ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Exploring the Application of Artificial Intelligence in Drone Target Recognition and Tracking

Zhenhua Li[1,*]

[1] China People's Police University, Langfang, Hebei, 065000, China

SUMMARY: *This paper first designs an object recognition algorithm based on YOLOv3. By incorporating an improved temporal-spatial context similarity feature fusion structure, the accuracy of object location prediction is enhanced. Combining the residual modules of the Darknet53 backbone with a multi-scale detection architecture optimizes object classification performance. An improved SiamFC++ object tracking algorithm is proposed to strengthen tracking robustness and consistency. Experiments validated using the VisDrone2023 dataset and multiple aerial video sequences demonstrate that our model outperforms several alternative models in both overlap rate and center error metrics. Notably, in the person1 scenario, it achieves an average coverage rate of 82.1% with a center position error of only 11.4 pixels, effectively handling scenarios involving target occlusion and rapid motion.*

KEYWORDS: *UAV; Object Recognition; Object Tracking; YOLOv3; SiamFC++*

1 Introduction

As a significant aerospace concept, unmanned aerial vehicles (UAVs) have found extensive applications across various fields. Target recognition and tracking constitute critical components of UAV operations. By accurately identifying and monitoring targets, UAVs can play a vital role in military, civil, and commercial sectors. The rise of artificial intelligence has further elevated the application capabilities of UAVs [1-5].

AI technologies can be applied to pattern recognition and tracking in UAVs [6]. For instance, UAVs can capture target features and contours through image recognition, then analyze and process this data using machine learning and deep learning techniques to achieve target tracking and early warning [7-9]. In UAV scenarios, this technology is particularly suited for target search and engagement, border patrol, and similar applications.

AI technology also empowers drones with autonomous navigation and obstacle avoidance capabilities [10]. During flight, drones equipped with sensors and cameras can perceive their environment. Image recognition technology identifies obstacles, which are then processed through machine learning algorithms to execute autonomous obstacle avoidance [11-14]. In military applications, autonomous obstacle avoidance technology is widely integrated into combat aircraft, effectively reducing enemy attacks while enhancing combat efficiency and precision [15-17]. In civilian sectors, this technology supports drone patrols, environmental monitoring, search-and-rescue operations, and other applications [18, 19].

AI further enables intelligent control and combat capabilities for drones [20]. By integrating smart chips, autonomous control algorithms, and autonomous controllers, drones achieve multiple functions including autonomous vehicle control, path planning, and specific task

*huanglei950531@163.com
https://doi.org/10.65102/is2026637
Published: 30 April, 2026

execution [21, 22]. Under such conditions, both the combat capabilities and risk levels of drones can be effectively managed. In civilian applications, intelligent control and combat technologies for drones can also be widely applied in outdoor operations such as agriculture, forestry, and fisheries [23, 24].

Reference [25] proposes a novel automatic target recognition method for unmanned aerial vehicles based on a backpropagation-artificial neural network algorithm. This approach aims to optimize the structure of the backpropagation network, thereby enhancing efficiency and reducing recognition time. Reference [26] highlights the positive impact of aerial vehicles and artificial intelligence on wildlife monitoring and conservation efforts. It introduces a system incorporating thermal imaging acquisition capabilities and a video processing pipeline to achieve target detection, classification, and tracking of wildlife. Reference [27] analyzes data analysis methods and AI technologies applied to UAVs, developing an image processing application using convolutional neural network algorithms for object recognition. It highlights how AI applications enhance UAV operational capabilities. Reference [28] systematically reviews research outcomes on UAV detection and tracking, examining radar-related technologies combined with various machine learning and deep learning approaches. It emphasizes that machine learning-based UAV detection and tracking methods show potential but remain imperfect. Reference [29] highlights the widespread adoption of UAVs in civil and military domains, exploring deep learning applications within an AI framework to enhance detection and tracking performance. Reference [30] discusses AI integration in UAVs, underscoring its role in advancing the field while improving flight safety and efficiency. Reference [31] highlights that the proliferation of UAVs poses risks to public safety and privacy protection, underscoring the necessity for efficient UAV detection systems. It proposes a hybrid approach combining two AI-based methods to enhance system accuracy.

This paper first proposes an improved YOLOv3 model. By incorporating a temporal context similarity feature fusion structure, it strengthens the contextual relevance of target location predictions. The residual module design of the Darknet53 backbone addresses the degradation issues inherent in deep neural networks. The multi-scale detection architecture is optimized to enhance small object recognition capabilities. A SiamFC++-based object tracking algorithm is designed, utilizing anchor-free detectors to prevent false matching of small targets. Experimental validation using public datasets and multiple aerial video sequences quantitatively assesses the proposed model's effectiveness across three dimensions. Ablation studies confirm the specific contributions of key components.

2 AI-Driven Drone Target Recognition and Tracking

With the rapid advancement of artificial intelligence technology, its applications in computer vision have become increasingly widespread, demonstrating significant advantages particularly in drone target recognition and tracking tasks. Drones, leveraging their agile maneuverability, high-resolution imaging capabilities, and diverse deployment scenarios, have become essential tools across numerous fields including military reconnaissance, disaster monitoring, traffic management, and agricultural inspection. However, when performing target recognition and tracking tasks, UAVs often face challenges posed by complex dynamic environments. Traditional methods struggle to meet practical demands in terms of real-time performance, robustness, and accuracy. Therefore, integrating artificial intelligence technologies, especially deep learning methods, has become a key pathway to enhance the performance of UAV target recognition and tracking.

