y}_i	The center position of the true box (target value)
w_i, h_i	Width and height of the target box (predicted value)
\hat{w}_i, \hat{h}_i	Width and height of the true box (target)
C_i	Confidence score for the target box
p_i	The probability of the target class

$\lambda_{coord}, \lambda_{noobj}$	Weighting factors that modulate the effect of different terms
v	The neighborhood of the current grid
$N(u)$	The neighborhood of the current raster u
$\rho(v)$	The density value of the raster
$g(n)$	The known cost to the starting point n
$h(n)$	n A heuristic estimate to the goal
$Z(x, y)$	Map state matrix
(x, y)	Missing point coordinates
γ	Number of dynamic obstacles

3.2.2. Path Planning Module

The path planning module proposed a Grid-Greedy-BFS hybrid search algorithm (GGB). The algorithm adopted a two-layer architecture: the upper layer (global coverage layer) generated a global coverage sequence based on BFS to ensure the completeness of the traversal. The lower layer (local optimization layer) optimized the subsequent selection based on the greedy strategy in the current local window, and gave priority to the raster with the highest density of unvisited neighbors to reduce unnecessary backtracking, as shown in Figure 2.

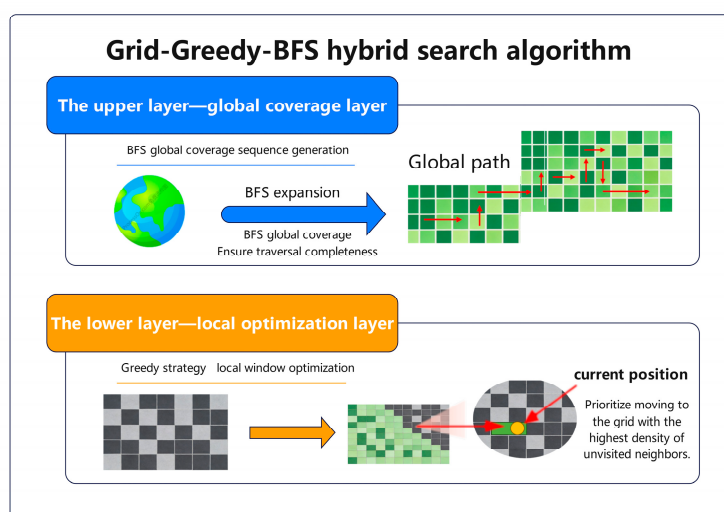


Figure 2.

The local optimum selection formula of the lower-level algorithm is as follows:

$$\text{Next Position} = \arg \max_{v \in N(u)} \rho(v) . \quad (5)$$

In addition, to speed up the path search and take advantage of the target location information, the A* algorithm was used to calculate the shortest path from the current point to the missing point. After each step, the A* algorithm selected the expansion node according to the formula:

$$f(n) = g(n) + h(n) , \quad (6)$$

Manhattan heuristic was used when 4-connected h_{Man} :

$$h_{\text{Man}}(n) = |x_n - x_{\text{goal}}| + |y_n - y_{\text{goal}}| , \quad (7)$$

The Euclidean heuristic was chosen when diagonal motion was allowed h_{Euc} :

$$h_{\text{Euc}}(n) = \sqrt{(x_n - x_{\text{goal}})^2 + (y_n - y_{\text{goal}})^2} . \quad (8)$$

At the same time, the module was gifted with an integrated mechanism for handling missed areas and dynamic replanning capabilities. Upon the initial traversal, it identified and charted the shortest route to access unvisited "void" regions. The processing of omission points was achieved through the computation of the shortest return path. When its formula was:

$$\text{Missing Points} = \{(x, y) | Z(x, y) = 0\} . \quad (9)$$

When the target recognition module found a suspected target, the system suspended the current BFS task, inserted the target point as a high-priority node, and the greedy algorithm planed a path to confirm it. After completing the task, coverage was continued from the breakpoint.

Compared with the pure BFS algorithm, the GGB algorithm reduces the invalid backtracking in the traversal process through local greedy inspiration, and the task completion time is saved by 83.8% compared with the pure BFS algorithm under the premise of ensuring the coverage rate. Compared with pure greedy algorithm, GGB algorithm avoids the omission of critical areas, and the coverage rate is increased by 16.4%.

