Hindawi
Scientific Programming
Volume 2022, Article ID 7150128, 14 pages
https://doi.org/10.1155/2022/7150128

Research Article

Anti-UAV High-Performance Computing Early Warning Neural Network Based on PSO Algorithm

Yang Lei,[1] **Honglei Yao**,[1] **Bo Jiang**,[1] **Tian Tian**,[1] **and Peifei Xing**[2]

[1]*China Academy of Railway Sciences, Beijing 100081, China*
[2]*School of Electronic Information Engineering, Beihang University, Beijing 100191, China*

Correspondence should be addressed to Peifei Xing; zhyg115@buaa.edu.cn

Received 15 March 2022; Revised 11 April 2022; Accepted 26 April 2022; Published 18 May 2022

Academic Editor: Tongguang Ni

In order to effectively solve the problem that the radar detection system is difficult to detect the "low, small, slow" UAV, the high-performance computing early warning neural network is used to recognize the air UAV in real time and extract the target category and image space location information; the PSO algorithm is used to optimize the parameters of the anti-UAV to ensure that the anti-UAV not only relies on factors but also fully combines the dependence of the visual field factor to quickly obtain the optimal solution through analyzing the high-performance computing early warning neural network in this paper. This algorithm is used to initialize the anti-UAV resources and improve the global optimization capability of the algorithm proposed in this paper. Finally, the experimental results show that the proposed PSO algorithm has better high-performance computing early warning performance to meet the actual needs of network high-performance computing early-warning neural networks.

1. Introduction

With the continuous development of anti-UAV high-performance computing early-warning analysis technology, the ways for people to obtain anti-UAV high-performance computing early warning information are also increasing. Among them, relying on its huge information content, the anti-UAV high-performance computing early warning technology can be used to share resources. By using the computer Internet, the anti-UAV high-performance computing early warning information you need can be obtained. Although users themselves can use the anti-UAV high-performance computing early warning to quickly obtain relevant data, the required data information cannot be achieved by the general browsing and query methods due to the huge amount of data information [1, 2]. In order to effectively solve this problem, Anti-UAV high-performance computing early warning technology came into being. This technology can speed up the user's access to data information, although the required data information is still not accurately obtained under such conditions. The conventional recommendation method has greatly reduced the accuracy of Anti-UAV high-performance computing early warning when processing batch data information requests. In the anti-UAV high-performance computing early warning information environment, the target information data can be obtained from the massive data information and sent to the target user simultaneously to determine the development direction of the anti-UAV high-performance computing early warning.

Regarding the shortcomings of the traditional anti-UAV testing method, the PSO algorithm calculation method in the anti-UAV detection method is adopted, the code characteristics in the detection method are compared with the true characteristics of the notified safety defects, and the similarity matching degree is used to check the Anti-UAV code for defects [3]. This method can be used to better improve the accuracy and work efficiency of detecting defects, and deal with the situation that the current anti-UAV detection method cannot quickly deal with the legacy anti-UAV and the complex structure of the anti-UAV [4]. Real-time analysis of the UAV flight trajectory in the continuous time domain is performed through the PSO algorithm to obtain the predicted trajectory results of the corresponding

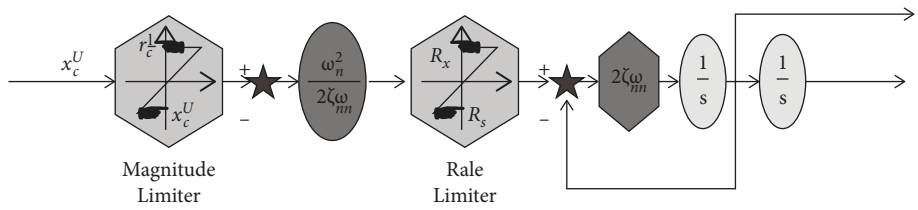

Figure 1: Structure diagram of the second-order control system. The state expression force of the second-order reference model.

UAV in the next stage and realize the full monitoring and prediction of the flight behavior of the UAV in the air. This method has high computational real-time and accuracy and can give anti-UAV capabilities to airports, military units, no-fly zones, and other important venues in the field of machine vision [5–8].

2. Method and Material

2.1. Mathematical Model of UAV Motion. In order to accurately reflect the motion state of the UAV and facilitate the simulation calculation, the impact of the UAV's elasticity is not considered, and the Earth is assumed as an inertial reference frame, ignoring the influence of the curvature of the Earth in this paper [9]. Establish a three-degree-of-freedom model of the UAV.

The motion equation of the UAV in the ground coordinate system is as follows:

$$\begin{cases} \dot{x} = V\ \cos\ \gamma\ \cos\beta + \omega_x, \\ \dot{y} = V\ \cos\ \gamma\ \sin\beta + \omega_y, \\ \dot{z} = V\ \sin\beta + \omega_z. \end{cases} \tag{1}$$

The motion equation of the UAV in the track coordinate system is as follows:

$$\begin{cases} \dot{V} = \dfrac{T\cos\alpha - D}{m} - g\sin\gamma - \dot{\omega}_x\cos\gamma\cos\beta - \dot{\omega}_y\cos\gamma\sin\beta - \dot{\omega}_z\sin\beta, \\ \dot{\beta} = \dfrac{(L + T\sin\alpha)\sin\varepsilon}{mV\cos\gamma} + \dfrac{\dot{\omega}_x\sin\gamma}{V\cos\gamma} - \dfrac{\dot{\omega}_y\cos\beta}{V\cos\gamma}, \\ \dot{\gamma} = \dfrac{(L + T\sin\gamma)\cos\varepsilon}{mv} - \dfrac{g}{V}\cos\gamma + \dfrac{\dot{\omega}_x\sin\gamma cos\beta}{V} + \dfrac{\dot{\omega}_y\sin\gamma\sin\beta}{V} - \dfrac{\dot{\omega}_z\cos\gamma}{V}, \\ \dot{Q} = \omega_z\cos\varepsilon + \omega_y\sin\varepsilon, \end{cases} \tag{2}$$

where, (x, y, z) is the position of the UAV; V is the flying speed of the UAV; α is the angle of attack; ε is the roll angle; γ is the flight path angle, and β is the course angle; Q is the pitch angle; T Expressed as the UAV thrust; D is the drag force; L is the lift force; m is the mass of the unmanned position: g is the acceleration of gravity; $(\dot{\omega}_x, \dot{\omega}_y, \dot{\omega}_z)$ and $(\omega_x, \omega_y, \omega_z)$ are the components of wind acceleration and wind, respectively, on the three coordinate axes.

In order to better adapt to the requirements of UAV control stability, it is necessary to further consider the speed and bandwidth constraints of state quantity and input quantity in the control system design process. The difficulty of this problem lies in how to mathematically describe the speed and bandwidth constraints of the control commands in the actual system. Therefore, a second-order reference model with nonlinear links is introduced to reflect the response characteristics of the actual system to control commands. The second-order reference model structure with amplitude, speed rate, and bandwidth constraints is shown in Figure 1.where sat $(\cdot)$ is the saturation function; x_c^0 is the input commands; x_c is the output command; x_c^L is the upper limit amplitude constraints of the command, x_c^U is the lower limit amplitude constraints of the command; $\pm R_x$ is the speed constraints of the command; ζ_n and ω_n are the damping ratio and natural frequency of the second-order reference model, respectively frequency, according to these two parameters, the command bandwidth constraint is as follows:

$$\omega_b = \omega_n\sqrt{1 - 2\zeta_n\omega_n + \sqrt{2 - 4\zeta_n^2 + 4\zeta_n^4}}. \tag{6}$$

$$E_c = \text{sat}\left(x_c^0, x_c^L, x_c^U\right) - x_1, \tag{3}$$

$$E_c' = \text{sat}\left(E_c, -\frac{2\zeta_n R_x}{\omega_n}, \frac{2\zeta_n R_x}{\omega_n}\right), \tag{4}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -2\zeta_n\omega_n \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix}E_c', \quad \begin{bmatrix} x_c \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \tag{5}$$

When the second-order reference model is introduced into the control system design process, the input of this link is the input quantity to be designed $\gamma_c^0, \alpha_c^0, Q_c^0$, and the output of this link is the executable control quantity γ_c, α_c, Q_c. In the meantime, the second-order reference model link can also provide the derivative $\gamma_c,\ \dot{\alpha}_c,\ \dot{Q}_c\}$ of the virtual control variable, which can avoid the direct analytical calculation of the virtual control variable. In addition, the introduction of the second-order reference model link greatly improves the autonomy of control-oriented modeling.