This paper systematically explores the design and optimization of deep learning-based

target recognition algorithms and tracking models, aiming to improve UAVs' target perception and continuous tracking capabilities in complex scenarios through technological innovation.

2.1 Multi-Object Detection Algorithm Based on YOLOv3

2.1.1 Target Prediction Based on Time-Context Feature Fusion

To address drastic changes in target positions, enhancing the learning of contextual target position relevance is crucial for the algorithm to accurately predict target locations. Since the maximum response point in the similarity map reflects target position information, and similarity feature fusion enables the algorithm to learn contextual target position associations, this section proposes two time-context-based similarity feature fusion structures on top of TCTrack to improve the algorithm's target position prediction accuracy. These include: (1) Decoder-based contextual similarity feature fusion

(2) Average pooling-based contextual similarity feature fusion.

(1) Decoder-based contextual similarity feature fusion

To efficiently learn prior knowledge from contextual similarity graphs, this section designs a decoder-based temporal contextual similarity feature fusion, whose structure is shown in Figure 1. Given the original similarity graph R_t at time step t, a 1×1 convolution operation is applied to obtain the similarity map F_t. Then, based on the multi-head attention function $MultiHead(\cdot,\cdot,\cdot)$, we construct the global dependencies of similarity features in F_t to enhance the algorithm's focus on important regions within F_t, thereby obtaining the deep similarity graph F_t^1. The specific formula is:

$$\begin{cases} F_t^1 = Norm\left(MultiHead(F_t,F_t,F_t)+F_t\right), \\ MultiHead\left(Q,K,V\right)=Cat\left(H_1,H_2,\ldots,H_N\right)W, \\ H_i = softmax\left(\dfrac{QW_i^Q(KW_i^K)^T}{\sqrt{d}}\right)VW_i^V, 1\le i\le N, \\ softmax\left(x\right)=\left[\dfrac{e^{x_1}}{\sum_{j=1}^{C}e^{x_j}},\ldots,\dfrac{e^{x_C}}{\sum_{j=1}^{C}e^{x_j}}\right], x=[x_1,x_2,\ldots,x_C]. \end{cases} \tag{1}$$

To prevent loss of critical information from the original similarity features, the original similarity graph map F_{t-1} at time $t-1$ is fused with the output similarity graph F_{t-1}^{*} are fused. The prior knowledge F_{t-1}^{**} is then output via average pooling downsampling $AvgPool\left(\cdot\right)$, enhancing the inference speed of the tracking algorithm. Here, F_{t-1}^{**} can be expressed as:

$$F_{t-1}^{**} = AvgPool\left(Norm(F_{t-1}+F_{t-1}^{*})\right) \tag{2}$$

Based on the prior knowledge F_{t-1}^{**} at time $t-1$, the network focuses on important features in the deep similarity graph F_t^1 at time t through $MultiHead(\cdot,\cdot,\cdot)$ and outputs F_t^2. The specific calculation formula is:

$$F_t^2 = Norm\left(MultiHead(F_t^1, F_{t-1}^{**}, F_{t-1}^{**}) + F_t^1\right) \tag{3}$$

Finally, based on $FFN(\cdot)$, the output similarity graph F_t^* at time t is obtained through the following calculation process:

$$\begin{cases} F_t^* = FFN(F_t^2), \\ FFN(X) = Norm\left(Linear(ReLu(Linear(X))) + X\right). \end{cases} \tag{4}$$

