

Article

Advanced UAV Material Transportation and Precision Delivery Utilizing the Whale-Swarm Hybrid Algorithm (WSHA) and APCR-YOLOv8 Model

Yuchen Wu , Zhijian Wei, Huilin Liu ^{*}, Jiawei Qi, Xu Su, Jiqiang Yang and Qinglin Wu

School of Computer Science and Engineering, Anhui University of Science and Technology, Huainan 232001, China; ycwu@aust.edu.cn (Y.W.); zjwei@aust.edu.cn (Z.W.); jwqi@aust.edu.cn (J.Q.);

2021304655@aust.edu.cn (X.S.); 2021304685@aust.edu.cn (J.Y.); 2021304651@aust.edu.cn (Q.W.)

* Correspondence: liuhl@aust.edu.cn

Abstract: This paper proposes an effective material delivery algorithm to address the challenges associated with Unmanned Aerial Vehicle (UAV) material transportation and delivery, which include complex route planning, low detection precision, and hardware limitations. This novel approach integrates the Whale-Swarm Hybrid Algorithm (WSHA) with the APCR-YOLOv8 model to enhance efficiency and accuracy. For path planning, the placement paths are transformed into a Generalized Traveling Salesman Problem (GTSP) to be able to compute solutions. The Whale Optimization Algorithm (WOA) is improved for balanced global and local searches, combined with an Artificial Bee Colony (ABC) Algorithm and adaptive weight adjustment to quicken convergence and reduce path costs. For precise placement, the YOLOv8 model is first enhanced by adding the SimAM attention mechanism to the C2f module in the detection head, focusing on target features. Secondly, GhoHGNetv2 using GhostConv is the backbone of YOLOv8 to ensure accuracy while reducing model Params and FLOPs. Finally, a Lightweight Shared Convolutional Detection Head (LSCDHead) further reduces Params and FLOPs through shared convolution. Experimental results show that WSHA reduces path costs by 9.69% and narrows the gap between the best and worst paths by about 34.39%, compared to the Improved Whale Optimization Algorithm (IWOA). APCR-YOLOv8 reduces Params and FLOPs by 44.33% and 34.57%, respectively, with mAP@0.5 increasing from 88.5 to 92.4 and FPS reaching 151.3. This approach can satisfy the requirements for real-time responsiveness while effectively preventing missed, false, and duplicate detections during the inspection of emergency airdrop stations. In conclusion, combining bionic optimization algorithms and image processing significantly enhances the efficiency and precision of material placement in emergency management.



Citation: Wu, Y.; Wei, Z.; Liu, H.; Qi, J.; Su, X.; Yang, J.; Wu, Q. Advanced UAV Material Transportation and Precision Delivery Utilizing the Whale-Swarm Hybrid Algorithm (WSHA) and APCR-YOLOv8 Model. *Appl. Sci.* **2024**, *14*, 6621. <https://doi.org/10.3390/app14156621>

Academic Editor: Yutaka Ishibashi

Received: 3 June 2024

Revised: 18 July 2024

Accepted: 25 July 2024

Published: 29 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the current context of globalization, emergency response has become a critical field of study. Effective deployment of supplies not only mitigates the effects of emergencies, but also saves lives at crucial times [1,2]. This study focuses on material transportation path planning and precise delivery via unmanned aerial vehicles (UAVs), addressing critical technical issues in military emergency response to improve the efficiency and precision of emergency management. These improvements are significant in enhancing overall emergency response capabilities [3,4].

Path-planning studies are crucial for material placement in military emergency response scenarios. The main challenge is to design an efficient method to quickly reach the desired destination while navigating complex and dynamic environments. Modern drones need real-time updates to avoid obstacles, such as skyscrapers, ensuring safe and efficient delivery routes in rapidly changing military operations. Intelligent algorithms such as the Genetic Algorithm (GA) [5], the Ant Colony Optimization (ACO) Algorithm [6],

the Particle Swarm Optimization (PSO) Algorithm [7], the Artificial Bee Colony (ABC) Algorithm [8], the Whale Optimization Algorithm (WOA) [9], and other intelligent algorithms are widely used in the computation of path planning because of their strong robustness, positive-feedback mechanism, and self-organization characteristics. Zheng et al. [10] proposed a PSO Algorithm based on migration learning, making it possible to find the optimal path for the TSP problem quickly. Pehlivanoglu et al. [11] introduced a way to improve the starting population in the GA for UAV path planning in the target-coverage issue. This method accelerates the convergence of the GA and resolves collisions on the terrain surface. Yan et al. [12] introduced a WOA using forward-looking sonar to address the challenge of planning three-dimensional paths for autonomous submersibles in the intricate undersea environment. This approach considers safety, smoothness, real-time constraints, and low integrity. In [13], a two-stage optimization algorithm based on bionic robotic fish is proposed to solve the path-planning problem, with currents and moving obstacles using staged optimization. Although many research results have effectively solved 2D or 3D path-planning problems, research for the Generalized Travelling Salesman Problem (GTSP) in military contexts is still insufficient. The path-planning problem in this paper is initially transformed into a computable GTSP. Considering the shortcomings of the WOA in terms of slow convergence speed and low solution accuracy, this paper proposes a Whale-Swarm Hybrid Algorithm (WSHA) based on the WOA's improvement, which combines the fine optimization ability of the ABC Algorithm in local search with the advantages of the Improved Whale Optimization Algorithm (IWOA) in global search. In addition, the convergence process is accelerated, and the total path cost is reduced by combining the 2-Opt Algorithm and the adaptive weight adjustment strategy [14].

Machine vision target detection research is also critical to ensure the accuracy of material airdrops in military emergency relief operations. Early deep-learning image processing models were mainly networks of the RCNN family, including R-CNN [15], Faster R-CNN [16], and so on. This two-stage model tends to be slow to detect, and thus challenging to utilize in industry. In 2016, Redmom et al. proposed a single-stage detection model YOLOv1 [17], and in the same year, this proposed single-stage algorithm also contained the SSD algorithm [18]. These two algorithms' accuracy is not as good as that of a two-stage algorithm model, such as the R-CNN, but the detection speed has been sufficient to meet the need for real-time detection. Nowadays, various scholars continue to improve on deep-learning algorithms, among which the target detection models with the most usage occasions include YOLOv3 [19], YOLOv5 [20], YOLOv7 [21], YOLOv8 [22], etc., which, thanks to the scholars' research on each model, make the deep-learning algorithms make breakthroughs in terms of the speed, accuracy, and performance of each of the params. Jawaharlal Nehru et al. [23] improved the YOLO network to improve the accuracy and speed of target detection in aerial images. They solved the problems of low-positional accuracy, slow speed, target omission, and false detection of multiscale targets by clustering the target box dimensions, pre-training the network, multiscale training, and optimizing the screening rules. Souza et al. [24] proposed Hybrid-YOLO using YOLOv5x and ResNet-18 classifiers, which provides a good classification of transmission line insulator defects by UAVs. Jiang et al. [25] used YOLO to extract features from ground-based thermal infrared (TIR) images and videos and applied it to target detection in TIR videos from UAVs. Although the techniques mentioned above somewhat enhance UAV target detection performance, the complexity and specificity of the UAV viewpoint image dataset, as well as the constraints of the UAV hardware platform, make it challenging to strike a balance between detection performance and hardware resource consumption. This paper introduces the APCR-YOLOv8 model as a solution to the challenges of detecting emergency airdrop stations under varying viewpoints, lighting conditions, and the need for high real-time performance. The model combines SimAM to improve target detection accuracy and utilizes the GhoHGNetv2 lightweight convolutional network to enhance model performance while reducing Params and FLOPs. In addition, a lightweight shared convolutional detection head (LSCDHead) is

proposed for single-target detection tasks to reduce the parameters and FLOPs further and meet the real-time requirements.

To better understand the methodologies used in this paper, it is essential to distinguish between artificial intelligence (AI) and metaheuristic algorithms (MHA). AI involves the simulation of human intelligence processes by machines and includes techniques such as machine learning and neural networks, which are often used for tasks like image recognition and decision-making. Conversely, MHA is a strategy designed to find suitable enough solutions for optimization problems by combining different heuristic methods, such as the GAs, PSO Algorithm, and WOA.

In this paper, we leverage the strengths of both AI and MHA by integrating the WSHA with the APCR-YOLOv8 model. This combination achieves seamless integration of path planning and target detection to optimize UAV logistics and ensure precise delivery of supplies. The innovations of this paper are mainly in two aspects: (1) in path planning; by defining the problem as a GTSP, improving the spiral position update and convergence factor of the WOA, and combining the ABC Algorithm with the IWOA, we can efficiently solve the GTSP, significantly improving the path-planning efficiency and reducing the total flight distance. (2) In target detection; the proposed APCR-YOLOv8 model introduces the SimAM attention mechanism, incorporating the GhoHGNetv2 network and the lightweight detection head LSCDHead. This improves detection accuracy and reduces the computational load, making the YOLOv8 model more suitable for real-time applications on UAV hardware platforms.

The remaining sections of this paper are organized in terms of content as follows: Section 2 describes in detail the GTSP model and the solution method for material drop; Section 3 presents the design of the APCR-YOLOv8 model and the application of the target-detection technique to identify the emergency airdrop stations with precise identification accurately; Section 4 verifies the effectiveness of the proposed method through experiments; and Section 5 summarizes the results of this research. To improve the paper's readability, we have summarized all abbreviations in Table 1, which is included as a nomenclature table.

Table 1. Nomenclature table.

Abbreviation	Full Form
UAV	Unmanned Aerial Vehicle
GTSP	Generalized Traveling Salesman Problem
WOA	Whale Optimization Algorithm
IWOA	Improved Whale Optimization Algorithm
ABC	Artificial Bee Colony
WSHA	Whale-Swarm Hybrid Algorithm
APCR-YOLOv8	Attention-based and Computational Reduction YOLOv8
SimAM	Simple Attention Module
GhoHGNetv2	Ghost HGNetv2
LSCDHead	Lightweight Shared Convolutional Detection Head

2. GTSP Solution for UAV Material Drop Paths

2.1. GTSP Model Construction and Description

This paper converted the actual material path-planning problem into a GTSP that can be solved computationally, aiming to optimize the material delivery path to minimize the total distance. Emergency airdrop stations are considered critical points for material delivery, and the two-dimensional coordinates of each point represent the actual location of a specific airdrop station. This paper addressed the GTSP by grouping these airdrop stations, each representing a collection of airdrop stations for a particular region. This approach can be utilized to optimize the cruising route of UAVs, ensuring rapid access to all critical groups in emergencies for effective material airdropping. This abstraction allows for the application of efficient algorithms to find the shortest airdrop path that covers all groups. The GTSP problem is represented by defining a weighted graph $G = (V, E, W)$. V

is a set of vertices, and each vertex v_i represents an emergency airdrop station. E is a set of edges, representing possible connections between drop stations, and the edge e_{ij} represents a path from the drop station v_i to v_j . w is a set of weights, representing the weight of a path from v_i to v_j and $w_{ii} = 0$ for all i . In the GTSP of this paper, the drop stations were divided into a total of m groups. The goal is to find the shortest closed path that contains precisely one airdrop station in each group, which ensures that each group is dropped at least once while the total material drop path is as short as possible. The path can be described by a sequence of vertices $(v_{p_1}, v_{p_2}, \dots, v_{p_m})$, where $v_{p_i} \in V_i$, each V_i is a collection of material airdrop stations.

This paper aimed to optimize path selection by minimizing the overall weight of the pathways as represented by the following equation:

$$\text{MinD} = \sum_{i=1}^{m-1} w_{v_{p_i} v_{p_{i+1}}} + w_{v_{p_m} v_{p_1}} \quad (1)$$

where m is the total number of vertices in the vertex set V and v_{p_i} is the index of the i th vertex in the path.