2.2. Design of Controller. Assume that the dynamic system of the UAV is a first-order nonlinear system:

$$\dot{x} = Ax + Bu, \tag{7}$$

where

$$x = \left[x_1(t), x_2(t), \ldots, x_m(t)\right]^T \in R^m,$$
$$u = \left[u_1(t), u_2(t), \ldots, u_n(t)\right]^T \in R^n. \tag{8}$$

They are the status of the flight control system and the input of the controlling quantity; $A \in R^{m\times n}$ is the system matrix and $B \in R^{m\times n}$ is the input matrix simultaneously. The purpose of the control system is to establish a suitable control scheme to ensure that the state vector quality can accurately track the constrained ideal $x_d \in R^{m\times n}$. Define the tracking error as follows:

$$e = x_d - x. \tag{9}$$

In addition, the sliding form surface is expressed as follows:

$$s(t) = e(t) + k\int_0^t e(\tau)\mathrm{d}\tau, \tag{10}$$

where $k = \mathrm{diag}(k_1, k_2, \ldots, k_m) \in R^{m\times m}$ is a nonzero positive definite matrix. If the UAV is under ideal conditions, the various parameters of the system are known and constant. The ideal controller can be designed as follows:

$$u^* = B^+\left[-Ax + \dot{x}_d + Ke\right]. \tag{11}$$

Among them, $B^+ = (B^TB)^{-1}B^T$, substituting equation (11) into equation (S) to get the following equation:

$$\dot{e} + Ke = 0. \tag{12}$$

In the above formula, if K satisfies the Huiwitza polynomial when t tends to infinity $\|e\| = 0$. However, the mathematical model of the system cannot be accurately obtained, and the UAV will be subject to parameter perturbation and external interference during flight. The controller based on sliding mode control cannot deal with the influence of these uncertain factors well. Therefore, it is necessary to design an intelligent, Robust control system to achieve accurate tracking of the flight trajectory. At the same time, the system must have a certain anti-interference ability.

The design of the intelligent control system is shown in Figure 2.

The identification of the system and the approximation of the function can be realized through a pure wavelet neural network, and this will also lead to other problems, such as the expansion of the network structure. In addition, the recurrent neural network can realize the identification of nonlinear systems, but it is not stable enough and the learning algorithm is also more complicated. The controller based on the PSO algorithm can not only solve the stability problem but also approximate the ideal mathematical model and reduce the dependence on the mathematical model.

The purpose of this paper is to reduce the high-performance computing early warning neural network overhead of the underlying network under the premise that the underlying network resources are limited, and path segmentation is not supported. A binary combinatorial optimization model for the anti-UAV high-performance computing early warning neural network problem is established.

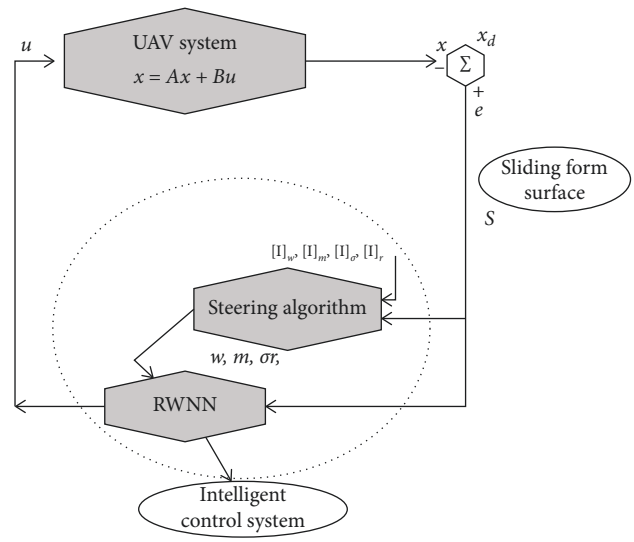

Figure 2: Intelligent control system structure.

First, define the remaining available CPU and memory resources of the underlying node $n_S \in N_S$ as $\mathrm{cpu}'(n_S)$ and $\mathrm{memory}'(n_S)$, respectively, and the remaining available bandwidth resources of the underlying link $e_S \in E_S$ is $b'(e_S)$.

$$\begin{aligned}\mathrm{cpu}'(n_S) &= \mathrm{cpu}(n_S) - \sum_{\forall n_V \perp n_S} \mathrm{cpu}(n_V),\\ \mathrm{memory}'(n_S) &= \mathrm{memory}(n_S) - \sum_{\forall n_V \perp n_S} \mathrm{memory}(n_V),\\ b'(e_S) &= b(e_S) - \sum_{\forall e_V \perp e_S} b(e_V).\end{aligned} \tag{13}$$

Among them, $n_V \perp n_S$ is defined that the virtual node n is allocated to the bottom node n, and $e_V \perp e_S$ is defined that the virtual link n is allocated to the bottom links.

The available bandwidth of any path $P \in P_S$ is expressed as the minimum remaining bandwidth along the path between two lower-level nodes.

$$b(P) = \min_{e_S(i,j)\in P} b(e_S(i,j)). \tag{14}$$

Let M_N be a binary $m \times n$ matrix, which represents the high-performance computing early warning relationship of the node. Each row vector and column vector represent a virtual node and the underlying node, $m = N_V|$, $n = |N_S|$, When the virtual node n_V^i is assigned to the bottom node n_S^j, the value of $M_N(i, j)$ is 1. Otherwise, it is 0. For the same anti-UAV request, each virtual node can only be assigned to one bottom node, and two virtual nodes cannot be assigned to the same bottom node simultaneously. The constraint conditions are formalized as follows:

$$\begin{aligned}\sum_i^m M_N(i,j) \le 1, \quad j \in \{1, 2, \ldots, n\},\\ \sum_j^n M_N(i,j) \le 1, \quad i \in \{1, 2, \ldots, m\}.\end{aligned} \tag{15}$$

The remaining available CPU and memory resources of the underlying node n_S^j need to be able to meet the needs of

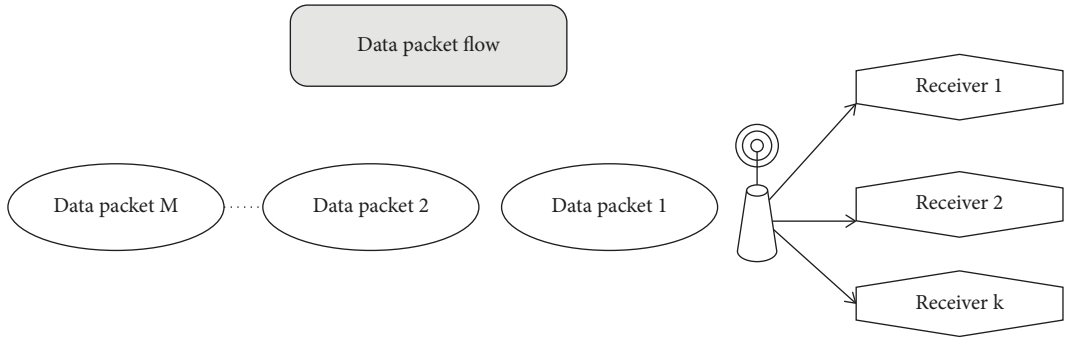

Figure 3: System model.

the virtual node n_V^i, and the node is within the constraint range of the requested location of n_V^i, before n_V^i can be alerted to the underlying node n_S^j by high-performance computing. The distance between nodes n_V^i and n_S^j is represented by Euclidean distance dis (). The remaining available CPU and memory resources of the underlying nodes and the node constraints are, respectively,

$$\begin{aligned} M_N(i,j)\left(\mathrm{cpu}'\left(n_S^j\right) - \mathrm{cpu}\left(n_V^i\right)\right) \geq 0, \\ M_N(i,j)\left(\mathrm{memory}'\left(n_S^j\right) - \mathrm{memory}\left(n_V^i\right)\right) \geq 0, \\ M_N(i,j) \times \mathrm{dis}\left(\mathrm{loc}\left(n_V^i\right), \mathrm{loc}\left(n_S^j\right)\right) \leq D_V^i, \end{aligned} \tag{16}$$

where, $M_N(i,j) \in \{0,1\}$, $i \in \{1,2,\ldots,m\}$, $j \in \{1,2,\ldots n\}$.

The anti-UAV high-performance computing early warning neural network model in this paper is shown in Figure 3 [10].

(1) The system model includes a source node and K ($K > 2$) receiving nodes are configured, and the packet set needs to be broadcast to K receiving nodes. In this paper, it is assumed that the source node is within a period of time. Broadcast data packet within Δt

(2) The receiving node sends ACK or NAK information to the source node, the source node receives and maintains the feedback matrix table T, $T = (K, N)$, and the matrix element $T(i,j)$ indicates whether the receiving node correctly receives the data packet. Here, $R_i P_j 1 \leq i \leq K$, $1 \leq j \leq M$.

(3) In short, here, it is assumed that all control messages ACK or NAK are sent instantaneously and are not lost.