3.2.3. Overall Time Complexity of the Algorithm

The time complexity of the object recognition module's algorithm is predominantly determined by the operations in the convolutional and fully connected layers. Given an input image of dimensions $W \times H$, with C representing the number of input channels and C' denoting the number of output channels, and assuming the convolution kernel has an area of k^2 , the time complexity of the convolution operation is:

$$O(C \times W \times H \times C' \times k^2) . \quad (10)$$

The detection part of YOLOv8n includes multiple convolutions and fully connected calculations, so its overall time complexity is usually a function of the image size and the number of model layers. Since the model is pruned, its time complexity is calculated according to the reduction of redundant channels. The new time complexity can be expressed as follows:

$$O(0.6 \times C \times W \times H \times C' \times k^2) . \quad (11)$$

Since the task environment is divided into a grid map of $M \times N$, the BFS full coverage time complexity should be $O(M \times N)$. The greedy algorithm selects the optimal neighbor node at each step, so for each grid node, it needs to traverse at most four neighbors (up and down, left and right). Therefore, in the worst-case scenario, the time complexity of the local optimization is $O(4 \times M \times N)$, indicating that each grid node undergoes at most four comparison operations. Because the overall

time complexity of the GGB algorithm is a combination of the full coverage layer (BFS) and the local optimization layer (greedy), the time complexity of the whole algorithm is: $O(M \times N)$.

Furthermore, the time complexity of dynamic replanning is dependent on the number of paths requiring replanning, denoted as γ . For the computation of paths involving missed points, the A* algorithm was employed to determine the shortest path. The time complexity of the A* algorithm was expressed as $O(\gamma^2)$, which corresponded to the overall time complexity of dynamic replanning and missed-point processing. Putting the above parts together, the time complexity of the overall algorithm is as follows: $O(M \times N) + O(\gamma^2)$. For large-scale scenes, the complexity of the algorithm is mainly determined by the raster coverage part.

4. Experimental Results and Analysis

4.1. Dataset and Training

The experiments of the proposed object recognition algorithm are mainly carried out with the COCO 2017 dataset. The dataset consisted of 118287 images in the training set, 5000 images in the validation set and 40670 images in the test set. In this part of the experiment, a total of 9483 images from four categories were selected as subsets, which were divided into a training set and a test set by the ratio of 7:3. In the overall experiment of UAV target search algorithm, the public data set from MaixCAM was selected. It contained a total of 2000 640×640 images with 10 categories of animal identities. The first experiment used AdamW optimizer with initial learning rate 1e-3, momentum 0.9, weight decay 0.0005, batch size 32, and 200 rounds of training. The second experiment employed SGD optimizer with initial learning rate 0.01, momentum 0.9, weight decay 0.0005, batch size 32, and 20 training rounds.

4.2. Experimental Setup

The experimental hardware platform with a UAV as the core unit incorporates a MaixCAM camera for target recognition and a Raspberry Pi 4B as the onboard computer for rapid path planning. The software environment utilizes Ubuntu 20.04 as the operating system and PyTorch 1.10 as the deep learning framework. The experiment was conducted in an indoor setting with a simulated 30m × 20m outdoor environment, as illustrated in Figure 3.

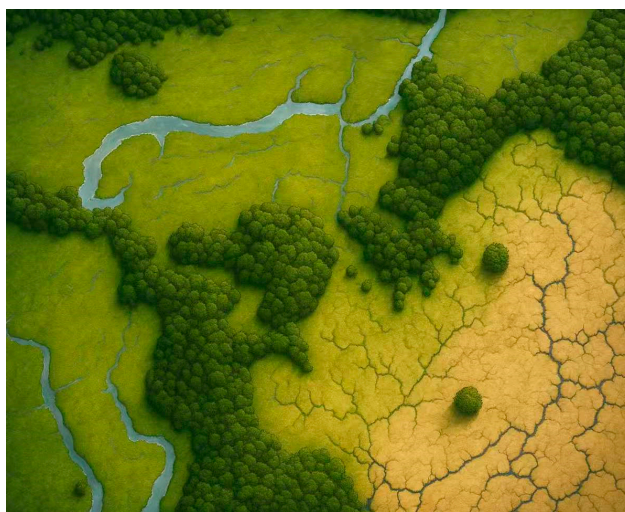


Figure 3.