2.2. Improved Whale Optimization Algorithm (IWOA) for Solving GTSP

2.2.1. Principles and Applications of the WOA

The WOA mimics the feeding behavior of humpback whales by iteratively updating their positions to improve spatial solutions [26]. This section is divided into three parts to elaborate on the WOA solution for the material placement GTSP.

Step 1: Encircling the prey (positioning of material drop-off points). In this step, the algorithm considers the optimal path as the “prey.” In a d -dimensional space, let the position of the current best whale be $X^* = (X_1^*, X_2^*, \dots, X_d^*)$, and the position of a random whale be $X^j = (X_1^j, X_2^j, \dots, X_d^j)$. Then, the latest position coordinates of the individual whale X^j are constantly updated $X^{j_new} = (X^{j_new1}, X^{j_new2}, \dots, X^{j_newd})$ as they approach the optimal whale individual X^* as described in the following equation:

$$X_k^{j_new}(t+1) = X_k^*(t) - A_1 \cdot D_k \quad (2)$$

$$D_k = |C_1 \cdot X_k^*(t) - X_k^j(t)| \quad (3)$$

where $X_k^{j_new}$ is the new position of the j th material drop path at the k th drop point; X_k^* is the position of the optimal path at the k th drop point; and t is the current iteration number. D_k is the distance between the k th whale individual and the optimal whale individual in the k th dimension, representing the difference between the two material placement sites. A_1 is the coefficient that controls the magnitude of the path update, and C_1 enhances the randomness of the path search. C_1 is determined by the random number r_2 and calculated by Equation (4):

$$\begin{cases} A_1 = 2\alpha \cdot r_1 - \alpha \\ C_1 = 2r_2 \end{cases} \quad (4)$$

where both r_1 and r_2 are taken randomly from among $[0, 1]$ and α is a linear convergence factor, converging linearly from $2 \sim 0$.

Step 2: Bubble net attack (Spiral Path Optimization). In this step, the study employs a spiral search strategy to refine and optimize material drop patterns in a path-planning optimization problem. The algorithm alternates between linear and spiral updates to explore the solution space flexibly and find shorter or less costly material drop paths. Its position change rule is as follows:

$$X_k^{j_new}(t+1) = X_k^*(t) + D_k \cdot e^{bl} \cdot \cos(2\pi l) \quad (5)$$

$$D_k = |X_k^*(t) - X_k^j(t)| \quad (6)$$

where $e^{bl} \cos(2\pi l)$ determines the spiral path's shape and direction to simulate the material delivery strategy's spiral adjustment; b is the spiral shape parameter that influences the whale's position update; and l is randomized across the range $[-1, 1]$.

The probability of both the whale's contraction encirclement behavior and the selection of a helical swimming update position was set to allow the whale to contract encirclement while swimming towards its prey, as shown in Equation (7):

$$X_k^{j_new}(t+1) = \begin{cases} X_k^*(t) - A_1 \cdot D_k & p < 0.5 \\ X_k^*(t) + D_k \cdot e^{bl} \cdot \cos(2\pi l) & p \geq 0.5 \end{cases} \quad (7)$$

where p is the probability.

Step 3: Search predation (global search path update). In this step, mutual information sharing and position updates help the algorithm to escape local optima and explore a more comprehensive solution space to find the optimal material delivery solution, as calculated by Equation (8):

$$\begin{cases} X_k^{j_new}(t+1) = X_k^{rand} - D_k \cdot A_1 \\ D_k = |C_1 \cdot X_k^{rand} - X_k^j(t)| \\ C_1 = 2r_2 \\ A_1 = 2\alpha \cdot r_1 - \alpha \end{cases} \quad (8)$$

where X_k^{rand} is the k th component of the position vector of a random whale individual in the solution space.

Whales can further decide whether to swim or search for prey locations based on information from other individuals in the solution space. This individual-to-individual interaction strategy enhances the algorithm's global retrieval capabilities.

Throughout the WOA, individual whales are forced to move away from the target prey when $|A_1| \geq 1$ and select other whales in the solution space to update their current position coordinates until a global positional optimal solution in the current space is found.

2.2.2. Improvements to the Whale Position Update Mechanism

Equation (8) demonstrates that the WOA utilizes the current position of a randomly selected individual or an optimally positioned individual as the navigation target to maintain population diversity while updating an individual whale's position. This is achieved by gradually updating the position through a linear decrease of the convergence factor as the number of iterations increases, leading the whales to search for encirclement during the feeding process.

The advantage of adopting this type of position updating method is that the whales gradually narrow the searched solution space while searching for food, which efficiently improves the local retrieval ability of the WOA. However, the disadvantages lie in the fact that the group is too centralized, the value of A_1 is unstable, and the lack of the navigation coordinates of the good individuals makes it easy for the whales to deviate from the optimal direction of foraging, which affect the convergence speed and reduce the global proficiency of the algorithm.

This work proposes a revised spiral position updating algorithm. It sorts the fitness of individual whales in ascending order, yielding the following results:

$$X^1, \dots, X^{i-1}, X^i, \dots, X^{popSize} \quad (9)$$

Second, whale individuals were categorized into three classes based on fitness: optimal, intermediate, and deviant individuals, and the number of individuals in each class was $0.05 \times popSize$, $0.9 \times popSize$, and $0.05 \times popSize$. Individuals in each of the three classes were denoted as X^{p_best} , X^m and X^{p_worst} .

The modified position update formula is as follows:

$$X_k^{j+1}(t+1) = X_k^{p-best}(t) + D_k \cdot e^{bl} \cdot \cos(2\pi l) \quad (10)$$

$$D_k = X_k^{p-best}(t) - X_k^{p-worst}(t) + |X_k^m(t) - X_k^j(t)| \quad (11)$$

where $X_k^{p-best}(t)$ is the position of the optimal individual in the k th dimension in the current iteration and $X_k^{p-worst}(t)$ is the position of the worst individual in the k th dimension in the current iteration. $X_k^m(t)$ is the position in the k th dimension of an intermediate-mass individual randomly selected in the current iteration. $X_k^j(t)$ is the position of the j th whale individual examined in the current iteration in the k th dimension. D_k is the final distance metric, reflecting the difference in relative position between the current individual and the optimal-, worst-, and intermediate-quality individuals.

In the standard WOA, the spiral shape parameter b is usually set as a constant, leading to a lack of diversity in the whales' postures during search and prey, often resulting in overly simplistic position update methods and making the algorithm prone to premature convergence. To improve the fixed posture search during the predation process, the enhanced WOA selects the parameter b as a dynamically changing sine function value, which varies with the number of iterations, allowing the whale individuals to adjust their spiral postures with the global iteration changes while searching and preying. The sine function value b effectively prevents the WOA from falling into local convergence and improves the global search capability of the algorithm, enhancing the convergence precision when seeking the optimal solution to the GTSP.

The improved parameter b is shown as follows:

$$b = -\lambda \sin\left(\omega\pi \cdot \sqrt{\frac{t_{\max} - t}{t_{\max}}}\right) \quad (12)$$

where λ is the spiral update coefficient, setting $\lambda = 10$, ω is the attitude influence constant, $\omega = 0.5$, and t_{\max} is the maximum number of iterations.

2.2.3. Adopting a Nonlinear Decreasing Convergence Factor

Parameter optimization in bionic optimization algorithms has an essential enhancement for convergence speed and searching ability; by aimlessly and randomly taking values, intelligent algorithms cannot be optimized for more efficient performance. An enormous value of the linear convergence factor α should be used to improve the search performance when the WOA initially enters the iteration. However, as the iteration proceeds to the end, the linear convergence factor α should be chosen to be smaller than the initial iteration to improve the local contraction ability of the WOA.

The original linearly decreasing convergence factor α is now improved to a nonlinear decrease in the form of an exponential function-based contraction decrease of the form and is expressed as follows:

$$\alpha(t) = -0.5 + 2.5 \times e^{t\beta_\alpha \times \ln(\frac{0.5}{2.5})} \quad (13)$$

where β_α is a constant and is an optimization factor for $\alpha(t)$ based on the nonlinear decreasing exponential function.

2.3. Based on a Hybrid Whale-Swarm Solution

Although the above IWOA has made progress in global search capability, it still suffers from the problem of insufficient local search refinement. When dealing with complex optimization problems, it is difficult for a single algorithm to balance global and local searches. Therefore, in this paper, a Whale-Swarm Hybrid Algorithm is explicitly proposed to solve path planning in the GTSP. The hybrid algorithm combines the careful optimization ability of the ABC algorithm in local search and the extensive exploration

ability of the IWOA in global search. With this combination, the algorithm can quickly identify potential high-quality solution regions and perform fine-grained optimization and refinement, leading to efficient exploration and exploitation of the solution space. This approach offers the advantages of balancing global and local search, enhancing adaptability, accelerating the convergence process, and providing a new solution strategy for complex optimization problems.

Step 1: Initialization variables. Initialize the set of paths $X = \{X_1, X_2, \dots, X_N\}$. Each path X_i represents a possible path for material placement. Initialize each parameter of the ABC algorithm and IWOA algorithm. Initialize the weight $w_{IWOA} = w_{ABC} = 0.5$ of the ABC Algorithm and IWOA strategies.

Step 2: Division of population size. The overall population size is divided based on weights through Equation (14):

$$\begin{cases} N_{IWOA} = \lceil N \times w_{IWOA} \rceil \\ N_{ABC} = N - N_{IWOA} \end{cases} \quad (14)$$

where $\lceil \cdot \rceil$ denotes an upward rounding function that rounds numbers to the nearest integer, making sure that N_{IWOA} and N_{ABC} are integers and that the entire population is distributed between the two strategies without overlap.

Step 3: Phase I—Global exploration phase of the IWOA algorithm. For each solution X_i , update its position is as follows:

$$\begin{cases} X_{i_new}(t+1) = X_{best}(t) + D_k \cdot e^{bl} \cdot \cos(2\pi l) \\ D = X_{best}(t) - X_{worst}(t) + |X_m(t) - X_k(t)| \end{cases} \quad (15)$$

where the dynamically changing spiral shape parameter b is adapted to be calculated according to Equation (12), which makes the postures of individual whales more diversified during the search and predation process; meanwhile, the nonlinearly decreasing convergence factor α is calculated according to Equation (13), which enhances the search performance in the initial iteration stage of the WOA while improving the local contraction ability of the algorithm and further optimizing the global search performance.

Step 4: Phase II—Localized search phase of the ABC algorithm. For the ABC algorithm, update the path formula as follows:

$$Y_i = X_i + \phi \cdot (X_i - X_k) \quad (16)$$

where X_i represents the current path, Y_i represents the updated path, X_k is other paths in the same category, and ϕ is a random number in the interval $[-1, 1]$ that controls the update magnitude.

Calculate the fitness value of the current solution as follows:

$$f(X) = \sum_{i=1}^{n-1} d(X_i, X_{i+1}) + d(X_n, X_1) \quad (17)$$

Based on the fitness value of the current solution, the list of quality paths is updated. Paths with low fitness values will be eliminated, and paths with high fitness values will be retained and optimized for the next iteration.