(4) The node R_i data packet loss rate obeys the binomial distribution with the parameter of p_i ($i = 1, 2, \ldots, K$).

In this article, if it is considered that there are n packets of the same size in the network, they need to be sent and to be packeted and expressed $X_1, X_2, \ldots, X_n$. The source node encodes the random linear asynchronous or the packet lost by the receiving node, and the new packet Y_i is expressed as follows:

$$Y_i = \sum_{j=1}^{n} g_{ij} X_j. \tag{17}$$

The coding coefficient g_{ij} ($1 \leq j \leq n$) is a value randomly selected from the defined area F_q. After each receiving node receives n coded packets, it can decode the original packet through the next linear equation.

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} = \begin{pmatrix} g_{11} & \cdots & g_{1n} \\ \vdots & \ddots & \vdots \\ g_{n1} & \cdots & g_{nn} \end{pmatrix}^{-1} \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}. \tag{18}$$

The PSO algorithm introduced in this paper is divided into a data broadcast stage and a retransmission stage. The specific steps are as follows:

(1) The source node broadcasts N data packets to K receiving nodes, and each data packet is sent at a certain time interval. The source node establishes a feedback matrix T through the received ACK or NAK feedback information and maintains the update.

(2) The source node enters the retransmission phase after the time $M\Delta t$ after broadcasting N packets. All lost data packets form a set $D = \{X_1, X_2, X_3, \ldots, X_n\}$, and the largest coefficient $M_{\max}$ in the coefficient vector $G = \{g_{i1}, g_{i2}, g_{i3}, \ldots, g_{in}\}$ ($1 \leq i \leq M_{\max}$) (selected randomly from the limited domain F_q) is used to encode all the lost data packets to generate $M_{\max}$ coded packet. $M_{\max}$ is the maximum number of lost data packets in all receiving nodes, which is determined by the following formula:

$$M_{\max} = \max_{i \in \{1,2,\ldots,K\}} \left\{ \sum_{j=1}^{K} T(i,j) \right\}. \tag{19}$$

(3) After resending the encoded data packet, each receiving node estimates and displays the arrangement of its own encoding vector matrix G. $M_{\max}$ r_i If $r_i \neq N$ means that G has not reached the full arrangement for the node R_i, then the node R_i will notify the source node how many coded packets need to be retransmitted before G can reach the full arrangement. The required coded packets are

expressed through N_i, the specific situation is shown in the following formula [11]:

$$N_i = \begin{cases} N - r_i, r_i \le N, \\ 0, r_i \ge N. \end{cases} \tag{20}$$

In the formula, $i = 1, 2, \ldots, K$

If the receiving node R_i receives a $M_{\max}$ coded packet in the data retransmission phase, N_i is 0. If the node R_i loses 2 encoded packets, then $N_i = 2$.

(4) The source node updates $M_{\max}$ according to the N_i value fed back by each receiving node and generates $M_{\max}$ coded packet in the new retransmission stage. The algorithm is shown in equation (4).

(5) (3) and (4) are repeated until the vector matrix of all receiving nodes is equal to N. That is $M_{\max} = 0$, if there is no lost packet, the receiving node can use the Gaussian elimination method to decode the original data packet.

The PSO algorithm is based on a complex adaptive system and belongs to a random search algorithm. This is also collective intelligence; everyone works together to solve problems. w belongs to the convolutional neural network group algorithm, which is a more important and changeable parameter and plays an important role in the improvement of the algorithm. If w becomes larger, the speed will become smaller, which is beneficial to the overall retrieval. If you use w to reduce the time, the speed will be shortened, which is good for local search. How to control the value of w and effectively solve the problem is a hot spot in the research process. Related research has proposed a linearly decreasing inertia weight, namely LDW, which linearly changes w to improve the convergence of the algorithm.

$$w = w_{\max} - \frac{t \cdot (w_{\max} - w_{\min})}{t_{\max}}. \tag{21}$$

In the formula: the value of w is $[w_{\min}, w_{\max}]$; t is the current iteration number; $t_{\max}$ is the maximum value obtained.

When w decreases linearly, the initial convergence speed decreases, and then as w decreases, the diversity of the algorithm decreases, and the local optimum is achieved. In this paper, we use a nonlinear weighting method to solve the deficiencies in the convolutional neural network group algorithm.

$$w = w_{\max} - (w_{\max} - w_{\min}) \cdot \arcsin\left(\frac{t}{t_{\max}} \cdot \frac{\pi}{4}\right). \tag{22}$$

According to the above formula, when the value of t is small, the approximate value of w is equal, and when the value of w is large, it is very advantageous for global search. In the process of increasing t, w decreases nonlinearly, and the value of w is relatively small, so the good local search ability of the algorithm is ensured, and the global search and local search can be adjusted flexibly. Generally speaking, the best design problem can be solved by using a three-layer structure network. Therefore, a three-layer neural network is used, namely input and output and implicit layers.

(1) Network initialization: Each level has a corresponding right coefficient, and a random small nonzero number is given to realize the threshold initialization of each level and determine the learning speed and neuron excitation function.

(2) Output calculation of each layer. First, realize the input of the sample $X = (x_1, x_2, \ldots, x_n)$ and realize the output of $Y = (y_1, y_2, \ldots, y_n)$, and calculate the neuron output by the following formula:

$$\begin{aligned} H_i &= f\left(\sum_{j=1}^{m} w_{ij} - a_i\right), \quad i = 1, 2, \ldots, q, \\ O_k &= \sum_{i=1}^{q} H_i w_{ki} - b_k, \quad k = 1, 2, \ldots, L. \end{aligned} \tag{23}$$

In the formula: H belongs to the hidden layer output. Number of nodes; a is the threshold activation function; the connection weight of the input layer and the implicit layer. O refers to the output of the output layer. B is the connection weight between the hidden layer and the output layer representing the threshold.

(3) Calculate the error e between the network output O and the expected output O_1.

$$e_k = O_{1k} - O_k, \quad k = 1, 2, \ldots, L. \tag{24}$$

(4) Update network connection and threshold:

$$\begin{aligned} w_{ij} &= w_{ij} + \mu H_i (1 - H_i) x(j) \sum_{k=1}^{L} w_{ki} e_i, \quad j = 1, 2, \ldots, m;\ i = 1, 2, \ldots, q, \\ w_{ki} &= w_{ki} + \mu H_i e_k, \quad i = 1, 2, \ldots, q;\ k = 1, 2, \ldots, L, \\ a_i &= a_i + \mu H_i (1 - H_i) x(j) \sum_{k=1}^{L} w_{ki} e_i, \quad j = 1, 2, \ldots, M;\ k = 1, 2, \ldots, L, \\ b_k &= b_k + e_k. \end{aligned} \tag{25}$$

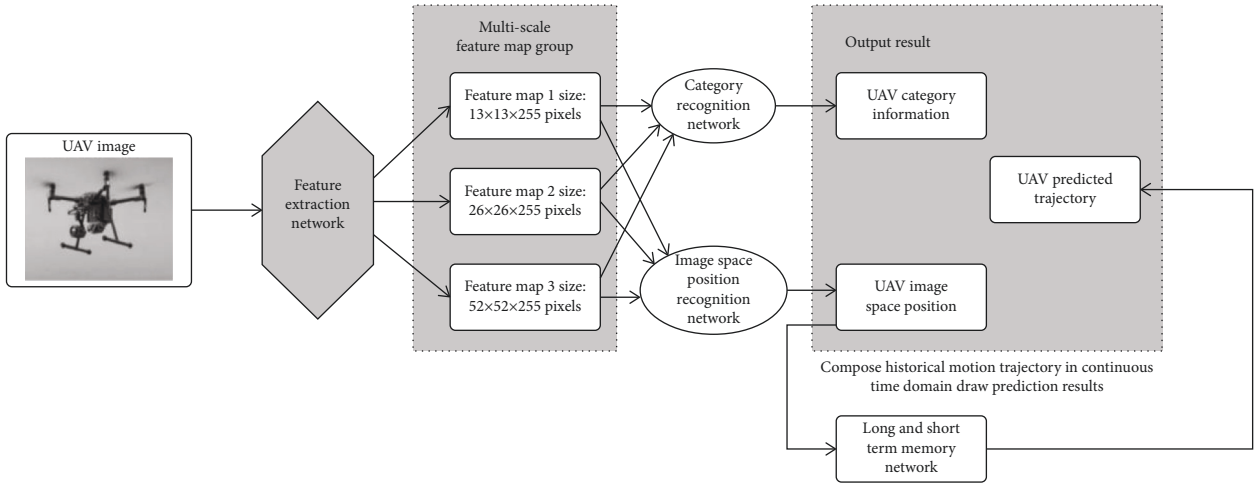

Figure 4: AUNN network structure.

The values of w_{ki} and w_{ij} can be adjusted by the network error e and the above formula, and the values of a and b can also be adjusted by e.