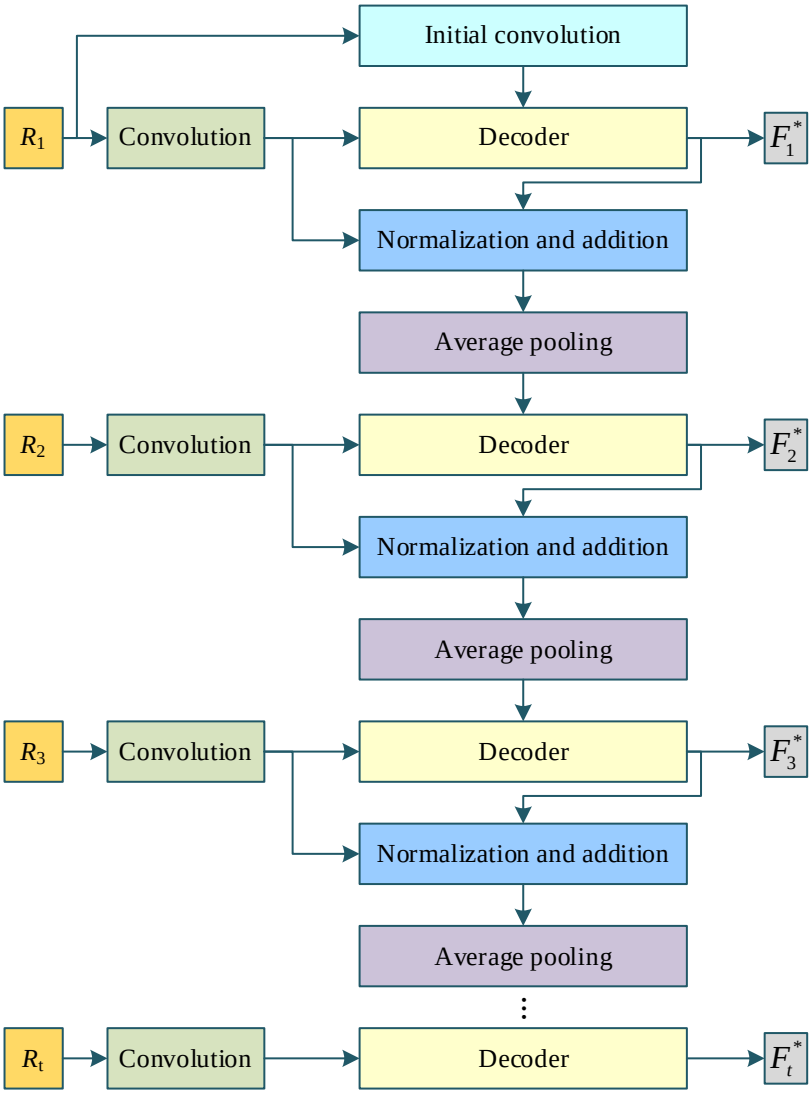

Figure 1: Time context similarity feature fusion structure based on decoder

Compared to the TCTrack tracking algorithm, this structure eliminates the original encoding process. Instead, it directly performs decoding using the prior knowledge F_{t-1}^{**} from the previous time step and the current similarity map F_t, thereby reducing the number of network parameters. Furthermore, average pooling reduces the spatial scale of prior knowledge while preserving the overall information of F_{t-1}^{**}, enabling the model to learn smoother similarity

maps and enhancing training stability. Ablation experiments demonstrate the structure's efficiency.

(2) Time-Contextual Similarity Fusion Based on Average Pooling

Since the network only learns similarity features from the previous time step, the limited scope of prior knowledge reduces the algorithm's ability to learn target motion characteristics. To address this issue, we propose a time-contextual similarity fusion structure based on average pooling. This structure performs feature transformations on the encoded similarity vectors F_{t-1}^{m}, F_{t-2}^{m}, and F_{t-3}^{m} from the previous three frames via convolutional operations, yielding vectors F_{t-1}^{C}, F_{t-2}^{C}, and F_{t-3}^{C} through convolutional operations. Subsequently, average pooling is applied to obtain the prior knowledge F_{t-1}^{f}. The specific computation process can be expressed as:

$$F_{t-1}^{f} = AvgPool\left(Norm\left(F_{t-1}^{C} + F_{t-2}^{C} + F_{t-3}^{C}\right)\right) \tag{5}$$

Then, the similarity graph R_t at time t is convolved to obtain the similarity graph map F_t. Subsequently, F_{t-1}^{f} and F_t undergo encoding through the encoder module. To enhance the attention of the similarity graph map F_t at time t toward the historical prior knowledge F_{t-1}^{f}, F_{t-1}^{f} and F_t are processed through the multi-head attention function $MultiHead(\cdot,\cdot,\cdot)$ and the feedforward network function $FFN(\cdot)$ to obtain the similarity graph F_t^{m} incorporating historical prior knowledge. The specific computational formula is as follows:

$$\begin{cases} F_t^{1} = Norm(MultiHead(F_t, F_{t-1}^{f}, F_{t-1}^{f}) + F_t), \\ F_t^{m} = FFN(F_t^{1}). \end{cases} \tag{6}$$

Next, the encoded similarity map F_t^{m} undergoes decoding to deepen the algorithm's understanding of the similarity features at the current time step. Specifically, F_t first obtains an intermediate vector F_t^{s} through self-attention. This vector then undergoes cross-attention with F_t^{m} and passes through a feedforward network function to produce the output similarity map F_t^{*}. As shown in the following formula:

$$\begin{cases} F_t^{s} = Norm(MultiHead(F_t, F_t, F_t) + F_t), \\ F_t^{d} = Norm(MultiHead(F_t^{s}, F_t^{m}, F_t^{m}) + F_t^{s}), \\ F_t^{*} = FFN(F_t^{d}). \end{cases} \tag{7}$$

In summary, The decoder-based contextual similarity feature fusion leverages only the decoder to enable efficient learning of historical similarity features, balancing inference speed and tracking accuracy. The mean pooling-based temporal contextual similarity feature fusion sets the historical similarity maps from three frames as prior knowledge, then encodes and decodes the current similarity map via a Transformer. This expands the algorithm's learning horizon for historical similarity features and enhances contextual learning of target location relevance. Subsequent ablation experiments demonstrate that the first similarity feature fusion structure achieves optimal inference speed, while the second structure exhibits superior tracking accuracy compared to the first.