The experimental validation of the unmanned aerial vehicle (UAV) target search algorithm was conducted in the simulated outdoor environment described above. A total of eight targets across five

distinct categories were deployed as subjects for target identification. Point A9B20 was designated as the takeoff and landing site for the UAV. Prior to the commencement of the experiment, no-fly zones were randomly established, as illustrated in Figure 4.

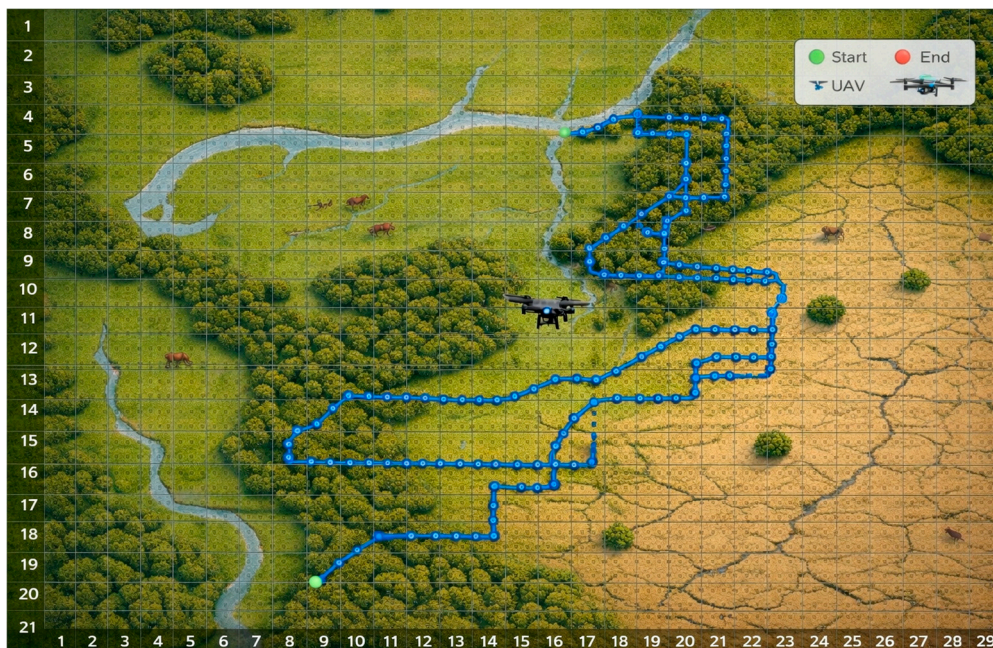


Figure 4.

4.3. Performance Evaluation

In this section, the performance of the proposed algorithm is comprehensively evaluated from three dimensions of target recognition, path planning and overall data analysis, so as to verify the effectiveness of each module and the overall framework.

4.3.1. Performance Comparison of Target Recognition Algorithms

YOLOv5 and YOLOv8 are the most mature in the existing generations of target recognition modules. Firstly, four small-volume models of YOLOv5 and YOLOv8 generations were compared, and it proved that YOLOv8n was the most suitable model for deployment on Raspberry Pi 4B. Secondly, in order to verify the effectiveness of the proposed lightweight strategy, the performance of the original YOLOv8n model and the lightweight version after channel pruning and INT8 quantization, YOLOv8N-Lite, on the Raspberry Pi 4B platform was compared on the COCO 2017 subset. At the same time, to test increased robustness effect in HSV space data, on the low illumination and strong reflective test subset the attenuation amplitude $mAP@0.5$ was analyzed statistically. The results are shown in Table 2.

Table 2. Performance comparison of target recognition module on Raspberry Pi 4B.