Step 5: Perform 2-OPT optimization every t rounds. Let a material placement path be denoted as a vertex sequence V_1, V_2, \dots, V_n , where V_i denotes the i th material placement point. The optimization is performed using the following steps: (1) Choose two paths to exchange: two pairs of vertices (V_a, V_{a+1}) and (V_b, V_{b+1}) of which $1 \leq a < b < n$ are chosen at random from the paths. (2) Calculate the difference in path lengths before and after optimization: let $L_{original}$ be the length of the original path and L_{new} be the length of the new path after performing the swap. The new path is formed by disconnecting the two selected pairs of vertices and reconnecting V_a to V_b and V_{a+1} to V_{b+1} . The new path is the length of the original path. (3) Determine whether to accept the new path: if the total

length L_{new} of the new path is less than the total length $L_{original}$ of the original path, the new path is accepted. The calculation of path length before and after optimization is expressed by the following formula:

$$L_{original} = \sum_{i=1}^{n-1} d(V_i, V_{i+1}) + d(V_n, V_1) \quad (18)$$

$$L_{new} = L_{original} - (d(V_a, V_{a+1}) + d(V_b, V_{b+1})) + (d(V_a, V_b) + d(V_{a+1}, V_{b+1})) \quad (19)$$

where $d(V_i, V_j)$ denotes the distance between vertices V_i and V_j .

Step 6: Dynamically adjust the weights of the IWOA and ABC Algorithm strategies. The adjustment factor is set to $\lambda = 0.1$. The IWOA has a solid global search capability; when $fit_{best}(X_i)(t+1) < fit_{best}(X_i)(t)$ this indicates that the global search is working well, the weight w_{IWOA} is augmented, and the formula is calculated as follows:

$$\begin{cases} w_{ABC} = w_{ABC} - \lambda \times w_{ABC} \\ w_{IWOA} = w_{IWOA} + \lambda \times w_{IWOA} \end{cases} \quad (20)$$

Conversely when $fit_{best}(X_i)(t+1) > fit_{best}(X_i)(t)$, the formula is as follows:

$$\begin{cases} w_{ABC} = w_{ABC} + \lambda \times w_{ABC} \\ w_{IWOA} = w_{IWOA} - \lambda \times w_{IWOA} \end{cases} \quad (21)$$

Additionally, ensure that the following equation holds:

$$\begin{cases} w_{IWOA} + w_{ABC} = 1 \\ 0.1 \leq w_{IWOA} \leq 0.9 \\ 0.1 \leq w_{ABC} \leq 0.9 \end{cases} \quad (22)$$

Step 7: Return to Step 2 to resume execution if the maximum number of iterations has not been reached. Once the maximum number of iterations is reached, the hybrid algorithm will showcase the ideal method for addressing the GTSP.

Through the above process, the proposed Whale-Swarm Hybrid Algorithm can achieve a good balance between the global and local search and improve the performance of solving GTSPs. This hybrid algorithm can be effectively applied to various complex optimization problems, providing an efficient and reliable solution for engineering practice. The solution flowchart is shown in Figure 1. The pseudocode of the Whale-Swarm Hybrid Algorithm is shown in Algorithm 1.

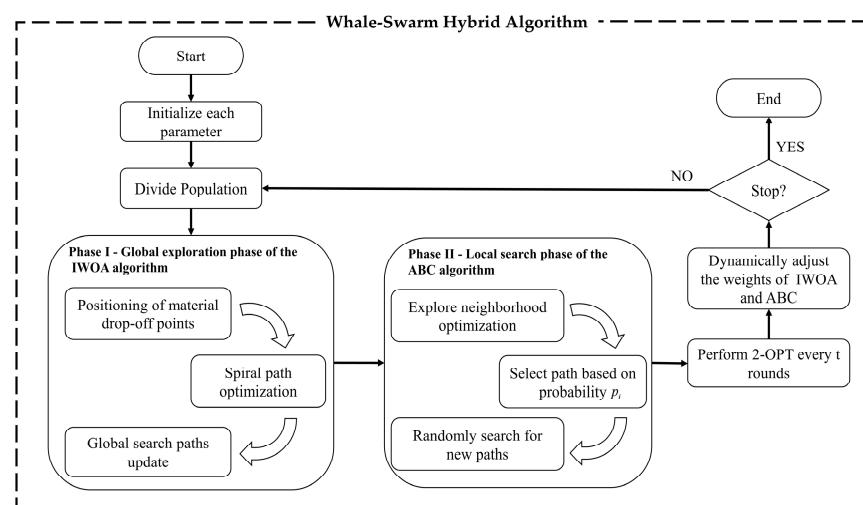


Figure 1. Flowchart of Whale-Swarm Hybrid Algorithm.

Algorithm 1: Whale-Swarm Hybrid Algorithm for solving GTSP

Input: Population size N , Number of cities, Distance matrix, City classes, Maximum iterations $MaxIter$, Optimization interval $OptInterval$.

Output: Best found path and its length.

```

1: Initialize: Path set  $X = \{X_1, X_2, \dots, X_N\}$ ,  $X_i$  represents a potential delivery path;
   weights for IWOA and ABC strategies:  $w_{IWOA} = w_{ABC} = 0.5$ .
2: Divide Population: Based on weights:  $N_{IWOA} = \lceil N \times w_{IWOA} \rceil$ ,  $N_{ABC} = N - N_{IWOA}$ .
3: for  $iter = 1$  to  $MaxIter$  do
4:   Phase 1 — Global Search with IWOA:
5:   for  $i = 1$  to  $N_{IWOA}$  do
6:     Update paths using IWOA.
7:   end for
8:   Phase 2 — Local Search with ABC:
9:   for  $i = N_{IWOA} + 1$  to  $N$  do
10:    Update paths using ABC.
11:   end for
12:   if  $iter \bmod OptInterval = 0$  then
13:     Selective Optimization — 2-OPT Technique:
14:     for  $i = 1$  to  $N$  do
15:       Optimize each path  $X_i$  using the 2-OPT algorithm.
16:     end for
17:   end if
18:   Adjust Weights: Based on the performance in the current iteration.
19:   Dynamically update  $w_{IWOA}$  and  $w_{ABC}$ .
20:   Re-divide the population based on updated weights.
21:   Record Current Best: Update if a better solution is found.
22: end for
```

3. UAV-Based Targeting for Emergency Material Delivery

In emergency material delivery, UAV path planning and target detection tasks are interdependent and constitute an efficient material delivery system. UAV path optimization aims to shorten flight distances to reduce energy consumption and ensure that the UAV reaches each predefined emergency drop station quickly and accurately. At the same time, target detection technology helps the UAV identify and locate drop points to ensure that supplies are delivered accurately. The collaboration of these two tasks is crucial for enhancing the effectiveness and precision of emergency supplies.

3.1. APCR-YOLOv8: Attention-Based and Parameter and Computational Reduction YOLOv8

This paper intended to apply the YOLOv8 model to the detection task of an emergency airdrop station. However, in practice, the significant difference in viewing angle between the area on top of a tall building and the conventional ground view may lead to significant changes in the target object's shape, size, and appearance. In addition, the lighting conditions at the top of a tall building can be highly variable, and different lighting conditions may significantly impact the target's visibility and detection performance. Although the YOLOv8 model has demonstrated excellent performance in several detection tasks, its direct application to the detection of emergency airdrop stations faces unique challenges posed by differences in viewing angles, variable lighting conditions, and the need for high real-time performance. These factors may affect the performance of the YOLOv8 model in such specific scenarios.

To optimize the performance of the emergency airdrop station detection task, this paper proposed the APCR-YOLOv8 model, which aims to adapt more accurately to the requirements of specific application scenarios. Its structure is shown in Figure 2. First, a parameter-free attention mechanism, SimAM, is introduced to the C2f module at the front end of the detection head; this mechanism focuses on effectively refining the critical information and enhancing the detection capability of the target without increasing the overall parameters of the model. Then, to optimize the model's core architecture, the

lightweight network GhoHGNetV2 is proposed as the backbone of the model, replacing the traditional backbone structure. This new backbone reduces both Params and FLOPs and improves detection accuracy while maintaining efficient feature extraction. Finally, for the single-target detection task, this paper proposed LSCDHead, a lightweight detection head suitable for single-target detection, which first employs a shared convolutional ShConv to capture the target features and then utilizes these features to perform the final task through different network branches. This shared convolutional layer strategy not only makes the model more efficient in terms of resources and computation, but also minimizes the loss of accuracy while reducing Params and FLOPs, further improving the model's practicality and flexibility.

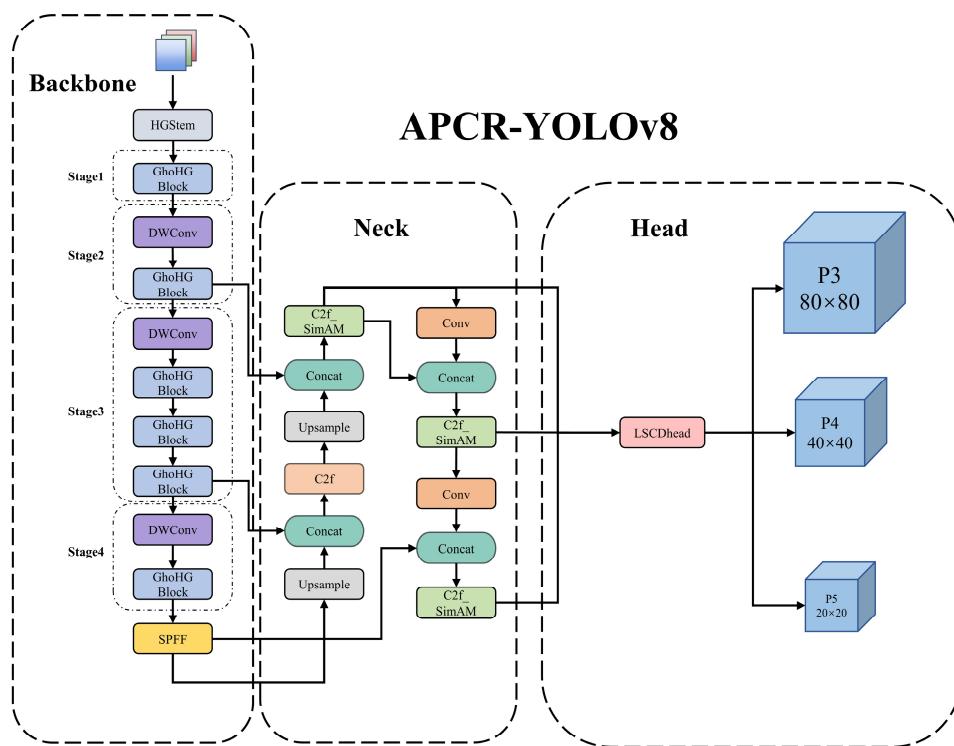


Figure 2. APCR-YOLOv8 network structure diagram.

3.2. SimAM: Simple Attention Module

This paper used a parameter-free attention mechanism named SimAM to improve the model's ability to resist interference [27]. The SimAM attention mechanism is grounded in the theory of visual neuroscience, which states that neurons with a high amount of information are more noticeable than other neurons and suppress the activity of neighboring neurons spatially. SimAM gives greater importance to neurons that convey more crucial information during activities connected to vision. The network neurons extract and enhance essential target elements while pinpointing the location of the emergency airdrop station inside a scene. The energy function representing the lowest state for the i th neuron within the SimAM attention mechanism is as follows:

$$e_t^* = \frac{4(\hat{\sigma}^2 + \lambda)}{(t - \hat{\mu})^2 + 2\hat{\sigma}^2 + 2\lambda} \quad (23)$$

where t denotes the target neuron within the input feature map's singular channel, with λ representing the canonical term, which is equivalent to the average of all neurons within that channel. $\hat{\mu} = \frac{1}{M} \sum_{i=1}^M x_i$ is defined as the mean across all neurons in a singular channel, while $\hat{\sigma}^2 = \frac{1}{M} \sum_{i=1}^M (x_i - \hat{\mu})^2$ signifies the variance among all neurons in the same channel, where M stands for the total count of neurons per channel, $M = H \times W$; i symbolizes the

index value of a particular neuron; and x_i refers to the remaining neurons within the input feature map's single channel.