2.3. Composition of System. The designed anti-UAV recognition and trajectory prediction neural network, AUNN). (AUNN) is composed of 4 links of UAV feature extraction, UAV type recognition, UAV image space position recognition, and UAV image space trajectory prediction, as shown in Figure 4.

First, the algorithm can obtain the multiscale deep semantic feature information of the UAV from the aerial UAV image captured by the imaging system through the feature extraction network; Subsequently, the obtained multiscale and high-dimensional feature maps are sent to the UAV type recognition network and the UAV image space position recognition network, to figure out the target UAV category and image space position. According to the change trend of the corresponding UAV's image space position center in a certain time domain, the historical movement trajectory of the UAV in the changed time domain is constructed. Finally, combined with the historical motion trajectory, the predicted motion trajectory of the target UAV in the future time domain is analyzed and output with the PSO algorithm, and the real-time recognition and tracking of the UAV in the current time domain and the trajectory prediction in the future time domain are completed.

Anticipating in the anti-UAV defect inspection, usually based on the behavior of other users in the community, your own participation behavior can be adjusted. According to the above-mentioned user behavior characteristics, the anti-UAV defect detection manager uses incentive guidelines and other methods to reasonably check the anti-UAV defects to improve the user's participation. This kind of management scheme manages the anti-UAV defect detection information resources intelligently. Anti-UAV high-performance computing early warning intelligent management: The anti-UAV defect inspection manager analyzes the user's behavior data and browsing history, and other user resources, explores the needs of users, predicts the user's behavior tendency, and provides users with personalized services that meet their needs, and ultimately realizes the added value of service experience through the organization and regeneration of information.

This article refers to the interactive mechanism model of enterprise knowledge management and service innovation and summarizes the anti-UAV high-performance computing early warning platform model (see Figure 5) to clarify the shortcomings of the anti-UAV, how to configure information, and use data to improve community environment and improve the viscosity of user participation.

The management of any information resource is a process, and the detection of the defects of the anti-UAV consists of several related ordered rings, which constitute the entire organic ring. Intelligent management refers to the analysis of information resources in anti-UAV high-performance computing early warning based on the above three links of resource accumulation, resource arrangement, and resource utilization. Please refer to Figure 6 for the specific process.

It can be seen from Figure 6 that the resource storage ring contains information storage and information collection. In the resource storage link, the anti-UAV system detects the collected anti-UAV defects. The data sorting link includes the sorting of the results of information organization, information analysis, and information data analysis. For target user data, the system data of the mobile phone is analyzed and compared, and in-depth, intelligent operations are taken to make it effective and reasonable. Resource utilization includes a system recommendation link, which classifies and organizes target users through the analyzed data, provides effective and systematic data to corresponding users, and improves resource utilization efficiency.

2.4. Algorithm Implementation. In order to quickly and intuitively obtain the category and location information of aerial UAV, it is first necessary to identify and locate the UAV in the image or video stream data captured by the

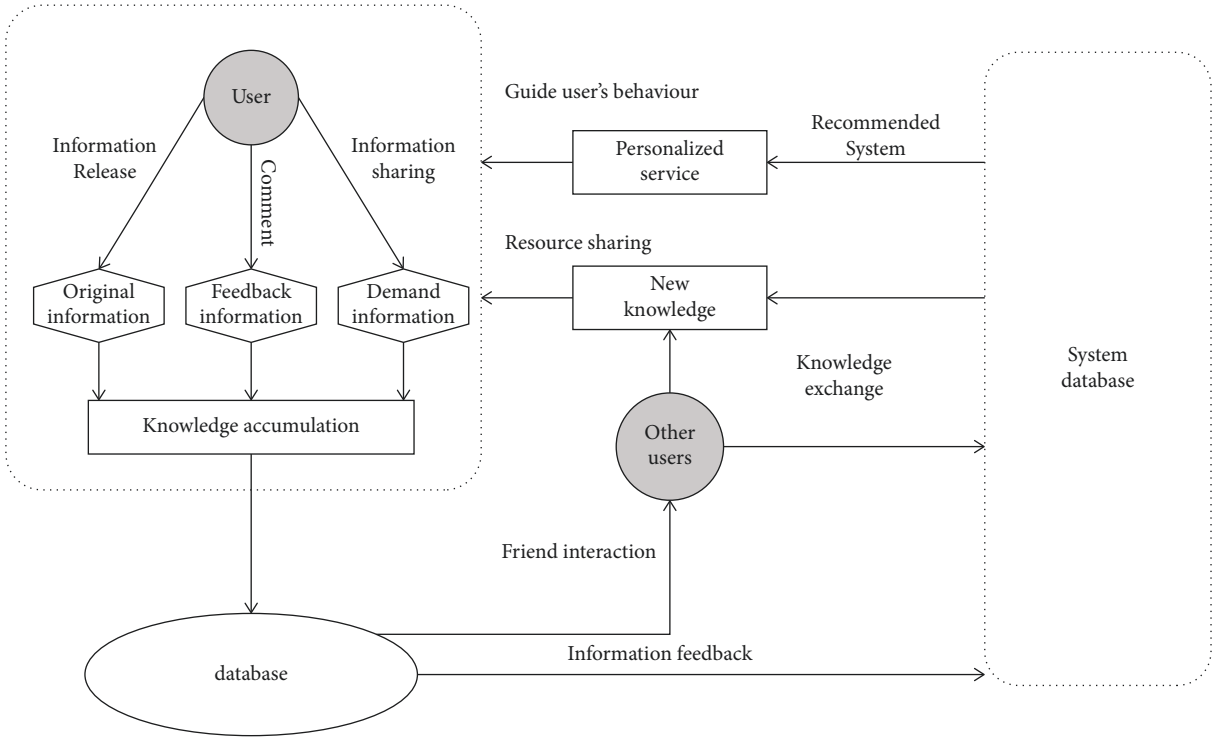

Figure 5: Anti-UAV high-performance computing early warning platform.

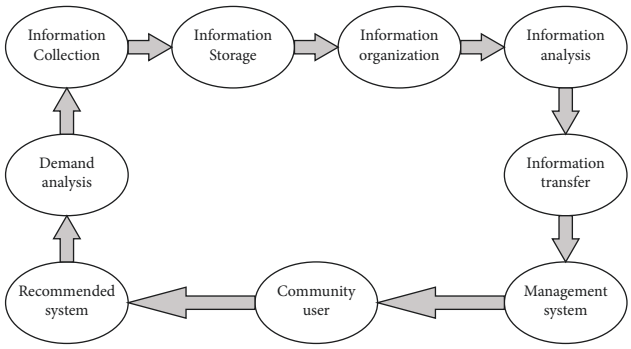

Figure 6: Anti-UAV high-performance computing early warning process.

camera. Orienting at ground-based computing platforms, in view of multiscale and multitarget UAV recognition tasks in the air environment, a UAV Feature Extraction Network (UFEN) is designed as the backbone of AUNN to extract deep semantic information of aerial UAV and use the YOLO V3 (you only look once) target recognition network as the middle layer (neck) and top layer (head) of the AUNN network to complete the target category and image space recognition calculations.

2.4.1. UAV Feature Extraction Network. The UFEN network is a deep residual network, which is composed of a standard convolutional layer, an expanded convolutional layer, and a stack of cyclic residual modules. The network structure of UFEN is shown in Figure 7.

Since the ground-based platform is shooting aerial UAVs, the distance from the UAV to the ground-based detection unit is generally 45 m. At this time, the amount of semantic information in the foreground of the UAV in the image is much smaller than the amount of background semantic information, and the whole colored sky dominates the background. When using the same effective size convolution kernel, the receptive field of standard convolution and the receptive field of expanded convolution is shown in the following equations:

$$r_n = r_{n-1} + (k_n - 1)\prod_{i=1}^{n-1} s_i, \tag{26}$$

$$r_n = r_{n-1} + d(k_n - 1)\prod_{i=1}^{n-1} s_i, \tag{27}$$

where r_n is the receptive field of each unit in the nth convolutional layer; i is the first $n-1$ layer of convolution, the index value of each layer of convolution; k_n and S_i are the size and step size of the convolution kernel of the nth convolutional layer, respectively; d is the expansion convolution coefficient.