The model	Platform	Reasoning time	$mAP@0.5$	Amplitude of decay
YOLOv5nu	Raspberry Pi 4B	126ms	54.9%	—
YOLOv5su	Raspberry Pi 4B	380ms	68.8%	—
YOLOv8n	Raspberry Pi 4B	162ms	86.2%	11.3

YOLOv8s	Raspberry Pi 4B	400ms	70.8%	—
YOLOv8n-Lite	Raspberry Pi 4B	83ms	84.6%	4.8

The average inference delay of the original YOLOv8n model on Raspberry Pi 4B was about 162ms, and the model size was 6 MB, which reached 86.2% on the training set of this paper. After pruning and quantization, the model size was compressed to 4.2 MB, the compression ratio reached 70%, the inference delay was lowered to 83ms, representing an improvement of approximately 49% compared to the original model, mAP@0.5 was 84.6%, and the loss of accuracy was controlled within 1.6%, which met the real-time processing requirements of <100ms on the airborne platform. After the introduction of HSV data enhancement, the mAP@0.5 attenuation of the model on the complex illumination test subset was decreased from 11.3% to 4.8%, which significantly improved the adaptability in the environment of color interference.

4.3.2. Performance Comparison of Target Recognition Algorithms

In a 30m × 20m simulated field scenario, the performance of the proposed Grid-Greedy-BFS hybrid search algorithm was experimentally evaluated and compared with those of random walk, BFS, and greedy algorithms. The results are shown in Table 3.

Table 3. Performance comparison of path planning algorithms.

Algorithms	Completion time	Coverage	Replanning delay
Random walk algorithm	1.15 ms	76.2%	—
BFS	1.92 ms	99.1%	45ms
Greedy algorithm	3.06 ms	82.3%	12ms
GGB	0.31 ms	98.7%	28ms

The GGB algorithm achieved 98.7% area coverage in 0.31ms, which saved 83.86% time compared with pure BFS algorithm, and improved the coverage rate by 16.4% compared with pure greedy algorithm. The algorithm also obtained a replanning delay of 28ms, which quickly responded to dynamic target confirmation tasks, taking into account both search efficiency and traversal completeness.

5. Discussion and Concluding

In this paper, we propose a comprehensive algorithm framework integrating lightweight target recognition and hybrid path planning for the autonomous target search task of UAVs. Aiming at the problem of limited resources of airborne platform, the YOLOv8n model is compressed jointly by channel pruning and INT8 quantization method. At the same time, the HSV spatial data enhancement strategy is introduced to significantly improve the color robustness of the model under complex lighting conditions. In the aspect of path planning, a grid-greedy-BFS hybrid search algorithm is designed. Through the two-layer architecture of “global coverage + local greedy optimization”, it ensures the ability of high coverage and fast cruising, and has the ability of dynamic target replanning. The experimental results verify the effectiveness and practicability of the proposed algorithm, which can provide technical support for the autonomous operating system of UAV.

In addition, it must be pointed out that although the algorithm has excellent test data in the simulation environment, the test environment is limited to the indoor line-of-sight scene, and the influence of the performance of the algorithm on the target recognition and path planning in the actual outdoor complex terrain needs to be further verified. In view of the above limitations, we plan to further study and expand the application scenarios and generalization ability of the method in the future, so as to better meet the needs of a wide range of tasks.

Author Contributions: Conceptualization, H.D., M.L. and Z.H.; methodology, H.D., M.L. and Z.H.; software, M.L. and Z.H.; validation, H.D. and Z.H.; formal analysis, H.D.; investigation, H.D.; data curation, M.L. and Z.H.; writing—original draft preparation, H.D.; writing—review and editing, H.D. and Z.W.; visualization, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: We would like to express our sincere thanks to all the editors, reviewers and staff who participated in the review of this article.

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

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned aerial vehicles
BFS	Breadth-First Search
CCP	Coverage Path Planning
GGB	Grid-Greedy-BFS hybrid search algorithm