Equation (23) demonstrates that when energy decreases, the target neurons in the feature map of the emergency airdrop station become more distinguishable from the surrounding neurons, which often contain more valuable information and are more significant. The ultimate output feature map is as follows:

$$\tilde{X} = \text{sigmoid}\left(\frac{1}{E}\right) \odot X \quad (24)$$

where X denotes the input features and E denotes all channel and spatial dimensions features.

Figure 3 displays the comprehensive architecture of SimAM in this research. The input feature maps undergo processing via the SimAM attention mechanism, followed by normalization of the weights using the Sigmoid function. The normalized weights are multiplied element-wise with the original feature maps' features to get the final output feature maps. The addition of the SimAM attention mechanism allows the model to concentrate more precisely on the topic target, enhancing the detection performance.

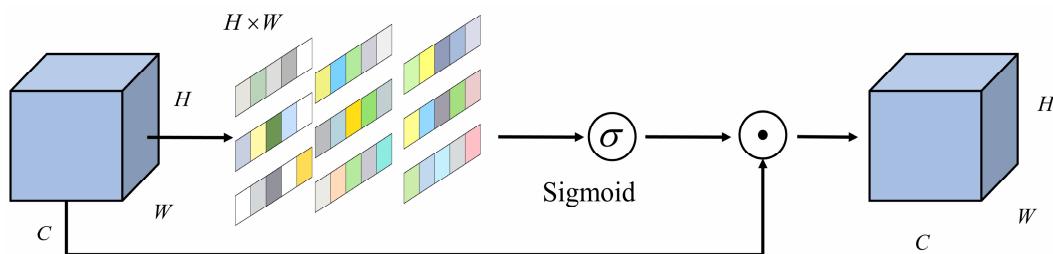


Figure 3. SimAM Attention Mechanism.

The C2f module is crucial in the target detection operation as it is responsible for obtaining valuable feature information from the input image. The C2f module faces challenges in effectively extracting all target elements from an image because of the targets' diverse sizes, shapes, and placements. This paper introduced the SimAM attention technique to strengthen the C2f module in the detecting head, improving its capacity to extract target characteristics. The structure is shown in Figure 4.

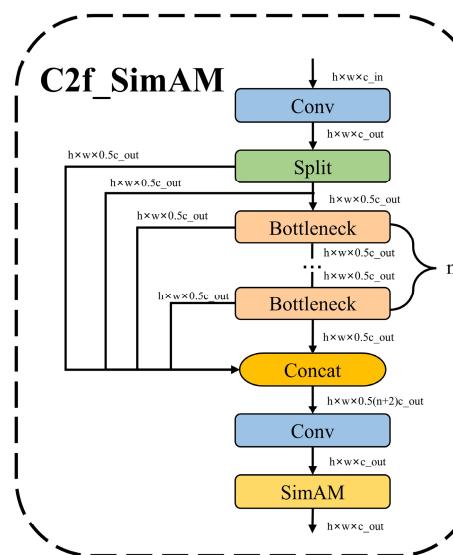


Figure 4. C2f_SimAM: C2f module with the addition of the SimAM attention mechanism.

3.3. GhoHGNetV2: Ghost HGNetv2

3.3.1. GhostConv Technology

The GhostNet network effectively compresses the network structure and FLOPs while maintaining accuracy by adopting a lightweight design concept, which mainly consists of a stack of Ghostconv modules that not only performs convolutional stacking, but also generates Ghost features and fuses them to form the modular outputs through linear operations [28]. The GhostConv schematic diagram is shown in Figure 5.

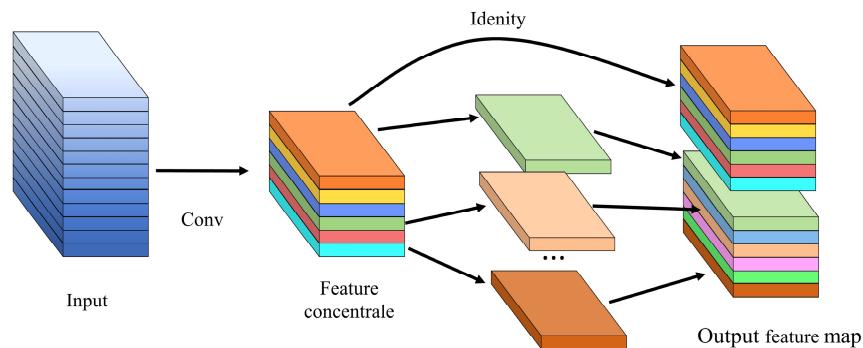


Figure 5. GhostConv Schematic.

Compared to conventional convolutional methods that output all feature maps directly, Ghostconv first performs a convolution operation to generate fewer feature maps. Subsequently, a convolutional transformation is performed on top of the generated feature maps to output a constant map and other feature maps. This strategy aims to reduce FLOPs and Params effectively, and its FLOPs vs. Params comparison is shown in the following equations:

$$r_{speed} = \frac{n \times h' \times w' \times c \times k \times k}{\frac{n}{s} \times h' \times w' \times c \times k \times k + (s-1) \times \frac{n}{s} \times h' \times w' \times d \times d} \approx \frac{S \times c}{S + c - 1} \approx S \quad (25)$$

$$r_{parameter} = \frac{n \times c \times k \times k}{\frac{n}{s} \times c \times k \times k + (s-1) \frac{n}{s} \times d \times d} \approx \frac{s \times c}{s + c - 1} \approx S \quad (26)$$

where c , h , and w are the channel, graph height, and graph width of the input data, n, h', w' is the depth, height, and width of the output channel after one convolution, k is the size of the convolution kernel, d is the size of the convolution kernel used for linear variation, and s is the number of transformations.

As can be seen from the formula, the GhostConv decreases very significantly compared to the FLOPs and Params of the traditional convolution. By replacing the Conv of HGBlock in the structure of HGNetv2, it can effectively reduce the network's number of operations and realize the network's lightweight design.

3.3.2. HGNetV2-Based Network Architecture Optimization

HGNetv2 is a SOTA target detection network RT-DETR for the backbone proposed during CVPR 2024 [29]. The performance of HGNetv2 is improved explicitly in this paper. Adopting the more lightweight GhostConv to optimize the HGNetV2 network aims to achieve higher accuracy and better model lightweight effects.

This paper substituted the CSPDarkNet-53 backbone in YOLOv8 with GhoHGNetv2, which consists of four primary modules: DWConv, HGStem, GhoHGBlock, and SPFF. Figure 6 displays the configuration of HGStem and GhoHGBlock.

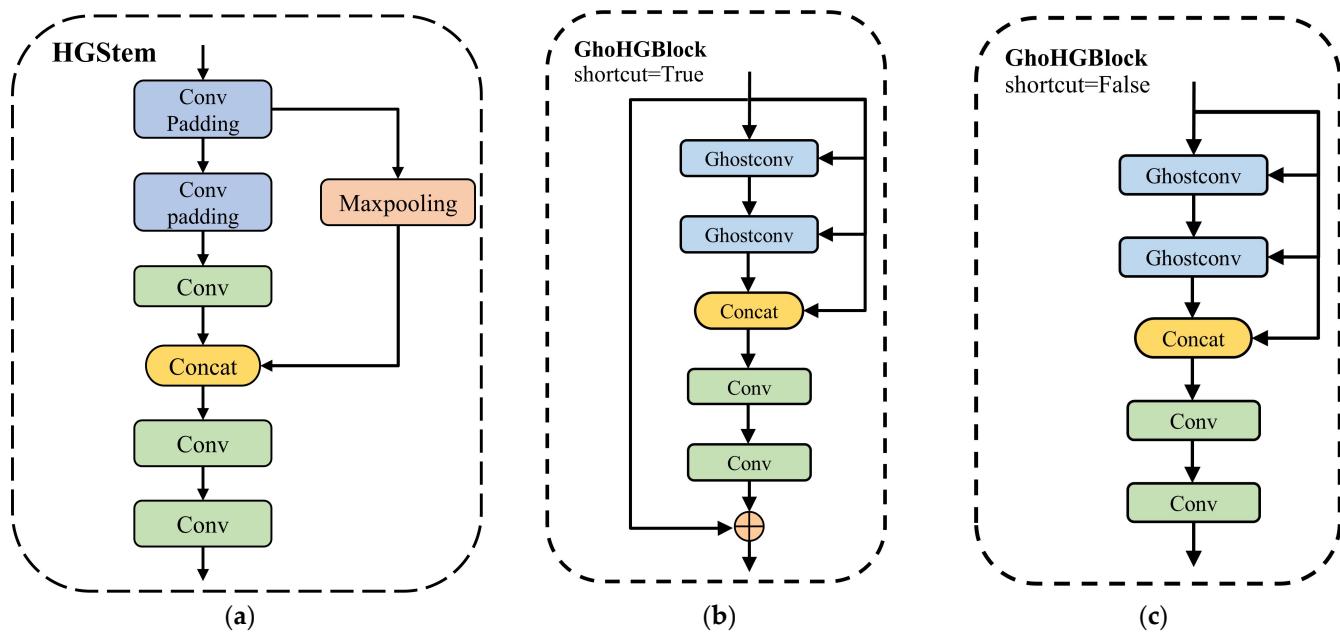


Figure 6. GhoHGNetv2 internal module structure. (a) HGStem module structure; (b) GhoHGBlock module structure (shortcut = True); (c) GhoHGBlock module structure (shortcut = False).

This improvement makes GhoHGNetv2 more suitable for application scenarios with stringent requirements on model size and computational efficiency. With these optimizations, GhoHGNetv2 increases mAP while FLOPs and Params are reduced in the backbone network.

3.4. LSCDHead: Lightweight Shared Convolutional Detection Head

The detection header in YOLOv8 contains two branches, localization (bounding-box regression) and classification, each of which typically has separate convolutional layers to extract features, leading to an increase in model Params and FLOPs. This design is not optimal for single-target detection tasks because both localization and classification tasks aim to identify the same class of objects and do not require independent feature learning paths. The classification branch only learns to recognize whether a target exists rather than distinguishing between different classes. Therefore, merging the two branches can reduce the Params and FLOPs and improve the model efficiency in single-target detection scenarios. Therefore, in this paper, the following measures were carried out to propose a LSCDHead:

- (1) Replace BatchNorm (BN) with GroupNorm (GN): GN has been shown to improve the detection and classification performance in FCOS [30]. The GN feature channel enhances the model's adaptability to targets of different scales and sizes. Replacing BN with GN improves the detection and enhances the model's adaptability to small batches of data. In this paper, GnConv is designed to replace the BN of Conv with GN to better extract P3, P4, and P5 features.
- (2) Propose Shared Convolution (ShConv): based on GnConv, ShConv is proposed to capture basic features to accomplish classification or localization tasks through different network branches. The shared convolution strategy improves model efficiency and performance by allowing the model to learn once and reuse standard features on multiple tasks.
- (3) Use scale layer: Considering the possible lack of size optimization due to shared convolution, this paper introduced a scale adjustment layer to dynamically adjust the scale of feature maps for targets of different sizes. This enhances the model's ability to adapt to targets of different sizes while reducing Params.

Combining the above improvements, the redesigned detection head in this paper achieved fewer Params and FLOPs, minimizing the loss of mAP as much as possible. The structure of GnConv and ShConv is shown in Figure 7.