2.1.2 YOLOv3 Network Backbone

The backbone of YOLOv3 is Darknet53, comprising 53 convolutional layers (excluding the two convolutional layers within the residual structure). In contrast, the YOLOv2 backbone contains only 17 convolutional layers. This increased network depth enables YOLOv3 to extract deeper features, resulting in higher accuracy for object detection tasks.

The convolutional module in YOLOV3 consists of a convolutional layer, a batch normalization layer, and an activation function layer. YOLOV3 omits pooling layers; feature map downsizing is achieved by traversing the feature map with a convolutional kernel at a stride of 2. Each subsampling operation halves the feature map size, with a total of five subsampling steps performed. Batch normalization regularizes the output of convolutional layers, ensuring data entering subsequent layers shares consistent distributions. This prevents gradient vanishing issues that arise as neural networks deepen. YOLOv3 employs the widely adopted, computationally simple LeakyReLU activation function.

The residual module represents one of the major improvements in YOLOv3. While batch normalization mitigates the issue of gradient vanishing caused by increased network depth, it does not resolve the degradation problem—where adding more layers fails to improve recognition accuracy, and deep networks actually perform worse on the training set than shallow ones. To address the aforementioned issues, YOLOv3 adopted the skip-connection architecture from ResNet networks by introducing residual modules, enabling significant increases in network depth while maintaining effective training.

One potential cause of network degradation is that if a shallow network with fewer layers represents the optimal solution for a task, the additional layers in a deeper network may function redundantly as identity mappings—where the output signal equals the input signal. At this point, the deep network functionally equates to the optimal shallow network. However, in practical applications, the redundant layers of deep networks often fail to accurately fit the identity mapping function, leading to the degradation phenomenon where deep networks perform worse than shallow ones.

The key to addressing network degradation in residual modules lies in implementing an identity mapping via skip connections. Taking the first residual module used after 2x downsampling in the YOLOv3 network as an example, its specific structure is shown in Figure 2. The residual module employs multiple 1×1 convolutional kernels to reduce the dimensionality of feature maps, followed by several 3×3 convolutional kernels to increase the dimensionality. The input to the residual module undergoes skip connections, and the output is concatenated as the input to the next convolutional layer. The mathematical expression for the residual module is:

$$H(x) = x + F(x, \omega) \tag{8}$$

In the equation: x is the input to the residual module, $H(x)$ is the output of the residual module, ω is the weight parameter of the convolution kernel, and $F(x, \omega)$ is the difference between the module's input and output, referred to as the residual term of the module. It can be observed that when this layer functions as an identity mapping layer, it only needs to learn how to adjust the weight parameter ω to make the residual term $F(x, \omega)$ equal to zero. Without the residual module, achieving the general identity mapping function $H(x) = x$ becomes challenging in deep neural networks, especially when the layer is located deep within the network and involves numerous parameters.

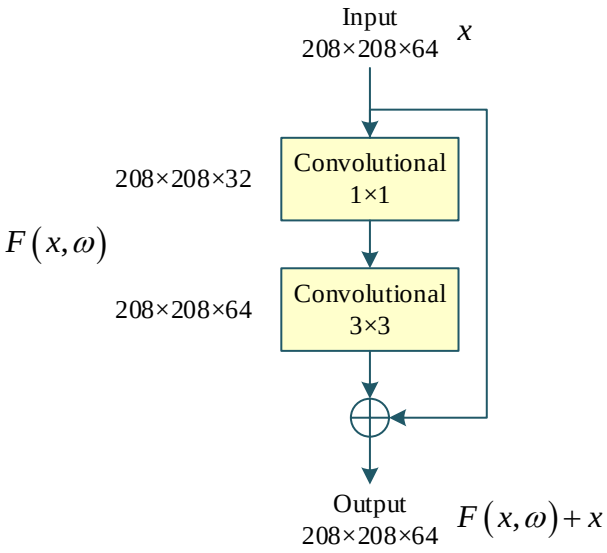

Figure 2: Residual Module

Similar to batch normalization, residual modules also help mitigate gradient vanishing during training. During backpropagation, the residual module's output yields the first-order partial derivative with respect to the input vector x as follows:

$$\frac{\partial H(x)}{\partial x} = 1 + \frac{\partial F(x,\omega)}{\partial x} \tag{9}$$

Due to the presence of the constant 1 in the above expression, even when the gradient of the residual term is small, the residual module does not suffer from gradient vanishing and cannot be trained.

2.1.3 YOLOv3 Network Detection Architecture

The specific detection architecture of the YOLOV3 network is shown in Figure 3. As illustrated in Figure 3, this architecture represents another significant improvement in the YOLOV3 network beyond its residual structure, effectively addressing the suboptimal small object detection performance observed in YOLO and YOLOV2. While YOLOv2 establishes a single object detection layer on the feature map output from the final convolutional layer, YOLOv3 constructs detection layers at three distinct scales. The network implements a dedicated detection layer at 8x downsampling for small object detection, alongside layers at 16x and 32x downsampling for medium and large object detection, respectively. Shallow feature maps possess higher resolution and richer positional information, enabling more precise object localization. Deep feature maps, conversely, contain richer semantic information beneficial for object classification. The network's feature pyramid architecture fuses deep feature maps with shallow ones after upsampling. This approach avoids excessive computational overhead while significantly enhancing the network's small object detection capabilities.