References

1. Chen, J.; Fang, H.; Zeng, X.-L. On the intelligent development of unmanned platforms in high-risk industries. *Sci. Sin. Inform.* 2021, *51*, 1397–1410. doi:10.1360/SSI-2021-0154.
2. Yuan, Y.; Sun, B.; Liu, G.-C. Drone-based scene matching visual geo-localization. *Acta Autom. Sin.* 2025, *51*, 287–311. doi:10.16383/j.aas.c230778.
3. Khashush, N.A.; Radif, M.J.; Abdalrdha, Z.K. A review of smart drone technologies for security surveillance and search and rescue. *J. Al-Qadisiyah Comput. Sci. Math.* 2025, *17*, 155–173. doi:10.29304/jqcs.2025.17.42549.
4. Katkuri, A.V.R.; Madan, H.; Khatri, N.; et al. Autonomous UAV navigation using deep learning-based computer vision frameworks: A systematic literature review. *Array* 2024, *23*, 100361. doi:10.1016/j.array.2024.100361.
5. Cheng, Z.K.; Yang, J.Y.; Sun, J.F.; et al. Trajectory planning of unmanned aerial vehicles in complex environments based on intelligent algorithm. *Drones* 2025, *9*, 468. doi:10.3390/drones9070468.
6. Haque, A.; Chowdhury, M.N.U.R.; Hassanalian, M. A review of classification and application of machine learning in drone technology. *ACRT* 2025, *4*. doi:10.5772/acrt.20240052.
7. Boiteau, S.; Vanegas, F.; Gonzalez, F. Dataset from: Framework for autonomous UAV navigation and target detection in global-navigation-satellite-system-denied and visually degraded environments (Version 1). Queensland University of Technology, 2024. doi:10.25912/RDF_1728355750197.
8. Civil Aviation Administration of China; National Development and Reform Commission; Ministry of Transport. Civil aviation development during the 14th Five Year Plan period exhibition planning. Available online: <https://www.gov.cn/zhengce/zhengceku/2022-01/07/5667003/files/d12ea75169374a15a742116f7082df85.pdf> (accessed on 31 December 2024).

9. Choi, U.; Lee, S. Bandwidth-aware coverage path planning for swarm of UAVs with aerial base station. In Proceedings of the 2023 International Conference on Unmanned Aircraft Systems (ICUAS), Warsaw, Poland, 6–9 June 2023; pp. 360–365. doi:10.1109/ICUAS57906.2023.10156244.
10. Kumar, P.A.; Manoj, N.; Sudheer, N.; et al. UAV swarm objectives: A critical analysis and comprehensive review. *SN Comput. Sci.* 2024, *5*, 6. doi:10.1007/s42979-024-03156-x.
11. El Ghazoual, S. Flight scope: A deep comprehensive review of aircraft detection algorithms in satellite imagery. *arXiv* 2024, arXiv:2404.02877.
12. Zhu, G.L.; Yuan, C.X.; Jiang, F. Lightweight YOLOv8 real-time object detection via progressive pruning and feature-aware knowledge distillation. In Proceedings of the 2025 IEEE 8th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 14–16 March 2025. doi:10.1109/IAEAC65194.2025.11166037.
13. Naveen, S.; Kounte, M.R. Optimized convolutional neural network at the IoT edge for image detection using pruning and quantization. *Multimed. Tools Appl.* 2025, *84*, 5435–5455. doi:10.1007/s11042-024-20523-1.
14. Altaie, U.K.; Abdelkareem, A.E.; Alhasanat, A. Lightweight optimization of YOLO models for resource-constrained devices: A comprehensive review. *Diyala J. Eng. Sci.* 2025, *18*, 1–18. doi:10.24237/djes.2025.18401.
15. Yang, D.; Solihin, M.I.; Zhao, Y. Model compression for real-time object detection using rigorous gradation pruning. *iScience* 2025, *28*, 111618. doi:10.1016/j.isci.2024.111618.
16. Laidig, R.; Shibli, F.; Tufekci, B.; et al. Improving drone communication QoS through adaptive redundancy. In Proceedings of the 2025 34th International Conference on Computer Communications and Networks (ICCCN), Tokyo, Japan, 28–31 July 2025; pp. 1–9. doi:10.1109/ICCCN65249.2025.11133801.
17. Zhou, J.; et al. Reliability-optimal UAV-assisted mobile edge computing: Joint resource allocation, data transmission scheduling and motion control. *IEEE Trans. Mob. Comput.* 2025, *24*, 4217–4234. doi:10.1109/TMC.2024.3521934.
18. Hazarika, A.; Rahmati, M. AdaptNet: Rethinking sensing and communication for a seamless internet of drones experience. *arXiv* 2024, arXiv:2405.07318.
19. Sinay, M.; Agmon, N.; Maksimov, O.; et al. Uncertainty with UAV search of multiple goal-oriented targets. *arXiv* 2022, arXiv:2203.09476. doi:10.48550/arXiv.2203.09476.
20. Zhang, J.; Du, X.; Dong, Q.; et al. Distributed collaborative complete coverage path planning based on hybrid strategy. *J. Syst. Eng. Electron.* 2024, *35*, 463–472. doi:10.23919/JSEE.2023.000118.
21. Sheltami, T.; Ahmed, G.; Ghaleb, M.; et al. UAV path planning and trajectory optimization: A comprehensive survey. *Arab. J. Sci. Eng.* 2026, *51*, 105–145. doi:10.1007/s13369-025-10971-8.
22. Rahman, M.; Sarkar, N.I.; Lutui, R. A survey on multi-UAV path planning: Classification, algorithms, open research problems, and future directions. *Drones* 2025, *9*, 263. doi:10.3390/drones9040263.
23. Chen, X.; Li, W.; Ma, J.; et al. Frontiers in cooperative control of unmanned systems in low-altitude environments (Supplement 1, 2026, 20250443). *Acta Aeronaut. Astronaut. Sin.* 2026. doi:10.7527/S1000-6893.2025.32976.
24. Mathi, S.C.; Deepa, K. SoC estimation and comparative analysis of lithium polymer and lithium-ion batteries in unmanned aerial vehicles. In Proceedings of the 2024 First International Conference on Innovations in Communications, Electrical and Computer Engineering (ICICEC), Davangere, India, 19–20 December 2024. doi:10.1109/ICICEC62498.2024.10808488.
25. Bauer, J.; Klein, A.; Bertram, S. Investigating the impact of communication delays and bandwidth restrictions on remote operations of unmanned systems. In Proceedings of the 1st International Conference on Drones and Unmanned Systems, 2025; pp. 273–279. doi:10.13140/RG.2.2.18747.94240.
26. Peng, C.; Keller, J.; Kumar, V. Time-optimal UAV trajectory planning for 3D urban structure coverage. In *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, 22–26 September 2008; pp. 2750–2757. doi:10.1109/IROS.2008.4650988.
27. Zhang, Y.; Wang, H.; Yan, Q.; et al. Research progress of unmanned mobile vision technology for complex dynamic scenes. *J. Image Graph.* 2025, *30*, 1828–1871. doi:10.11834/jig.240458.