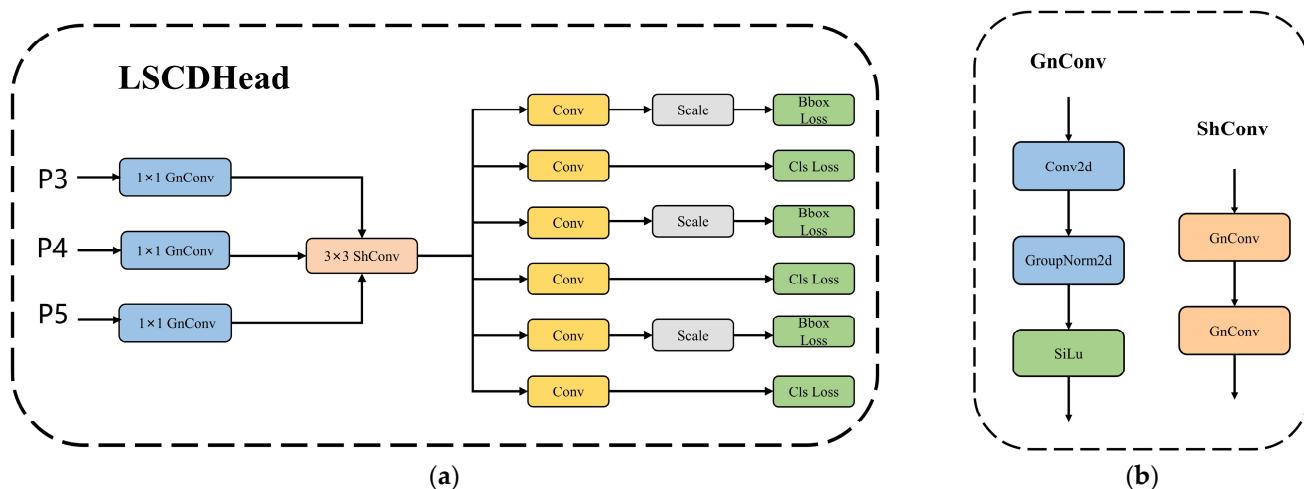


Figure 7. LSCDHead structure and internal modules. (a) LSCDHead structure; (b) Internal structure of GnConv and ShConv.

4. Analysis of Experimental Results

4.1. GTSP-Based Algorithm Performance Evaluation

This paper used Python on a PC platform with an Intel i5-13500H CPU processor and 32 GB of RAM to assess the suggested algorithm's effectiveness and performance. The approach is used for four datasets: City (17,11), (24,15), (31,16), and (39,25). This document calculated the distance between any two cities using Euclidean distance, rounding the results to two decimal places.

4.1.1. Algorithm Parameter Configuration

This paper explored multiple parameter combinations to achieve a compromise between optimal performance and computing economy across different scenarios. Table 2 displays the parameters selected for the GA, the IWOA, the ABC Algorithm, and the WSHA. The table's first column displays the parameter names, the second column shows the GA parameter values, the third column lists the IWOA parameter values, the fourth column presents the ABC Algorithm parameter values, and the fifth column showcases the WSHA parameter values.

Table 2. Parameterization of the algorithm.

Parameter	GA	IWOA	ABC	WSHA
Population size	50	50	50	50
Crossover rate	0.2	N/A	N/A	N/A
Mutation rate	0.8	N/A	N/A	N/A
w_{IWOA}	N/A	N/A	N/A	0.5
w_{ABC}	N/A	N/A	N/A	0.5

4.1.2. Performance Comparison and Analysis

The GA, IWOA and ABC Algorithm are implemented on four GTSP benchmark instances in this section. The analysis will consider the algorithms' best solution cost, worst solution cost, average solution cost, and the difference between known best and worst solutions. Each method was executed 20 times for every instance, yielding the best, worst, and average solutions. In these tables, best indicates the lowest cost. Worst indicates the

highest cost. The average is the average of the results of all runs. The difference is the comparison between the best and the worst favorable solution. The algorithm with lower best, worst, average, and difference values is considered the preferred algorithm. The results are shown in Tables 3–6.

Table 3. City (17,11) experimental results.

Algorithm	City Number	Goods Number	Iteration	Best	Worst	Average	Difference
GA	17	11	150	313.49	332.52	322.19	19.03
IWOA	17	11	150	303.86	324.25	314.29.	20.39
ABC	17	11	150	289.02	305.49	295.81	16.47
WSHA	17	11	150	274.02	288.81	280.41	14.79

Note: Bold values indicate the best performance in the algorithm. This note applies to all subsequent tables.

Table 4. City (24,15) experimental results.

Algorithm	City Number	Goods Number	Iteration	Best	Worst	Average	Difference
GA	24	15	450	369.82	392.34	380.12	22.52
IWOA	24	15	450	366.12	389.71	375.38.	23.59
ABC	24	15	450	356.64	385.49	369.37	28.85
WSHA	24	15	450	334.48	349.18	343.81	14.7

Table 5. City (31,16) experimental results.

Algorithm	City Number	Goods Number	Iteration	Best	Worst	Average	Difference
GA	31	16	700	378.02	403.34	389.52	25.32
IWOA	31	16	700	364.79	389.71	375.38.	24.92
ABC	31	16	700	340.52	361.49	352.37	20.97
WSHA	31	16	700	319.40	335.18	325.81	15.78

Table 6. City (39,25) experimental results.

Algorithm	City Number	Goods Number	Iteration	Best	Worst	Average	Difference
GA	39	25	1000	848.05	912.39	870.28	64.34
IWOA	39	25	1000	802.66	847.32	828.31	44.66
ABC	39	25	1000	812.57	858.29	834.56	45.72
WSHA	39	25	1000	759.21	787.91	775.52	28.7

As seen in Tables 3–6, the WSHA converges quickly in all city configurations. After testing the four datasets and obtaining the Best, Worst, Average, and Difference results for each dataset and averaging them, it is found that compared with the IWOA, the WSHA reduces the average best path cost by about 9.08%; the worst path cost by about 10.58%; the average path cost by about 9.69%; the difference between the best path cost and the worst path cost is reduced by about 34.35%. In addition, the difference between the best and worst costs of WSHA is minimized in all datasets, reflecting the advantage of WSHA in reducing the difference between the best and the worst values. This implies that WSHA can maintain high stability and consistency during the solution process, which is undoubtedly a significant advantage for practical application scenarios that require fast path optimization and scheduling.

Further, this paper visualized the optimal cost of the path, as shown in Figure 8. Figure 8 shows the superiority of the Whale-Swarm Hybrid Algorithm in finding the

shortest path length on four different datasets. Both on the smaller dataset City (17,11) and the larger dataset City (39,25), the can converge to the shorter path length faster, and its performance is relatively stable throughout the iterations. This further demonstrates the efficiency and stability of the WSHA in dealing with such problems.

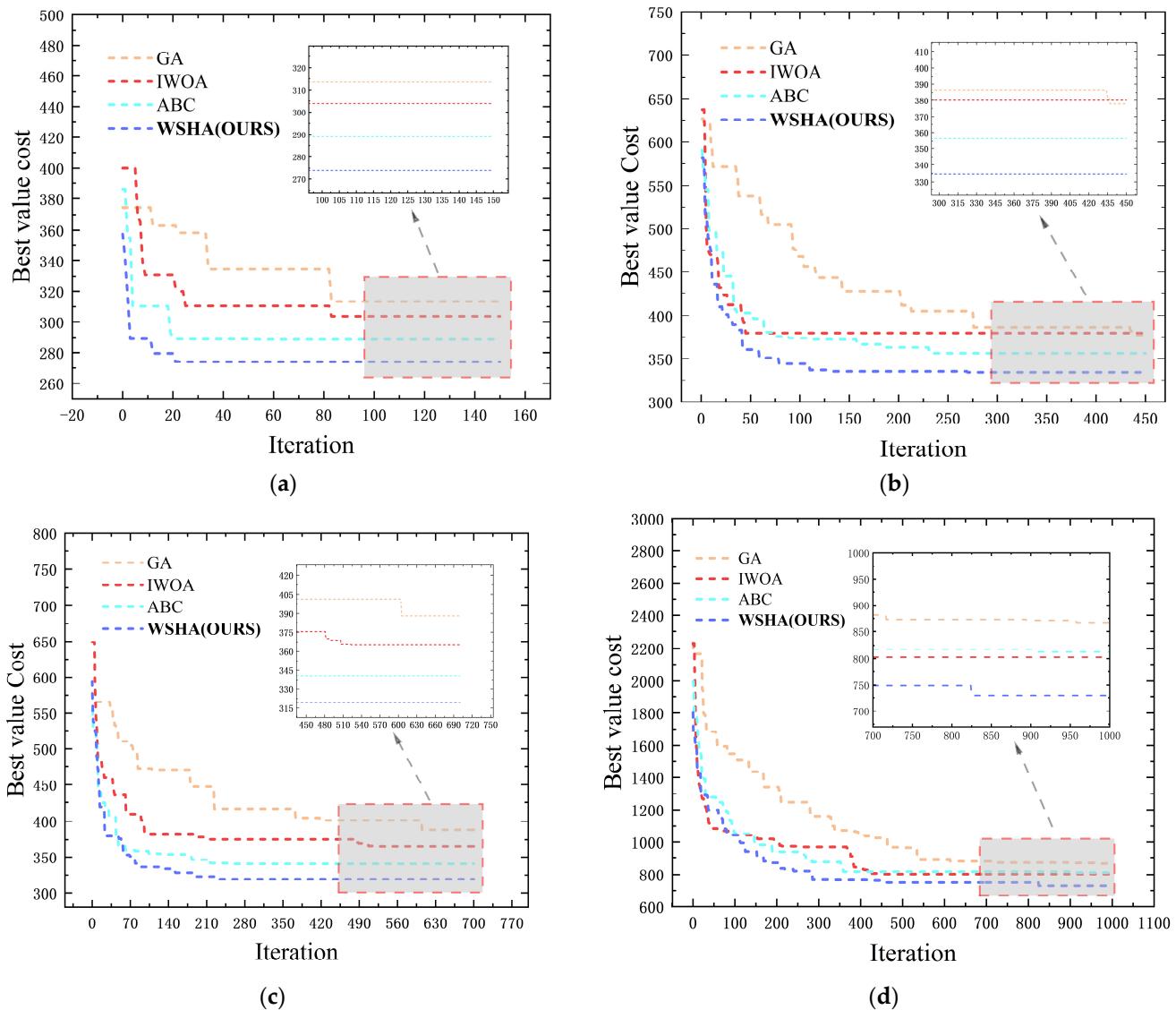


Figure 8. Comparison of the variation of the shortest path lengths of the four datasets. (a) City (17,11); (b) City (24,15); (c) City (31,16); (d) City (39,25).

Secondly, this paper also visualized the paths that pass through the city, as shown in Figure 9. From Figure 9, one can visualize the path-planning graphs generated by the WSHA on different datasets. These path planners show the entire route from the starting point to each target city, and back to the starting point. For the smaller dataset City (17,11), the paths generated by the WSHA are relatively simple but can effectively cover all the necessary city nodes. For the larger dataset City (39,25), the WSHA generates more complex but equally efficient paths, demonstrating the algorithm's scalability and robustness in dealing with large-scale problems.