By comparing equations (26) with (27), it can be seen that under the premise that the moving step of the convolution kernel is the same as the input image size, the receptive field of expanded convolution in the same layer network is larger than that of standard convolution. For images of aerial UAV, expansion convolution can more effectively obtain the deep semantic features of the image and reduce the multiple redundant calculations of the

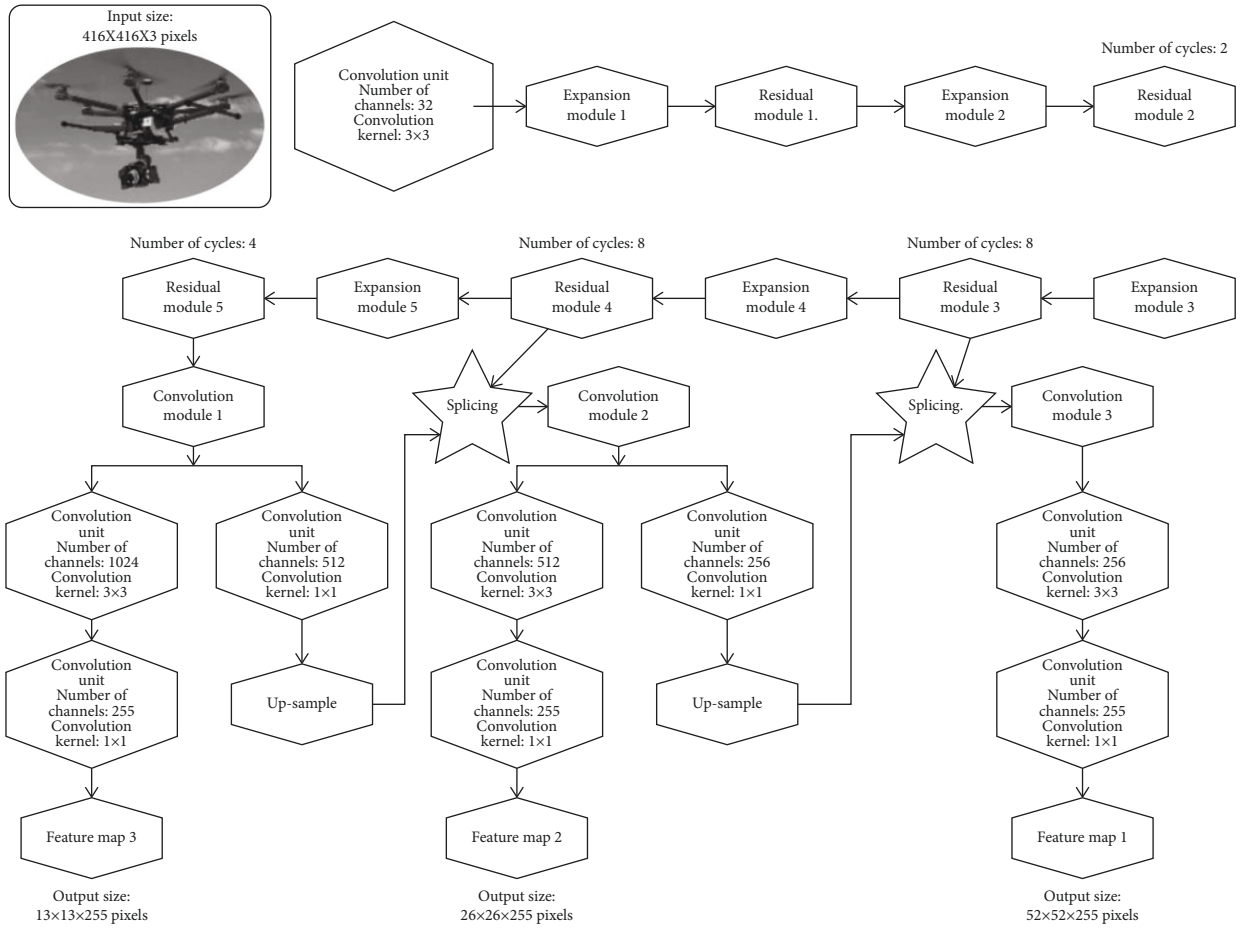

Figure 7: NFEN network structure.

background environment in the standard convolution iteration process. However, due to the discontinuous sampling of the convolution kernel during the calculation process of the expansion convolution, for small aerial UAV targets, the sky dominates loss of spatial hierarchical information, and the inability to reconstruct small object information are prone to occur, as shown in Figure 8(a). In order to solve the above problems, the expanded convolution module with a zigzag structure is used to replace the standard convolution [12, 13] to avoid the expansion convolution kernel with no spatial structure connection to skip or dilute the semantic information points of the UAV. The calculation method of the expanded convolution module with a saw tooth structure is shown in the following formula:

$$\text{Feature} = \text{conv}_{\text{dil}}^{5}\left[\text{conv}_{\text{dil}}^{2}\left(\text{conv}_{\text{dil}}^{1} * \text{input}\right)\right]. \tag{28}$$

In formula (28): input is the input data; Feature is the feature map obtained after calculation; conv_{dil} is the expansion convolution calculation; 1, 2, and 5 are the expansion coefficients. Meanwhile, the expansion coefficient of the high-dimensional expansion convolution is controlled, and the maximum expansion coefficient M_i in the expansion convolution is as follows:

$$M_i = \max\left[M_{i+1} - 2r_i, M_{i+1} - 2\left(M_{i+1} - r_i\right), r_i\right]. \tag{29}$$

In formula (29): r_i is the receptive field of the i-th expanded convolution. The expanded convolution module structure of the sawtooth structure is shown in Figure 9 and Table 1.

After the above mixing and matching, the zigzag structure is fused and convolved, and the full coverage calculation of the feature map information points, as shown in Figure 8(b), is realized. That is, the larger receptive field can be used to extract global semantic information, which also prevents ignorance of target feature information.

When performing deep semantic feature extraction on multiscale UAV targets, due to the continuous deepening of the network and the increase in the number of loop iterations, the size of the feature map will be reduced after each convolution calculation. The small-scale UAV target with less semantic information has a smaller feature map area in each layer. When the network depth is too large, the feature information of the small target will be difficult to be distinguished, and the internal detail texture in the deep semantic information will be weakened. In order to improve the recognition efficiency of small targets, it is necessary to condense deep semantic information and retain shallow feature information [14, 15]. UFEN uses a residual model. Each residual module is composed of two residual units. Each residual unit is shown in Figure 10 including convolutional

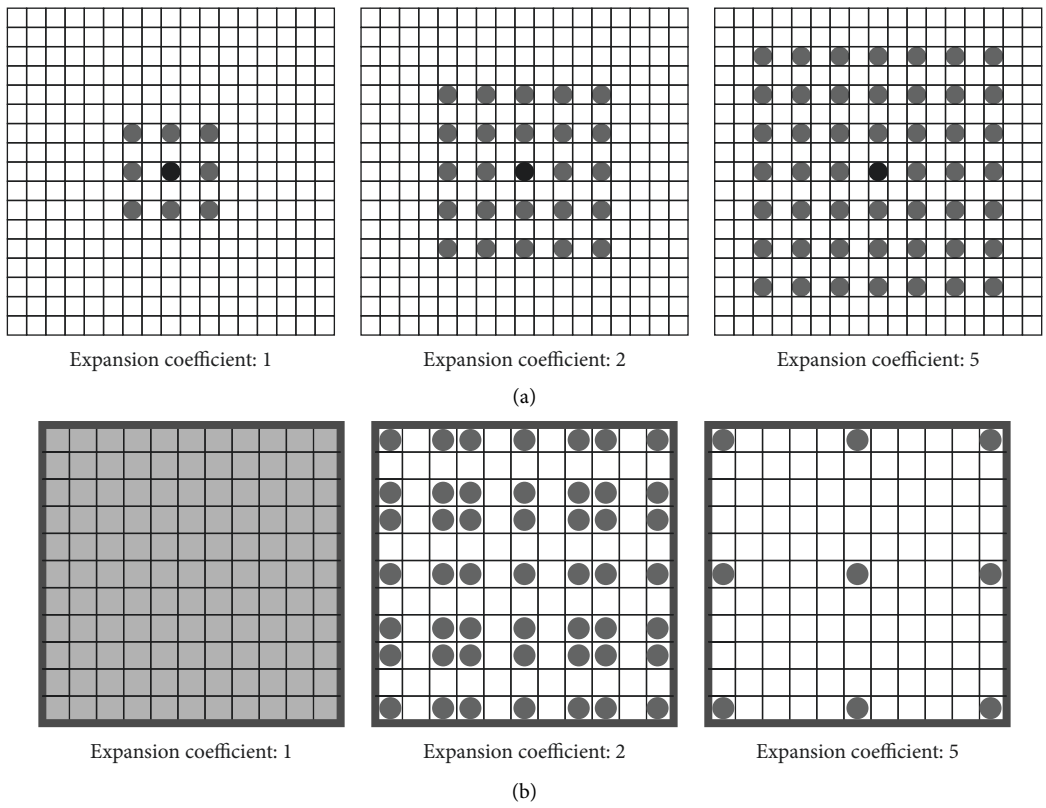

Figure 8: Feature extraction of the expanded convolution module with a sawtooth structure. (a) Standard dilated convolution feature extraction. (b) Zigzag structure fusion convolution feature extraction.