28. Cheng, Z.; Yang, X.; Lu, R.; et al. Visible-infrared incremental domain adaptive object detection based on prototype alignment and adaptive recovery. *Infrared Laser Eng.* 2025, *54*, 20250388. doi:10.3788/IRLA20250388.
29. Song, B.; Zhao, S.; Wang, Z.; et al. daf-detr: A dynamic adaptation feature transformer for enhanced object detection in unmanned aerial vehicles. *Knowl.-Based Syst.* 2025, *323*, 113760. doi:10.1016/j.knosys.2025.113760.
30. He, W.; Hu, Y.; Li, W. Review of optimization algorithms for UAV routes. *Mod. Def. Technol.* 2024, *52*, 24–32. doi:10.3969/j.issn.1009-086x.2024.04.003.
31. Xu, S.; Zhou, Z.; Li, J.; et al. Communication-constrained UAVs' coverage search method in uncertain scenarios. *IEEE Sens. J.* 2024, *24*, 16778–16789. doi:10.1109/JSEN.2024.3384209.
32. Tang, J.; Ma, H. Mixed integer programming for time-optimal multi-robot coverage path planning with efficient heuristics. *IEEE Robot. Autom. Lett.* 2023, *8*, 6491–6498. doi:10.1109/LRA.2023.3306996
33. Chua, W.H.; Uttraphan, C.; Choon, C.C.; et al. Optimizing FPGA-based YOLO series accelerators: A survey of techniques. *Neurocomputing* 2025, *650*, 130874. doi:10.1016/j.neucom.2025.130874.
34. Ait Saadi, A.; Soukane, A.; Meraihi, Y.; et al. Intelligent path planning algorithms for UAVs: Classification, complexity analysis, hybrid ablation insights, and future directions. *Adv. Mech. Eng.* 2025, *17*, doi:10.1177/16878132251355020.

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