For the object classification subtask, YOLOv3 employs a Logistic function to establish multiple binary classifiers for target classification, abandoning the Softmax classifier used in previous versions. The Softmax classifier maps the input from the preceding layer to a distribution of [0, 1], where higher probabilities indicate greater likelihood of belonging to the corresponding category. The network selects the target category corresponding to the maximum probability as the output. The Softmax function assumes an object belongs to only one category,

but in some task scenarios, an object may belong to multiple categories. For example, in the hierarchical relationship between the “pedestrian” category and the subcategories ‘male’ and “female,” both categories can describe the object. In such scenarios, using the Logistic function to establish multiple binary classifiers for object classification is more suitable.

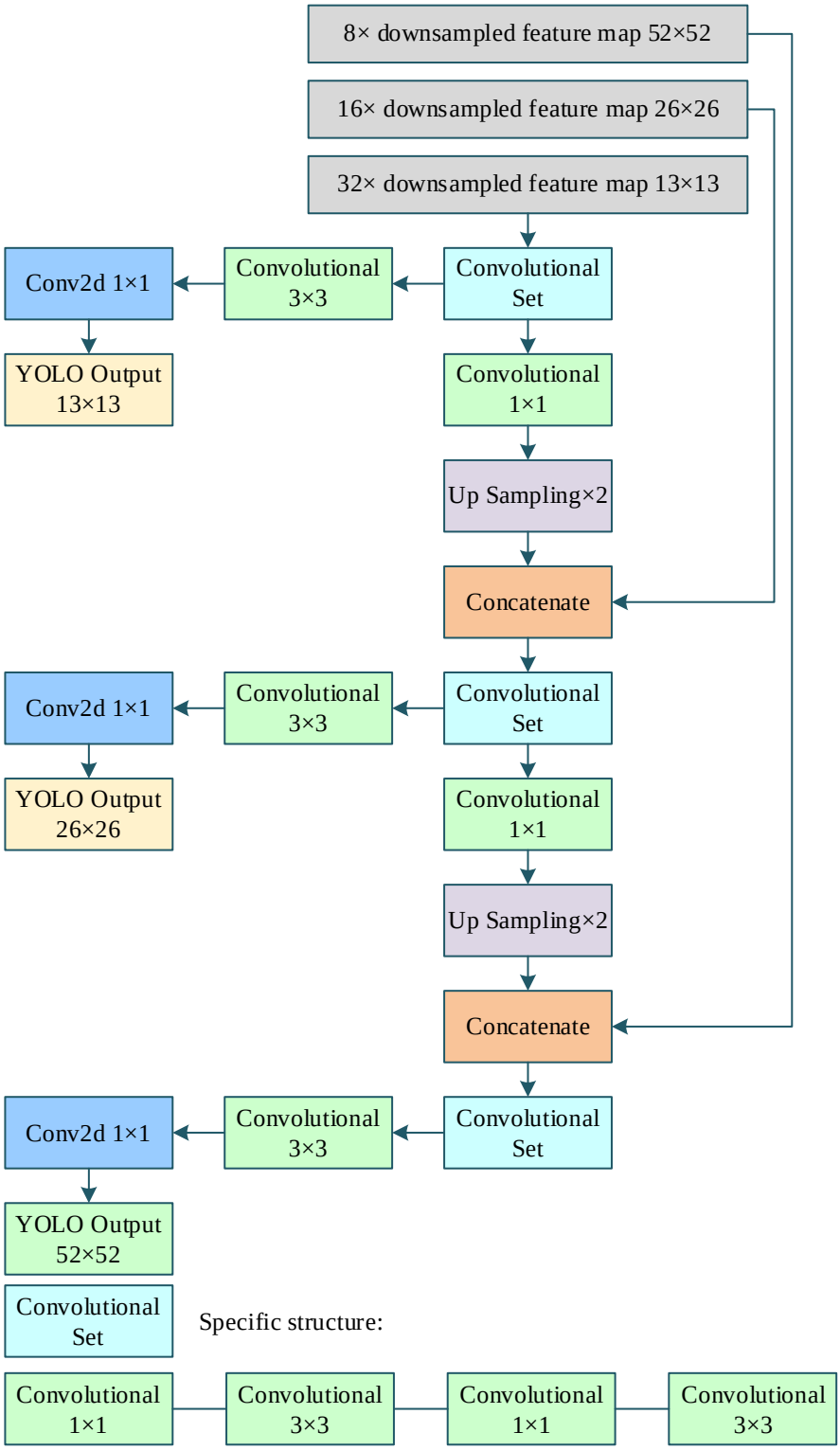

Figure 3: YOLOV3 Detection Architecture

2.2 Object Tracking Algorithm Based on SiamFC++

Visual tracking tasks can be viewed as a combination of classification and object estimation tasks. Specifically, the classification task yields a coarse location, while the estimation task provides the precise bounding box of the object. Although object tracking algorithms are advancing rapidly, the methods used for object state estimation vary significantly. These approaches can be broadly categorized into three types: The first type, exemplified by SiamFC, employs a crude multi-scale test. This method suffers from issues such as low efficiency and poor accuracy. The second category, exemplified by ATOM, first obtains a rough object location through the maximum confidence score from the classification module. It then generates multiple prior boxes using Gaussian noise based on the previously estimated object width and height. While this approach enhances accuracy, it also introduces additional challenges such as requiring more predefined hyperparameters and imposing a heavier computational burden. The third category, exemplified by the SiamRPN series, relies on Region Proposal Networks (RPNs) to generate candidate regions for precise target estimation. However, this approach necessitates predefined bounding boxes, significantly compromising model robustness. It also requires prior knowledge of target shapes, contradicting the fundamental goal of universal object tracking.