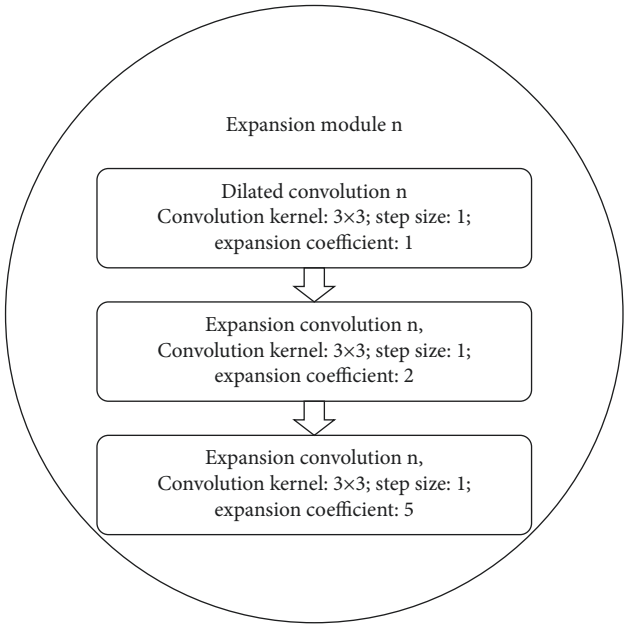

Figure 9: The structural diagram of the expanded convolution module with a sawtooth structure.

Table 1: The number of convolution kernels in each layer of the zigzag structure expansion convolution module.

Number of modules	Level number		
	$n1$	$n2$	$n3$
1	32	32	64
2	128	64	128
3	256	128	256
4	512	256	512
5	1024	512	1024

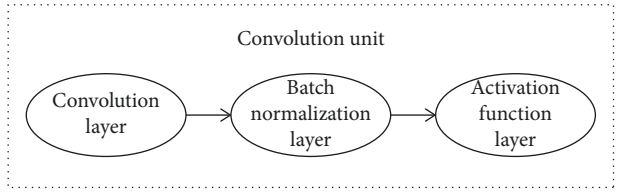

Figure 10: Convolution unit structure diagram.

layer, batch normalization layer, and activation function layer. Among them, the activation function uses Leaky ReLU. Each residual module can integrate the shallow semantic information and deep semantic information inside the module, connect each subsegment network with a shortcut method, integrate large-size, low-dimensional features and small-size, high-dimensional features to improve the accuracy of multiscale target recognition, control gradient propagation, and prevent gradient dispersion or gradient explosion. The residual module structure in UFEN is shown in Figure 11. UFEN has a total of 5 residual modules, which are, respectively, connected to the 5 expansion convolution modules. The first

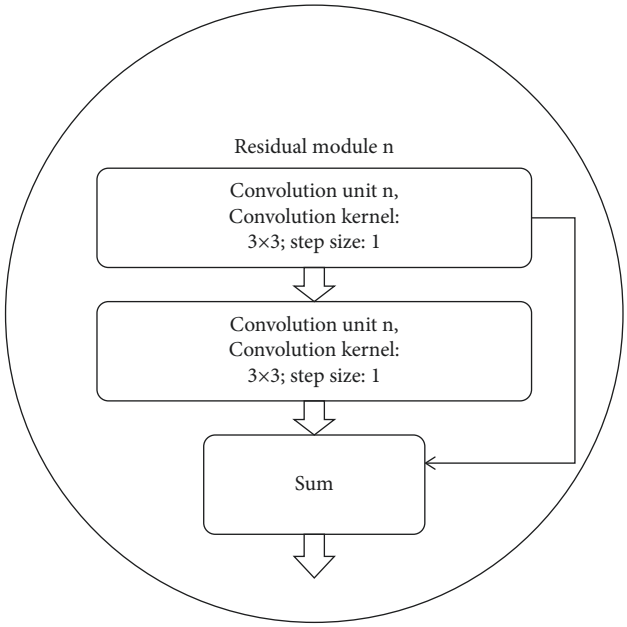

Figure 11: Structure diagram of residual convolution module.

residual unit of the residual module will downsample the picture once and will predict the picture in the last 3 times of downsampling. The cycle times of the 5 residual modules of UFEN are 1, 2, 8, 8, and 4, respectively. The specific number of convolution kernels is shown in Table 2.

2.4.2. UAV Target Recognition Network. After UFEN completes the feature extraction of the UAV target and generates 3 feature maps with scales of 13×13, 26×26, 52×52, respectively, AUNN will use the YOLO V3 target recognition network to identify the feature map group.

The YOLO V3 target recognition network will mesh the recognition area on the input feature map, and the number of grid divisions corresponds to the size of the input feature map. The anchor box in each grid in the feature map is responsible for identification and detection, and the information N contained in each grid is expressed as follows:

$$N = \left[b_x, b_y, b_w, b_h, p_0, p_1, \ldots, p_c\right] \times B, \tag{30}$$

where b_x, b_y, b_w, b_h are the coordinates and size information of the central point of the current prediction frame; p_0 is whether the target is included in the current grid and the accuracy of the target location; $p_0, p_1, \ldots, p_c$ is the probability that the target in the frame belong to the type to be identified. If the target center falls in the semantic information pixel point of a certain feature map, the grid will detect the target in this area, B is the number of anchor boxes, and the confidence value Po is the product of the probability of detecting the target and IOU (intersection over union), which is shown in the following formula:

$$\text{confidence} = P(\text{object}) \times \text{IOU}_{\text{pred}}^{\text{truth}}, \tag{31}$$

where P (object) is whether there is a target in the grid. If it exists, the value is 1, and the value is 0 if it does not exist;

Table 2: The number of convolution kernels in each layer of the residual convolution module.

Number of modules	Level number	
	$n1$	$n2$
1	32	64
2	64	128
3	128	256
4	256	512
5	512	1024

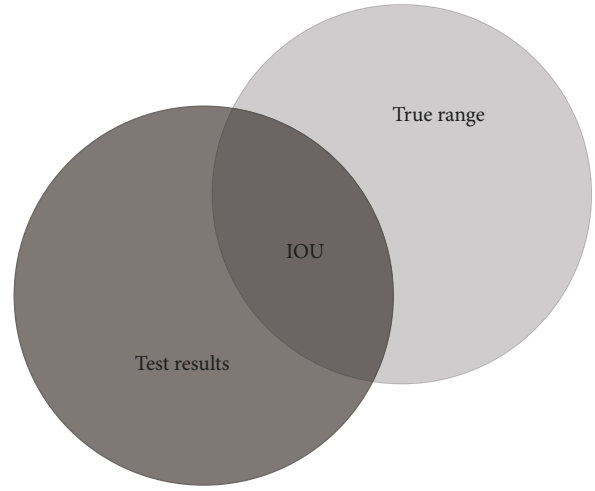

Figure 12: Schematic diagram of IOU.

IOU is the occurring simultaneous ratio, that is, the truth frame truth generated by the target and the range frame pred generated by the target recognition, the expression is shown in the following formula:

$$\text{IOU} = \frac{\text{DR} \cap \text{GT}}{\text{DR} \cup \text{GT}}. \tag{32}$$

In the formula: DR is the detection result; GT is the ground truth, and the result of the intersection of the detection target and the ground truth is shown in Figure 12.

By calculating the IOU between the detection target range frame and the ground truth, the network can distinguish the foreground target from the background target.

For feature maps at different scales, when each divided grid classifies internal targets, it needs to predict the class probability of the internal c targets, that is, the probability of the i-th target falling in the grid $P(\text{Class}_i|\text{object})$:

$$\begin{aligned} P\left(\text{Class}_i|\text{object}\right) &= P(\text{object}) \times \text{IOU}_{\text{pred}}^{\text{truth}} \\ &= P\left(\text{Class}_i\right) \times \text{IOU}_{\text{pred}}^{\text{truth}}. \end{aligned} \tag{33}$$

Each grid in the YOLO V3 network must first calculate whether there is a target inside. When it is determined that a target exists, the classification of the target will be judged according to its category prediction probability. When the target's prediction probability of a certain category exceeds the threshold and is greater than other classification prediction values, the target is considered as the current category.