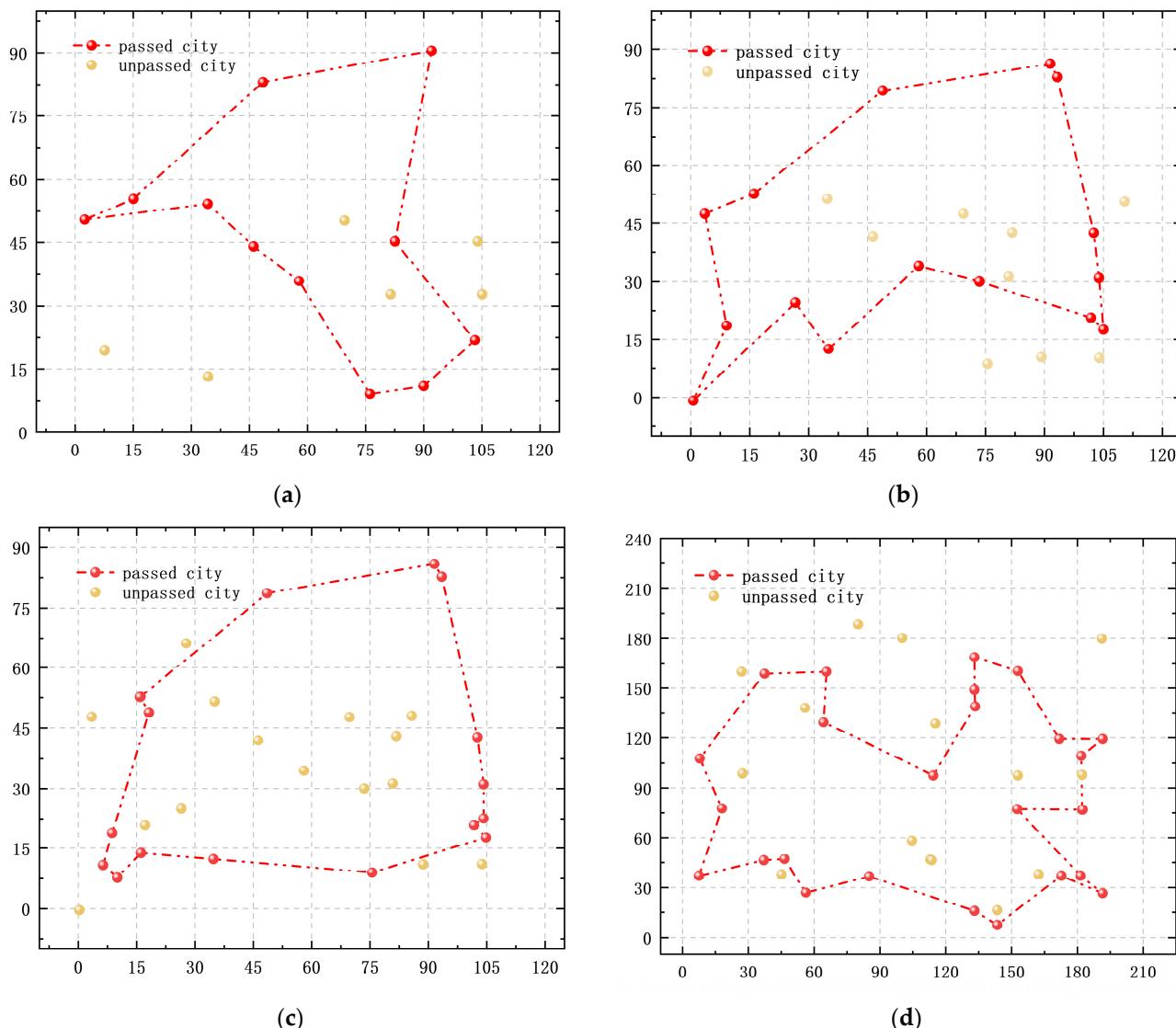


Figure 9. Four dataset path-planning diagrams. (a) City (17,11); (b) City (24,15); (c) City (31,16); (d) City (39,25).

Finally, in this paper, the shortest path planning of City (39,25) is mapped in detail and simulated with multiple rounds of simulation with the actual space located in Texas, USA. The results obtained are shown in Figure 10. By comparing Figure 10, the close correspondence between the shortest path planning of the City (39,25) and the actual space layout in Texas can be seen. This precise mapping relationship not only helps to visualize the significant advantages of the WSHA in path planning but also reinforces the applicability and effectiveness of the algorithm in real spatial applications. Each city in the figure is represented as a point, while the paths are represented by line segments connecting these points. The result not only visualizes the effect of the algorithm, but also provides a solid theoretical and practical foundation for the promotion and application of the WSHA in actual spatial planning.

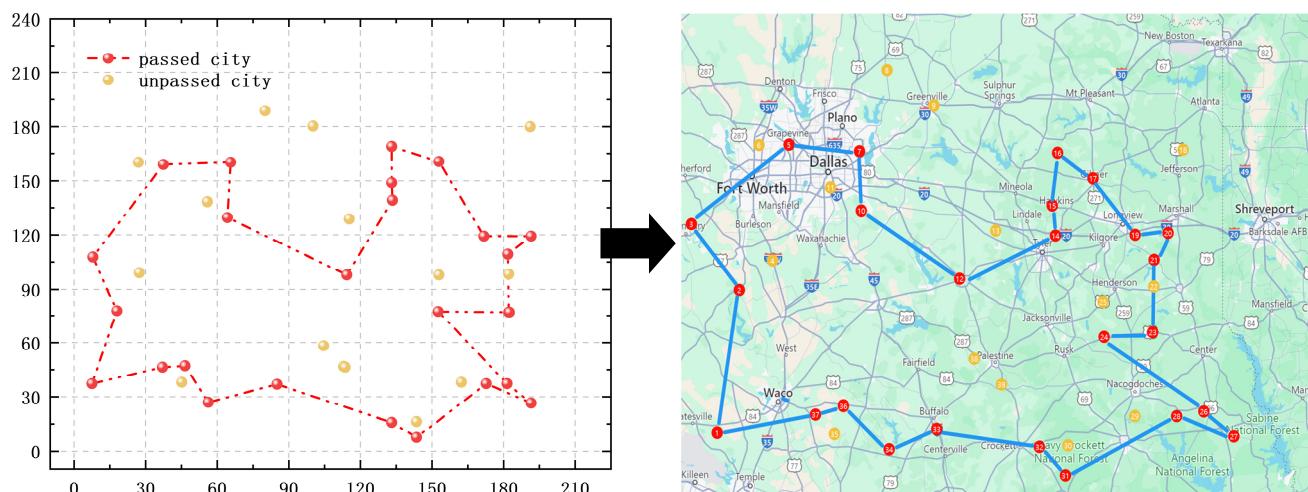


Figure 10. City (39,25) path planning and real space mapping.

4.2. Precise Identification Technology for Emergency Airdrop Stations

4.2.1. Data Preparation and Enhancement Strategies

The dataset used in this paper contained a total of 1189 images. Of these, the number of original images was 312. Given the relative scarcity of images of the emergency airdrop stations and the apparent visual difference between the emergency airdrop station built on top of a tall building and the conventional ground view, this may lead to significant changes in the shape, size, and appearance of the target object. In addition, different lighting conditions may also have a profound effect on the visibility and detection performance of the target. In this paper, geometric transformations (e.g., flipping, rotating, cropping, scaling) and pixel transformations (e.g., Gaussian blurring, saturation adjustment) are used to enhance the data from the photographs. After these processes, the total number of original pictures with data-enhanced pictures reaches 1189. For these data-enhanced and original pictures, labelImg data annotation software was used in this paper for manual annotation. The images are shown in Figure 11. To ensure the rationality of the dataset, this paper divided the dataset into a training set, validation set, and test set in the ratio of 7:1:2.



Figure 11. Example dataset. (a) Original image; (b) Original image labelling; (c) Data-enhanced image; (d) Data-enhanced image labelling. Note: The red boxes indicate the position of the target object to be detected.

To further illustrate the reality of UAV load drops, Figure 12 depicts the path of the UAV and the load drop process in detail. The UAV is capable of accurately delivering supplies to a target location on the ground or to a designated drop zone on top of a high-rise building. This process requires careful planning and real-time adjustments to effectively respond to complex real-world environments. Accurate delivery of supplies in such scenarios is critical for military emergency relief operations.

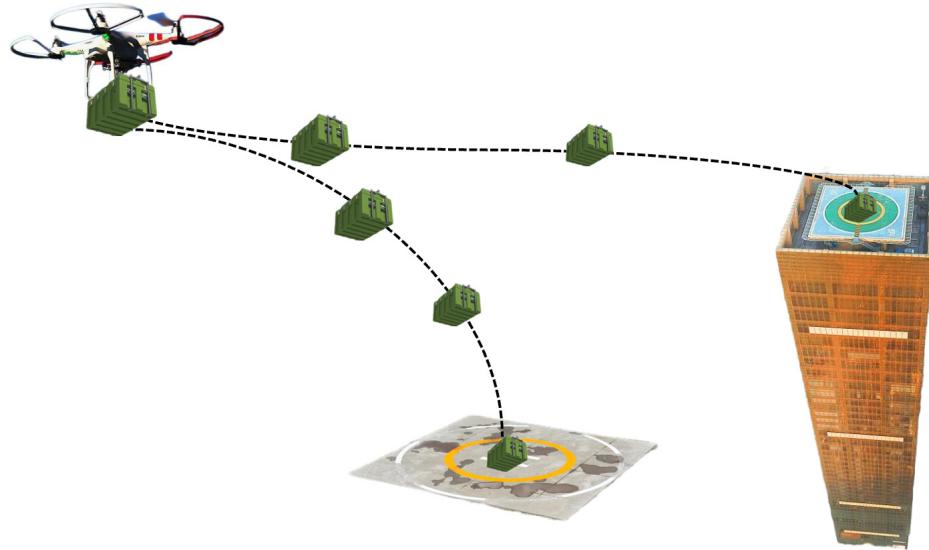


Figure 12. Illustration of the physical situation of UAV load drop.

4.2.2. Experimental Platform

Table 7 shows the experimental hardware setup: an Intel(R) Xeon(R) Platinum 8255C CPU (2.50 GHz), and model training was performed on a GeForce GTX 3080 with 10 GB of graphics memory and 40 GB of system memory. The experiments were run on an Ubuntu 20.04 operating system, programmed in Python 3.8.0, and built based on the Pytorch 2.0.0 framework, which includes CUDA 11.8 for processing power enhancement. Uniform parameters were used in the training phase to ensure the validity of the comparison experiments. The experimental dataset is 1189 and the input image size is 640×640 . The optimizer selects SGD and the learning rate is set to 0.01. The momentum is set to 0.937; the number of iterations is 80; the batch size is 8; and no pre-training weights are loaded.

Table 7. Experimental parameter configuration.

Configuration	Name	Type
Hardware	CPU	Intel(R) Xeon(R) Platinum 8255C
	GPU	NVIDIA GeForce GTX3080
	Memory	40GB
Software	CUDA	11.8
	Python	3.8
	Pytorch	2.0.0
Hyperparameters	Operating system	Ubuntu20.04
	Learning Rate	0.01
	Image Size	640×640
	Workers	8
	Batch Size	8
	Epoch	80
	Optimizer	SGD
	Momentum	0.937

4.2.3. Ablation Experiments

To verify the validity of SimAM, GhoHGBlock, and LSCDHead, four groups of ablation experiments were set up in this paper. The first group of experiments used YOLOv8n

to validate the dataset, and the second, third, and fourth groups of experiments add SimAM, GhoHGBlock, and LSCDHead, respectively, based on YOLOv8. The results of the experiments are shown in Table 8.

Table 8. Results of ablation experiments.

ID	Baseline	SimAM	GhoHGBlock	LSCDHead	mAP@0.5/%	mAP@0.5-0.95/%	Params/M	FLOPs/G	FPS
0	✓				88.5	64.8	3.00	8.1	188.4
1	✓	✓			92.8(↑4.3%)	68.0(↑3.2%)	3.00	8.1	179
2	✓	✓	✓		93.1(↑0.3%)	67.3(↓0.7%)	2.31(↓23.0%)	6.8(↓16.1%)	154.5
3	✓	✓	✓	✓	92.4(↓0.7%)	68.2(↑0.9%)	1.67(↓27.7%)	5.3(↓22.1%)	151.3

Note: The upward (↑) and downward (↓) arrows indicate the percentage change in performance compared to the previous row.