To address these issues, this paper proposes a set of design metrics for object trackers:

G1: Object tracking algorithms should perform two subtasks: classification and object location estimation. The classification branch separates objects from the background to enhance robustness, while the location estimation branch ensures tracking accuracy.

G2: Tracking algorithms must match the original image template, not predefined bounding boxes. Otherwise, the model's discrimination capability will decrease, leading to more misclassifications.

G3: A zero-prior tracking method should operate without any prior knowledge about the target, such as aspect ratio information. Incorporating such information into the model for discrimination will compromise its robustness;

G4: Directly using classification confidence for bounding box regression leads to performance degradation. An estimation quality score independent of classification should be employed.

Based on these principles, this paper designs a network architecture resulting in the SiamFC++ model.

The SiamFC++ feature extraction network employs AlexNet and GoogLeNet. First, SiamFC++ adds a position estimation branch parallel to the classification branch after the backbone network, with each branch utilizing distinct feature maps—addressing constraint G1. Furthermore, the model avoids predefined bounding boxes, satisfying constraints G2 and G3. Finally, a separate quality assessment branch is added to the classification branch, corresponding to criterion G4.

Regarding quality assessment scores, this paper employs two methods (10) for calculation:

$$PSS^* = \sqrt{\frac{\min(l^*, r^*)}{\max(l^*, r^*)} \times \frac{\min(t^*, b^*)}{\max(t^*, b^*)}} \tag{10}$$

Among them:

$$\begin{cases} l^* = \left(\frac{s}{2} + xs\right) - x_0, \quad t^* = \left(\frac{s}{2} + ys\right) - y_0 \\ r^* = x_1 - \left(\frac{s}{2} + xs\right), \quad b^* = y_1 - \left(\frac{s}{2} + ys\right) \end{cases} \tag{11}$$

where (x_0, y_0) and (x_1, y_1) denote the upper-left and lower-right corners of the true bounding box associated with point (x, y). s represents the total stride of the backbone network, set to 8 in the original text.

The second approach utilizes the IoU score:

$$IoU^* = \frac{Intersection(B, B^*)}{Union(B, B^*)} \tag{12}$$

where B is the predicted bounding box and B^* is the true target box. The quality assessment score is multiplied by the target prediction classification score to obtain the final target box selection score. This approach significantly reduces the weight assigned to bounding boxes far from the object's center, thereby improving tracking accuracy.

The model's optimization objective is:

$$\begin{aligned} L\left(\{p_{x,y}\}, q_{x,y}, \{t_{x,y}\}\right) = & \frac{1}{N_{pos}} \sum_{x,y} L_{cls}(p_{x,y}, c^*_{x,y}) \\ & + \frac{1}{N_{pos}} \sum_{x,y} 1\{c^*_{x,y} > 0\} L_{quality}(q_{x,y}, q^*_{x,y}) \\ & + \frac{1}{N_{pos}} \sum_{x,y} 1\{c^*_{x,y} > 0\} L_{reg}(t_{x,y}, t^*_{x,y}) \end{aligned} \tag{13}$$

Here, $1\{\cdot\}$ denotes the indicator function, which takes the value 1 if the condition within the parentheses holds, and 0 otherwise. L_{cls} represents the classification loss, $L_{quality}$ denotes the classification quality loss, and L_{reg} signifies the IoU loss for regression bounding boxes.

SiamFC++ is an anchor-free tracking algorithm. This successfully resolves the issue in anchor-based algorithms where significant changes in target object appearance lead to erroneously high scores being calculated for nearby objects and background. Another potential cause of this problem is that anchor-based algorithms match targets against prior boxes rather than the targets themselves, which can introduce drift and hinder robustness.

3 Experimental Analysis of UAV Target Recognition and Tracking

To validate the effectiveness of the designed model, the performance evaluation section conducted a series of experiments on the challenging VisDrone2023 public competition test set. The VisDrone2023 dataset comprises 7,037 training images, 611 test images, and 3,265 test images across 10 categories (pedestrians, people, cars, vans, buses, trucks, motorcycles, bicycles, rickshaws, and tricycles). The “pedestrians” and “people” categories are particularly prone to feature confusion, making this dataset highly suitable for research. For the overall model performance analysis section, this paper selected six distinct aerial video sequences for experimentation.