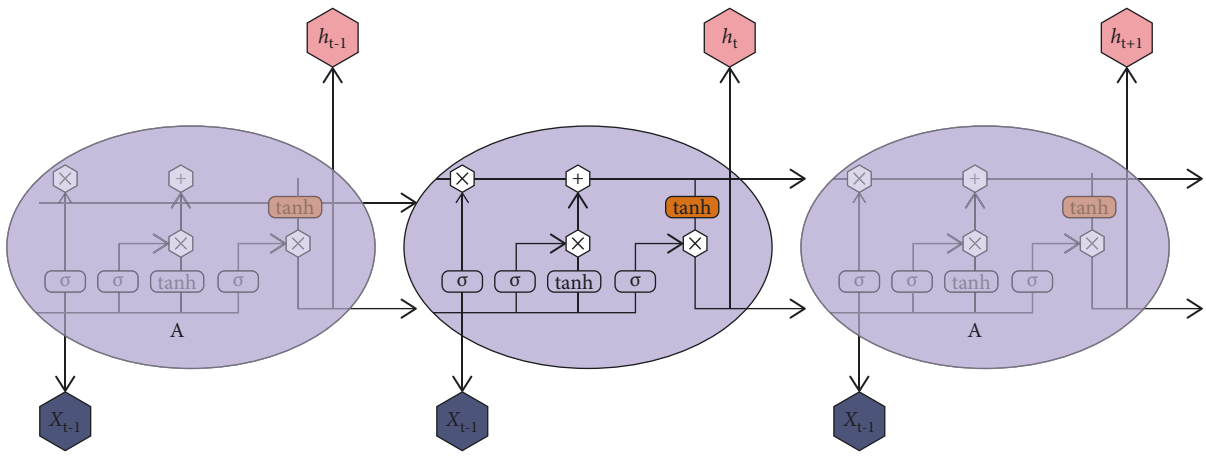

Figure 13: PSO algorithm structure.

For the judgment of the target's position information, YOLO V3 will continuously learn and correct the size of the anchor box through fine-tuning iteratively so that the result of the prediction frame is close to the truth frame. The adjustment process is shown in the formula:

$$\begin{cases} b_x = \sigma(t_x) + c_x, \\ b_y = \sigma(t_y) + c_y, \\ b_w = p_w e^{t_w}, \\ b_h = p_h e^{t_h}. \end{cases} \tag{34}$$

In formula (34): c_x and c_y are the coordinates of the upper left corner of the corresponding grid on different scale feature maps; t_x, t_y, t_w and t_h are the deviation between the prediction box and the truth frame; p_w and p_h are the length and width dimensions of the anchor box, and finally, the coordinates b_x and b_y of the upper left corner of the prediction frame and the corresponding length and width b_w and b_h of the prediction box are obtained.

The UAV trajectory prediction network will take the UAV space historical time domain motion trajectory as input data, use the long and short term memory network PSO algorithm to learn the flight behavior characteristics of the UAV, and use the image space position in the existing time domain to predict the iteratively, position of the image space in the future time domain.

Long and short term memory network is an RNN optimized network that overcomes the problems of gradient explosion and gradient disappearance. Compared with the traditional recurrent neural network (RNN), LSTM has an additional "forget gate" mechanism, which determines whether to forget the content of the moment through the correlation between the input and output at a certain moment and the previous moment, so that only important information is retained in all periods [15]. The structure of the PSO algorithm unit is shown in Figure 13.

x_t is the current UAV image space motion trajectory of input in the time domain; h_t is the UAV image space prediction motion trajectory of output in the next time domain; A is the calculation unit in the PSO algorithm. Each unit is connected from beginning to end, the calculating unit in the same layer will use the output of the previous layer as the input of the next layer; σ (sigmoid) and tanh are the activation functions.

A reshaped trajectory matrix C will be introduced from left to right in the LSTM unit. For the input information at time t, the matrix passed in from the left end in the LSTM unit is C_{t-1}, and the matrix passed out from the right end is C_t. Among them, the C_{t-1} matrix is multiplied by a coefficient by the multiplier, and then linearly superposed by the adder, and finally C_t is obtained.

The h_{t-1} matrix on the left is connected with the input h_t matrix, and the coefficient f_t is calculated by the sigmoid function through a linear unit. The coefficient is the multiplier coefficient in the x matrix transfer process. The expression is shown in the following equation:

$$f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right), \tag{35}$$

where W_f and b_f are the undetermined coefficients to be learned during the training process. In the "forget gate" of LSTM, if the output of the sigmoid function is 1, the input will be completely remembered; if the output is 0, the input will be completely forgotten; if it is the intermediate value of 0~1, the input will be remembered in proportion.

Finally, LSTM passes the input information through a "forget gate" again to generate an output h_t. h_t generated has two parts. One part is passed to the same layer unit, and the other is passed to the next layer unit. Then the final output predicted trajectory at time t of the LSTM unit is as follows:

$$h_t = \sigma\left(W_t\left[h_{t-1}, x_t\right] + b_t\right) \times \tan h C_t. \tag{36}$$

Through the above calculations, AUNN has been able to obtain the category information and location information of the target UAV, and project the center point of the continuous frame recognition position of the target UAV to the time domain coordinates to obtain the historical time domain motion trajectory of the UAV image space, and use the PSO algorithm to predict the position of the UAV in the future time domain. The feedback flow of the calculation process is shown in Figure 14.

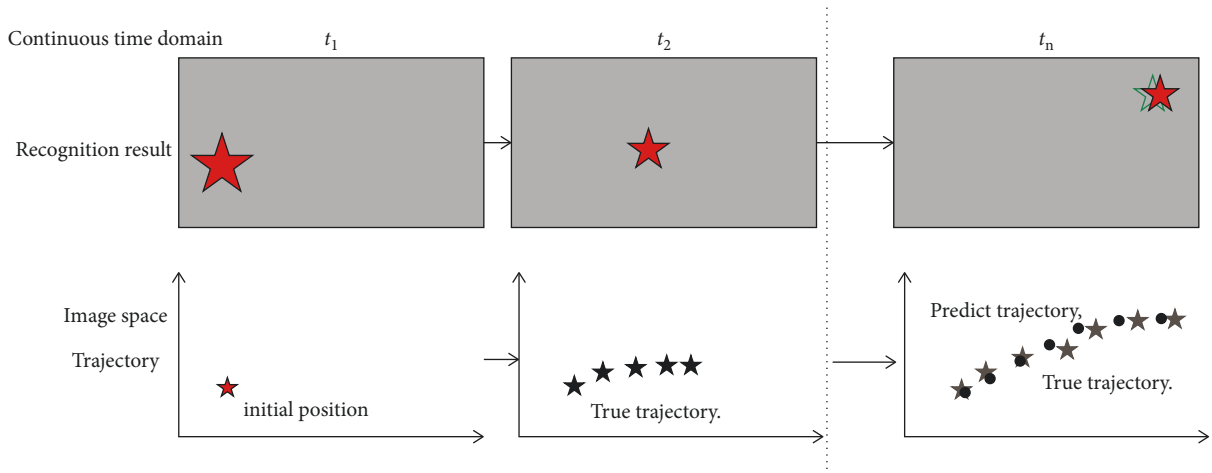

Figure 14: UAV image space history and time domain motion trajectory mapping method.

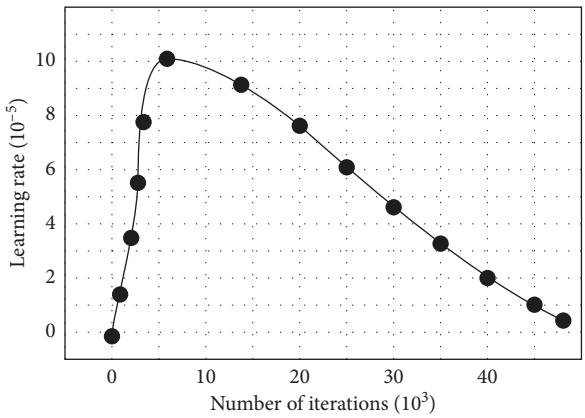

Figure 15: AUNN network learning rate curve.

Table 3: UAV data set structure.

Category	Number of pictures
Attack UAV	373
Payload/Command UAV	190
Scout UAV	197
Total	760

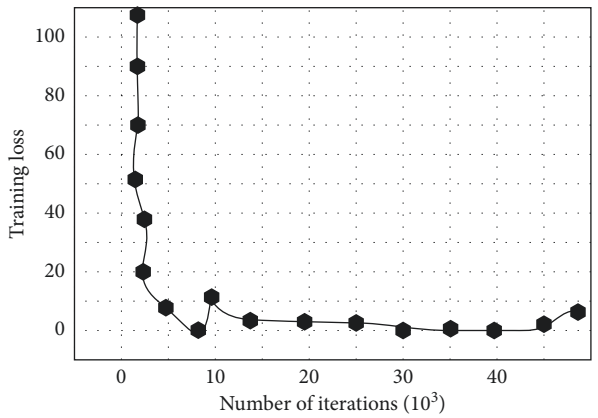

Figure 16: AUNN network loss curve.

3. Experimental Results and Analysis

The initial learning rate in the training phase is set to 0.001, and the learning rate is gradually increased in the first two generations. When the number of iterations is 380 times, the learning rate is reduced until the learning rate reaches 0.000001, which will not decrease, by which the loss function is further converged, and the learning rate curve is shown in Figure 15.