The results from ID 0 to ID 1 indicate that the Simple Attention Module effectively improved the model's capability for feature extraction and attention allocation to target objects, thereby enhancing detection performance. The mAP@0.5/% increased from 88.5% to 92.8%, while the mAP@0.5-0.95/% also rose from 64.8% to 68.0%. The results from ID 1 to ID 2 show that the inclusion of the GhoHGBlock increased the mAP@0.5/% to 93.1%, reduced Params by 23.0%, and decreased FLOPs by 16.1%, although the mAP@0.5-0.95/% experienced a slight decline. This indicates that the GhoHGBlock managed to reduce the model complexity while maintaining a certain detection performance level. The results from ID 2 to ID 3 indicate that the introduction of the LSCDHead further improved the model's performance, with mAP@0.5-0.95/% increasing from 67.3% to 68.2%, Params decreasing by 27.7%, and FLOPs decreasing by 22.1%. This demonstrates that the LSCDHead successfully implemented the idea of ShConv, achieving fewer Params and FLOPs while minimizing the loss in accuracy as much as possible.

In summary, incorporating SimAM, GhoHGBlock, and LSCDHead positively affected the model's performance, resulting in an efficient and accurate object detection model.

4.2.4. Comparative Experiments

To verify the superiority of SimAM, Bi-Level Routing Attention, MLCA, Triplet Attention, CPCA, MPCA, SegNext Attention, and SimAM were added to the C2f module at the front end of the detection head for experimental comparison, and the experimental data obtained are shown in Table 9.

Table 9. Comparative results of multiple attention mechanisms.

Attention	P%	R%	mAP@0.5/%	Params/M	FLOPs/G
Bi-Level Routing Attention	95.9	84.4	91.1	3.36	9.0
MLCA	91.5	81.8	89.3	3.00	8.1
Triplet Attention	93.7	81.4	91.9	3.00	8.1
CPCA	88.4	83.6	90.2	3.19	8.7
MPCA	89.8	88.2	89.3	3.00	8.1
SegNext Attention	93.5	86.4	92.3	3.14	8.4
SimAM	91.6	89.6	92.8	3.00	8.1

According to the data presented in Table 9, SimAM demonstrates significant advantages in object detection tasks. Compared with other attention modules, SimAM performs best regarding mean Average Precision (mAP@0.5/%) and Recall Rate (R%) and maintains the lowest levels of Params and FLOPs, thoroughly verifying its efficiency. It is worth noting that although some attention modules perform excellently in Precision (P%), they perform poorly in Recall Rate (R%). This may be due to the excessive complexity of these modules, which leads to a slowdown in model inference speed, thereby affecting the Recall Rate. In contrast, SimAM can achieve the best results in both mean Average Precision

(mAP@0.5/%) and Recall Rate (R%) while maintaining high efficiency. This is primarily attributed to SimAM's comprehensive focus on channel and spatial positional features and the practical application of its energy function.

In summary, SimAM outperforms other attention modules in object detection tasks. It performs best in mAP@0.5/% and Recall Rate (R%) while maintaining lower Params and FLOPs. Thus, SimAM was selected as the attention module for this paper.

To verify the superiority of the proposed APCR-YOLOv8 network, this paper listed the performance metrics comparison among Faster R-CNN, SSD, YOLOv3-tiny, YOLOv5n, YOLOv6n, YOLOv7-tiny, YOLOv8n, and our proposed model (OURS). Performance metrics include Precision (P%), Recall (R%), Average Precision at IOU = 0.5 (mAP@0.5), Average Precision from IOU = 0.5 to 0.95 (mAP@0.5-0.95), number of parameters in millions (Params/M), GFLOPs (billion floating-point operations per second), and frames per second (FPS). The comparison results are shown in Table 10.

Table 10. Comparison results of multiple testing models.

Models	P%	R%	mAP@0.5/%	mAP@0.5-0.95/%	Params/M	FLOPs/G	FPS
Faster R-CNN	--	--	80.69	--	136.69	369.7	38.1
SSD	--	--	60.70	--	23.13	60.8	182.3
YOLOv3-tiny	73.5	70.7	77.6	47.2	8.67	12.9	474.3
YOLOv5n	64.6	68.6	66.0	42.6	1.76	4.1	199.7
YOLOv6n	82.3	71.4	0.78	56.5	4.30	11.1	203.8
YOLOv7-tiny	72.6	58.9	73.3	45.5	6.01	13.1	201.3
YOLOv8n	89.1	76.4	88.5	64.8	3.00	8.1	188.4
OURS	90.8	85.5	92.4	68.2	1.67	5.3	151.3

Based on the data analyzed in Table 10, APCR-YOLOv8 demonstrates the best performance in the emergency airdrop station detection task. Compared to the other models, APCR-YOLOv8 achieves the highest scores in both Precision and Recall, which fully demonstrates its accuracy in the target detection task. It is worth noting that APCR-YOLOv8 improves detection accuracy and optimizes model complexity and computational efficiency. The model has fewer Params and lower FLOPs, demonstrating its lower demand for computational resources and the advantage of high efficiency. Although there is a slight reduction in the frames per second (FPS) performance, it still meets the requirements of real-time detection. In summary, APCR-YOLOv8 maintains a high detection rate and reduces computational resource consumption while improving detection accuracy; a significant improvement scheme over other models.

4.2.5. Visual Analysis

In this paper, target detection of emergency airdrop stations using the APCR-YOLOv8 algorithm was implemented to understand the model before and after improvement for the task. The before-and-after comparison is demonstrated by heatmap visualization, as shown in Figure 13. Figure 13a is the original input image, Figure 13b is the target-detection heatmap of YOLOv8, and Figure 13c is the target-detection heatmap of APCR-YOLOv8. The heat map uses a gradient of colors to indicate the spatial distribution and recognition priority of potential targets. Hotter regions (red to yellow) are where targets will likely be present in the algorithm's predictions, while cooler regions (blue to purple) indicate less attention.

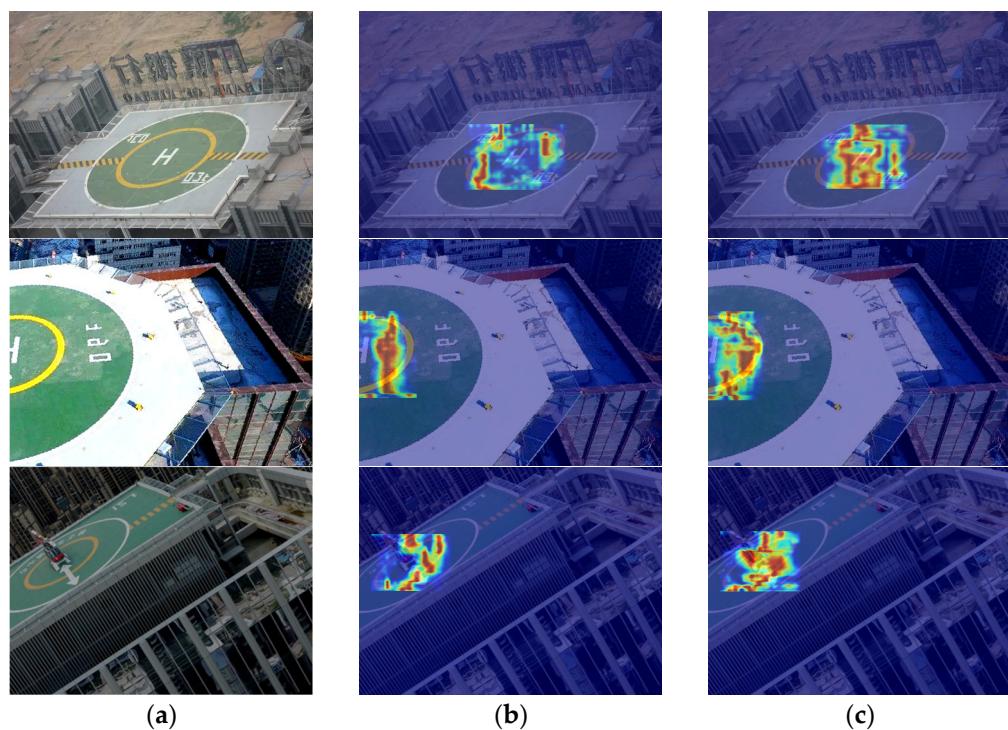


Figure 13. Comparison of YOLOv8 and APCR-YOLOv8 heat maps. (a) Original plot of the dataset; (b) heat map of YOLOv8; (c) heat map of APCR-YOLOv8.

As can be seen from Figure 13, there are the following differences before and after the model improvement. (1) Concentration of the heat map: In the heat map of YOLOv8, the hotspots are more scattered, which indicates that the algorithm's identification of the potential target location is carried out in the broader area, not specifically targeting a specific target. On the other hand, the heat map of APCR-YOLOv8 shows more concentrated hotspot areas, indicating that the improved model algorithm is more accurate and focused on target localization. (2) Clarity of the heat map: the edges of the hotspot areas in the APCR-YOLOv8 heat map are more precise and sharper than in the YOLOv8 heat map. This means that the algorithm has improved the recognition of the target boundary, reducing the background interference and improving the accuracy of target detection. (3) Difference in heat level: compared to YOLOv8, the heat map of APCR-YOLOv8 shows more excellent contrast in color intensity, especially in the target area. This indicates that the algorithm has more confidence in the target, reinforcing the contrast between the target area and the surroundings.

In summary, the heat map performance of the APCR-YOLOv8 algorithm in the target detection task is significantly improved compared to the original YOLOv8 algorithm. Specifically, APCR-YOLOv8 performs better in critical metrics such as clarity, concentration, and confidence.

To reflect the performance of the APCR-YOLOv8 algorithm, it was qualitatively analyzed with the YOLOv8 algorithm in this paper. Examples of the detection results of the two detection networks in the same test environment are shown in Figure 14.