3.1 Target Recognition Performance Evaluation

3.1.1 Comparative Experiment

This paper comprehensively compares the performance differences between the proposed object detector and other state-of-the-art detectors, with the comparison results shown in Table 1. On the VisDrone2023 public test set, the object detector proposed in this paper achieves excellent performance of AP@0.5=0.395 and mAP=0.254, which is better than the current most advanced YOLO detector of the same scale in terms of accuracy, parameter efficiency and computational complexity. Additionally, this paper compares our object detector with recent lightweight detectors of similar scale, such as EfficientVit, Fasternet, and RT-DETR. Although our detector achieves slightly higher GFLOPS performance than these detectors, it demonstrates a significant lead in terms of accuracy and parameter utilization. Therefore, we conclude that this design possesses considerable practical value.

Table 1: Comparison of Performance Differences

Model	Epoch	mAP@0.5	mAP	AP	AP_{50}	AP_{75}	AP_S	AP_M	Params	$GFLOP_S$
EfficientVit	500	0.316	0.211	0.209	0.361	0.193	0.098	0.318	18.96M	49.3
Efficient Formerv2	500	0.337	0.235	0.213	0.382	0.214	0.104	0.326	30.92M	60.8
Fasternet	500	0.326	0.214	0.207	0.374	0.185	0.109	0.312	19.38M	55.4
Yolov8m	500	0.298	0.212	0.202	0.359	0.191	0.083	0.357	23.77M	80.1
SwinTrans	500	0.356	0.228	0.176	0.377	0.202	0.145	0.334	32.16M	105.4
Vanillanet	500	0.274	0.173	0.188	0.331	0.183	0.098	0.308	19.84M	123.7
RT-DETR	500	0.351	0.206	0.194	0.382	0.222	0.133	0.341	19.36M	69.2
CSwinTrans	500	0.354	0.221	0.211	0.376	0.195	0.142	0.339	27.66M	92.5
The proposed	500	0.395	0.254	0.232	0.403	0.251	0.157	0.367	15.23M	67.4

In addition, to conduct a more detailed analysis of the model's specific recognition performance, this paper generated confusion matrices for 10 categories. The model's confusion matrix is shown in Figure 4. It can be observed that the accuracy results for all categories are above 0.6.

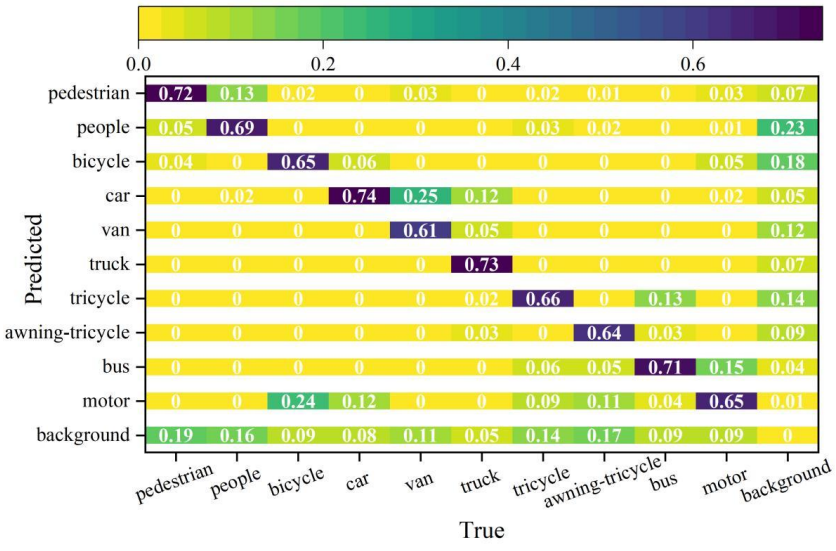

Figure 4: Confusion Matrix of the Model

3.1.2 Ablation Experiment

To validate the effectiveness of the time-contextual similarity feature fusion proposed in this paper, it was compared with three corresponding alternatives: TCTrack, TCTrack-decoder, and TCTrack-Avg-Pooling. The ablation experiment results are shown in Figure 5. Compared to other loss functions, the proposed method converges faster. Specifically, it achieves near-optimal performance around 310 epochs, whereas the other methods require 400 epochs or more. This represents a 22.5% improvement in convergence speed over the alternative loss functions.