In order to verify the computing power of AUNN, an image data set for aerial UAV recognition was constructed based on the military and civilian fields. There are a total of 760 images in the data set, which are classified into reconnaissance, payload/control, and offensive targets according to the UAV's structure, functions, and executable tasks. The data components of each type of UAV are shown in Table 3.

The verification platform is DELLZ840, the CPU configuration of the central processing unit is Intel Xeon E5-2643 V3, the main frequency is 3. 4 GHz, the GPU is Quadro P5000, and the running memory is 32 GB, and the computing environment is Ubuntu 18.04. During the test, the IVFNN programming language was Python 3.7, with Tensorflow 2.0 and Opencv 3.2 as auxiliary high-level APIs.

After AUNN completes training, the total loss curve of the network is shown in Figure 16. After about 40,000 iterations, the final loss stabilizes at about 0. From the loss curve, it can be seen that the results of AUNN network training are relatively ideal.

AUNN's network training curve has a good convergence state, without gradient explosion, dispersion, disappearance, etc., which proves that the network designed by this research has a good feature learning ability. The average target recognition rate of AUNN is 82%, the average trajectory prediction rate is 80%, and the calculation speed is 24 frames/s. The network's UAV recognition and trajectory prediction effects are shown in Figure 17. Figures 17(a)~17(c) recognition effect of the 17 Figure network on command UAV, scout UAV, and attack UAV; Figures 17(d)~17(f)

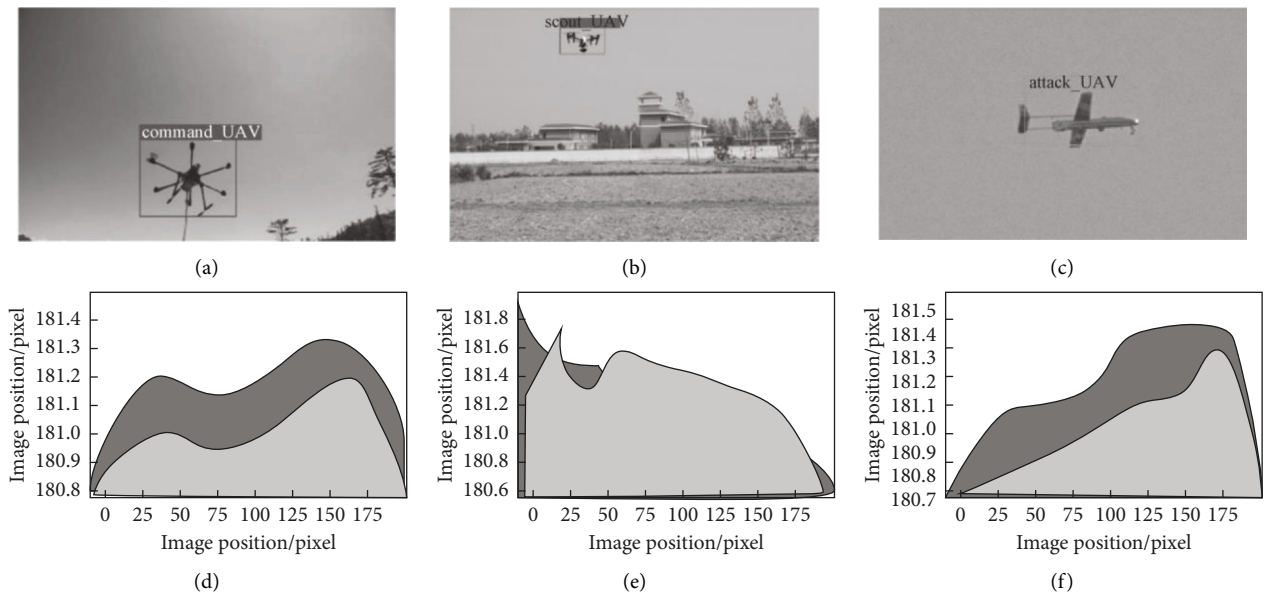

Figure 17: UAV recognition and trajectory prediction effect of AUNN network. (a) Recognition result of command UAV (b) Recognition result of scout UAV (c) Recognition result of attack UAV (d) Trajectory prediction result of command UAV (e) Trajectory prediction result of scout UAV (f) Trajectory prediction result of attack UAV.

correspond to the attacca trajectory prediction effect of 3 types of UAVs.

According to the results shown in Figure 17, AUNN can accurately identify and locate multitarget UAVs in the air under a ground-based platform. In the meantime, it can predict the UAV image space in the future time domain based on the current UAV image space trajectory to achieve rapid and accurate early warning for the "low, small, slow" UAV target, and provides machine vision support for the anti-UAV system.

4. Conclusion

The imbalance, turbidity, and instability of the high-performance computing early warning neural network in the use process are more important in the daily use process. The PSO algorithm in this paper can realize the optimization processing of the support vector regression model. While constructing the network high-performance computing early warning model, it uses three sets of different time granular data in the MAWI data set to carry out a detailed analysis with sample analysis. Finally, the experimental results show that the method proposed in this paper has better early warning performance for high-performance computing and can effectively solve the early warning problem of network high-performance computing. Finally, the flight state curve of the UAV is obtained through simulation, which verifies the effectiveness of the control law design. Therefore, this method can solve the flight control problem of UAV in complex situations.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research study was sponsored by the Project of Science & Technology Research and Development Plan of China Railway Corporation, the project number is J2020S001. The authors thank the project for supporting this article.

References

[1] W. Lu, X. Xu, F. Lu et al., "Resource optimization in anti-interference UAV powered cooperative mobile edge computing network," *Physical Communication*, vol. 42, no. 6, Article ID 101128, 2020.

[2] N. Jiang, K. Wang, X. Peng et al., "Anti-UAV: a large-scale benchmark for vision-based UAV tracking," *IEEE Transactions on Multimedia*, vol. 7, no. 3, p. 1, 2021.

[3] J. Chen, "Research on intelligentized anti-uav command control scheme technology," *E3S Web of Conferences*, vol. 51, no. 17, pp. 489–494, 2021.

[4] Z. Zhou, C. Cheng, X. Zhang, and W. Bian, "Numerical simulation and optimization on launching process of an anti-uav capture gun," *Dandao Xuebao/Journal of Ballistics*, vol. 31, no. 2, pp. 304–314, 2019.

[5] L. Yang, Q. Sun, and Z.-S. Ye, "Designing mission abort strategies based on early-warning information: application to uav," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 277–287, 2020.

[6] W. Chen, H. Wu, Wu et al., "Design and analysis for early warning of rotor uav based on data-driven dbn," *Electronics*, vol. 8, no. 11, p. 1350, 2019.

[7] W. Y. Gao, "Mine geological disaster early warning model based on uav aerial photography technology," *World Nonferrous Metals*, vol. 119, no. 4, p. 57, 2019.

[8] B. Wu, R. Fu, J. Chen, J. Zhu, and R. Gao, "Research on natural disaster early warning system based on uav technology," *IOP Conference Series: Earth and Environmental Science*, vol. 787, no. 1, Article ID 012084, 2021.

[9] H. E. Cheng, L. Zhenyi, L. I. Jin, Y. Wang, and L. Shu, "Research on measuring wind speed and direction of rotor uav," *Bulletin of Surveying and Mapping*, vol. 20, no. 13, pp. 1–14, 2019.

[10] X. Hu, K. K. Wong, K. Yang, and Z. Zheng, "Uav-assisted relaying and edge computing," *Scheduling and Trajectory Optimization*, vol. 14, no. 16, pp. 136–154, 2018.

[11] D. L. Tan and Z. H. Liu, "Effective civil uav management and control system," *Communications Technology*, vol. 20, no. 3, pp. 1160–1169, 2019.

[12] G. Xiao, Q. Wang, G. Liu, Y. Pan, and N. G. Party, "Method and application of extracting fracture information from high and steep dangerous rock based on uav image," *Site Investigation Science and Technology*, vol. 7, no. 2, pp. 1–12, 2019.

[13] Q. Wang, B. Zhang, Z. Zhou, J. Sun, and X. University, "Research on detection and alarm of air threat situation for uav collision avoidance," *Advances in Aeronautical Science and Engineering*, vol. 101, no. 1, pp. 152–160, 2019.

[14] Z. Yang and J. Wang, "A new air quality monitoring and early warning system: air quality assessment and air pollutant concentration prediction," *Environmental Research*, vol. 158, pp. 105–117, 2017.

[15] C. N. Vong, L. S. Conway, J. Zhou, N. R. Kitchen, and K. A. Sudduth, "Early corn stand count of different cropping systems using UAV-imagery and deep learning," *Computers and Electronics in Agriculture*, vol. 186, no. 5, pp. 106214–106281, 2021.