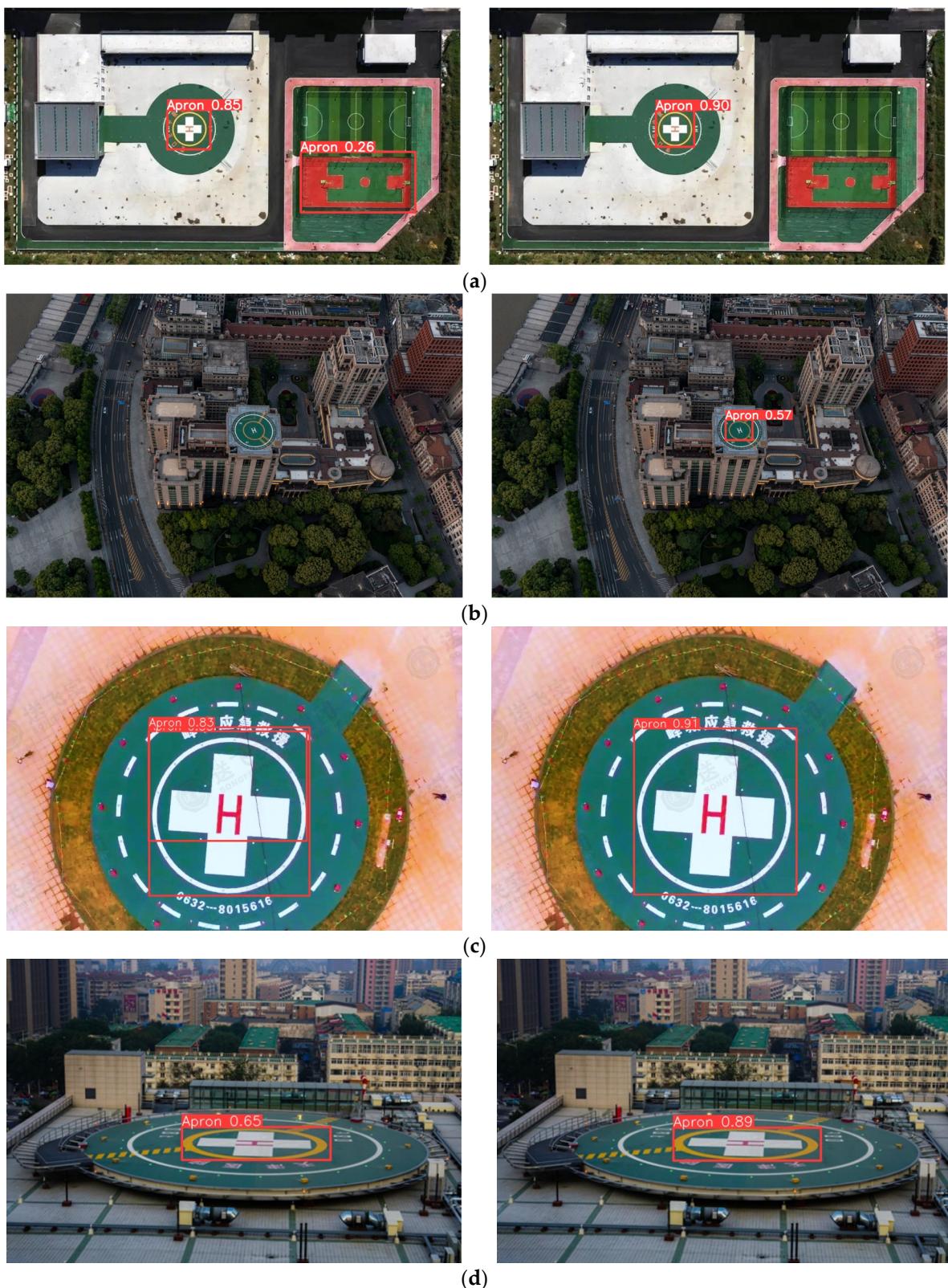


Figure 14. Example analyses of emergency airdrop station detection using APCR-YOLOv8. (a) Solving the false detection problem by accurately distinguishing between actual and false targets; (b) Solving the tiny target missed detection problem by enhancing the detection of small and difficult-to-detect targets; (c) Solving the repeat detection problem by avoiding repeated detections of the same target; (d) Enhancing detection confidence with higher confidence scores in the bounding boxes.

As shown in Figure 14, the APCR-YOLOv8 algorithm demonstrates superior performance over YOLOv8 in several aspects. (a) In solving the false detection problem, APCR-YOLOv8 accurately distinguishes between actual and false targets, reducing the false positive rate and improving overall detection accuracy. (b) For the problem of missed detection of tiny targets, APCR-YOLOv8 enhances the detection capabilities for small and difficult-to-detect targets which were initially ignored by YOLOv8, thanks to its finer extraction and aggregation of target features. (c) In dealing with the repeated detection problem, APCR-YOLOv8 significantly reduces repeated detections of the same target, showing higher stability and robustness in the detection process. (d) Regarding detection confidence, APCR-YOLOv8 improves the confidence level of detections from 0.65 to 0.89 for the same frame, indicating a significant enhancement in the reliability of target recognition in complex backgrounds.

In summary, the APCR-YOLOv8 algorithm demonstrates significant advantages in several key areas, including reducing false positives, minimizing missed detections of small objects, eliminating duplicate detections, and enhancing detection confidence. These improvements make APCR-YOLOv8 more accurate and robust for emergency airdrop station detection tasks, providing a more reliable solution for practical applications.

5. Conclusions

This paper proposed effective solutions for the problems encountered in the UAV delivery process, such as complex route planning, low detection accuracy, and hardware limitations, which have essential theoretical value and broad application prospects. Integrating bionic optimization algorithms and image processing technology improves the efficiency and accuracy of material delivery and plays a vital role in enhancing emergency response capabilities. In the simulation conducted in Texas, USA, the Whale-Swarm Hybrid Algorithm, when optimizing paths, reduced the average optimal path cost by about 9.08%, the worst path cost by about 10.58%, the average path cost by about 9.69%, and the difference between the best and worst path costs by about 34.39%, compared to the single IWOA algorithm. In target detection for emergency airdrop stations, the APCR-YOLOv8 model, by integrating SimAM, a lightweight backbone, and a lightweight LSCLHead, reduced Params and FLOPs by 44.33% and 34.57%, respectively, increased mAP@0.5% from 88.5 to 92.4, and achieved an FPS of 151.3, effectively addressing the original model's issues of missed detection, false detection, and duplicate detection while meeting real-time requirements. Despite the apparent success of the current research work, some limitations still exist. The adaptability of the currently proposed algorithms in highly complex environments or specific situations still needs to be strengthened. Future research will focus on further exploring the multiple GTSP issues of multiple UAV cooperative material delivery, aiming to improve the efficiency of material delivery and enhance the model's adaptability. In addition, we plan to deploy the algorithm to different UAV platforms to broaden its application scope further.

Author Contributions: Conceptualization, H.L. and Y.W.; Data curation, Z.W. and J.Y.; Formal analysis, Y.W. and J.Q.; Methodology, H.L., Y.W. and J.Y.; Resources, H.L.; Software, Q.W. and X.S.; Validation, Z.W. and Q.W.; Writing—original draft, Y.W.; Writing—review and editing, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the Anhui Provincial Natural Science Foundation (No. 2308085MF218) and the open Foundation of Anhui Engineering Research Center of Intelligent Perception and Elderly Care (No. 2022OPB01).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the conclusions of this paper are accessible from the corresponding author upon reasonable request. The data are not publicly available due to privacy.

Acknowledgments: The authors would like to thank each team member sincerely for their efforts.

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

References

1. Gentili, M.; Mirchandani, P.B.; Agnetis, A.; Ghelichi, Z. Locating Platforms and Scheduling a Fleet of Drones for Emergency Delivery of Perishable Items. *Comput. Ind. Eng.* **2022**, *168*, 108057. [[CrossRef](#)]
2. Shi, Y.; Lin, Y.; Li, B.; Li, R.Y.M. A Bi-Objective Optimization Model for the Medical Supplies' Simultaneous Pickup and Delivery with Drones. *Comput. Ind. Eng.* **2022**, *171*, 108389. [[CrossRef](#)] [[PubMed](#)]
3. Wen, X.; Wu, G. Heterogeneous Multi-Drone Routing Problem for Parcel Delivery. *Transp. Res. Part C Emerg. Technol.* **2022**, *141*, 103763. [[CrossRef](#)]
4. Amicone, D.; Cannas, A.; Marci, A.; Tortora, G. A Smart Capsule Equipped with Artificial Intelligence for Autonomous Delivery of Medical Material through Drones. *Appl. Sci.* **2021**, *11*, 7976. [[CrossRef](#)]
5. Lambora, A.; Gupta, K.; Chopra, K. Genetic Algorithm-A Literature Review. In Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 14–16 February 2019; pp. 380–384.
6. Miao, C.; Chen, G.; Yan, C.; Wu, Y. Path Planning Optimization of Indoor Mobile Robot Based on Adaptive Ant Colony Algorithm. *Comput. Ind. Eng.* **2021**, *156*, 107230. [[CrossRef](#)]
7. Phung, M.D.; Ha, Q.P. Safety-Enhanced UAV Path Planning with Spherical Vector-Based Particle Swarm Optimization. *Appl. Soft Comput.* **2021**, *107*, 107376. [[CrossRef](#)]
8. Han, Z.; Chen, M.; Shao, S.; Wu, Q. Improved Artificial Bee Colony Algorithm-Based Path Planning of Unmanned Autonomous Helicopter Using Multi-Strategy Evolutionary Learning. *Aerosp. Sci. Technol.* **2022**, *122*, 107374. [[CrossRef](#)]
9. Dai, Y.; Yu, J.; Zhang, C.; Zhan, B.; Zheng, X. A Novel Whale Optimization Algorithm of Path Planning Strategy for Mobile Robots. *Appl. Intell.* **2023**, *53*, 10843–10857. [[CrossRef](#)]
10. Zheng, R.; Zhang, Y.; Yang, K. A Transfer Learning-Based Particle Swarm Optimization Algorithm for Travelling Salesman Problem. *J. Comput. Des. Eng.* **2022**, *9*, 933–948. [[CrossRef](#)]
11. Pehlivanoglu, Y.V.; Pehlivanoglu, P. An enhanced genetic algorithm for path planning of autonomous UAV in target coverage problems. *Appl. Soft Comput.* **2021**, *112*, 107796. [[CrossRef](#)]
12. Yan, Z.; Zhang, J.; Zeng, J.; Tang, J. Three-Dimensional Path Planning for Autonomous Underwater Vehicles Based on a Whale Optimization Algorithm. *Ocean Eng.* **2022**, *250*, 111070. [[CrossRef](#)]
13. Tian, Q.; Wang, T.; Wang, Y.; Wang, Z.; Liu, C. A Two-Level Optimization Algorithm for Path Planning of Bionic Robotic Fish in the Three-Dimensional Environment with Ocean Currents and Moving Obstacles. *Ocean Eng.* **2022**, *266*, 112829. [[CrossRef](#)]
14. Sun, X.; Chou, P.; Koong, C.-S.; Wu, C.-C.; Chen, L.-R. Optimizing 2-Opt-Based Heuristics on GPU for Solving the Single-Row Facility Layout Problem. *Future Gener. Comput. Syst.* **2022**, *126*, 91–109. [[CrossRef](#)]
15. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
16. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-Cnn: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015; Volume 28, pp. 91–99.
17. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 779–788.
18. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single Shot Multibox Detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14. Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
19. Hurtik, P.; Molek, V.; Hula, J.; Vajgl, M.; Vlasanek, P.; Nejezchleba, T. Poly-YOLO: Higher Speed, More Precise Detection and Instance Segmentation for YOLOv3. *Neural Comput. Appl.* **2022**, *34*, 8275–8290. [[CrossRef](#)]
20. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-Captured Scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 2778–2788.
21. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 7464–7475.
22. Dumitriu, A.; Tatui, F.; Miron, F.; Ionescu, R.T.; Timofte, R. Rip Current Segmentation: A Novel Benchmark and Yolov8 Baseline Results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 1261–1271.
23. Jawaharlal Nehru, A.; Sambandham, T.; Sekar, V.; Ravikumar, D.; Loganathan, V.; Kannadasan, R.; Khan, A.A.; Wechtaisong, C.; Haq, M.A.; Alhussen, A. Target Object Detection from Unmanned Aerial Vehicle (UAV) Images Based on Improved YOLO Algorithm. *Electronics* **2022**, *11*, 2343. [[CrossRef](#)]
24. Souza, B.J.; Stefenon, S.F.; Singh, G.; Freire, R.Z. Hybrid-YOLO for classification of insulators defects in transmission lines based on UAV. *Int. J. Electr. Power Energy Syst.* **2023**, *148*, 108982. [[CrossRef](#)]

25. Jiang, C.; Ren, H.; Ye, X.; Zhu, J.; Zeng, H.; Nan, Y.; Sun, M.; Ren, X.; Huo, H. Object Detection from UAV Thermal Infrared Images and Videos Using YOLO Models. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102912. [[CrossRef](#)]
26. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
27. Yang, L.; Zhang, R.-Y.; Li, L.; Xie, X. Simam: A simple, parameter-free attention module for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 21–24 July 2021; pp. 11863–11874.
28. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
29. Lv, W.; Zhao, Y.; Xu, S.; Wei, J.; Wang, G.; Cui, C.; Du, Y.; Dang, Q.; Liu, Y. DETRs Beat YOLOs on Real-Time Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024; pp. 16965–16974.
30. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: A simple and strong anchor-free object detector. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 1922–1933. [[CrossRef](#)] [[PubMed](#)]

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.