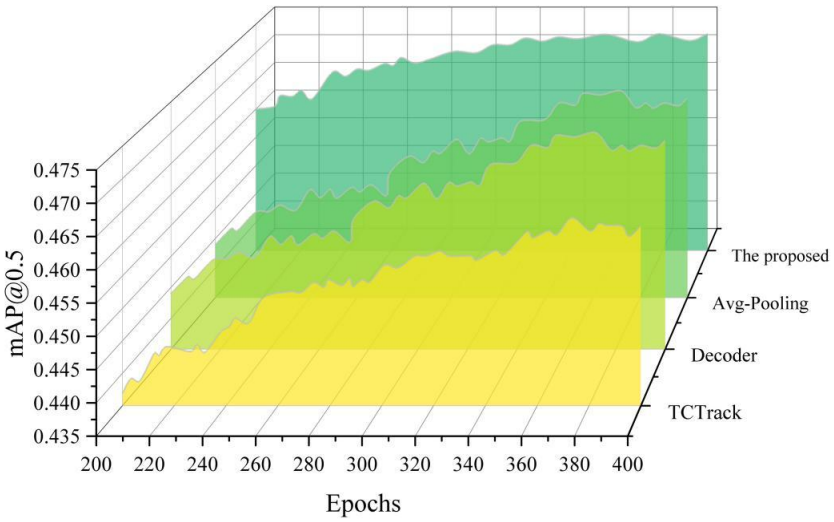

Figure 5: Results of the ablation experiment

3.2 Target Tracking Performance Evaluation

3.2.1 Comparative Experiments

This paper summarizes the results of SiamFC++ and state-of-the-art multi-object tracking methods on the VisDrone2023 dataset. The comparative experimental results are shown in Table 2. Compared with the current eight state-of-the-art tracking methods, SiamFC++ achieves the best performance. Experimental results demonstrate that this method holds certain advantages for multi-object tracking tasks in drone scenarios. Among the three comprehensive evaluation metrics, it achieves outstanding scores: 35.6 for MOTA, 85.2 for MOTP, and 50.6 for IDF1. Furthermore, results from MT and ML metrics reveal that SiamFC++ accurately maintains target consistency.

Table 2: The results of the comparative experiment

	Method	MOTA↑	MOTP↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDS↓	FM↓
One-shot method	JDE	9.8	77.2	9.2	98	244	24862	50177	606	985
	FairMOT	30.2	73.1	30.4	114	206	9831	45264	147	362
	CenterTrack	29.7	83.6	29.7	126	142	9062	41543	116	297
	UMA-MOT	27.4	69.8	30.1	96	221	9275	40186	126	408
Two-step method	DeepSort	9.5	75.2	40.5	122	265	24642	39176	616	1256
	IoU Tracker	13.2	76.3	39.6	109	262	20186	38755	593	995
	Flow Tracker	27.8	80.2	42.3	125	238	10118	39624	145	468
	Tracktor	30.4	79.8	44.8	108	214	9365	38116	102	365
The proposed	SiamFC++	35.6	85.2	50.6	142	197	9213	39275	113	293

3.2.2 Ablation Experiment

On the VisDrone2023 dataset, this paper analyzes the impact of anchor-free and anchor-based detectors on multi-object tracking in drone scenes. Most mainstream multi-object tracking algorithms currently employ anchor-based detectors, achieving satisfactory performance in numerous challenges. However, target crowding in drone scenes can degrade the performance of anchor-based detectors. Therefore, this paper employs an anchor-free detector to construct the detection module instead of the commonly used anchor-based detector. Comparative experiments validate the superiority of the anchor-free detector in drone scenarios. For ease of comparison, only the detection module was replaced while other model components remained unchanged. Experimental results are shown in Table 3. The comparison reveals that the anchor-free method consistently outperforms the anchor-based method in MOTA scores. When stride is set to 8, the anchor-free method achieves a MOTA score of 35.2, while the anchor-based method scores only 23.3. Similarly, with stride set to 16, the anchor-free detector's MOTA score is 5.3 points higher than that of the anchor-based detector. Analysis of other evaluation metrics reveals that the anchor-free detector's primary advantage lies in data association. For instance, in IF1 evaluation, the anchor-free detector significantly outperforms the anchor-based approach. Furthermore, superior performance is observed in metrics reflecting trajectory stability, such as MT, ML, and IDS.

Analysis of the experimental results reveals that this phenomenon stems from anchor-based detectors' inability to accurately align with small targets in drone scenarios. Furthermore, frequent misalignment issues intensify in crowded scenes, introducing interference to subsequent learning and processing stages. Anchor-free detectors mitigate these problems to some extent by utilizing heatmap-predicted center points for target localization. This explains why anchor-free detectors achieve superior performance in drone detection scenarios.

Furthermore, increasing the stride in anchor-free detection methods improves detection accuracy but compromises data association performance. For instance, increasing the stride from 8 to 16 raised the MOTP score to 77.2 while decreasing MOTA and IDF1 scores. This indicates that lower-resolution features are more conducive to learning robust identity information.

Table 3: Experimental Results

Stride	Detector	MOTA↑	MOTP↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDS↓	FM↓
8	anchor-based	23.3	71.4	30.8	129	266	15164	45243	162	408
8	anchor-free	35.2	76.6	51.6	145	198	9375	43532	128	327
16	anchor-based	25.1	73.5	32.4	133	254	13752	43115	141	386
16	anchor-free	30.4	77.2	47.3	137	229	9018	43462	135	334

3.3 Model Effect Analysis

The average coverage and average center position error results of the model across six distinct aerial video sequences are shown in Tables 4 and 5, respectively. The proposed model achieved optimal tracking performance in the aerial video sequences group2, person1, and UAV2, exhibiting the highest average coverage and lowest center position error across these three sequences. Notably, in person1, the average coverage reached 82.1%, while the center position error was only 11.4 pixels. The model also performed well in the wakeboard sequence. Although its average coverage was not the highest, it was only slightly lower than the best-performing model, CLRST, and it achieved the smallest center position error among all algorithms.

In the UAV2 sequence, while our model showed better performance than others in both

evaluation metrics, the combination of factors—small object scale, high flight speed, and numerous background speckles—led to target loss and drift, resulting in very poor tracking performance. In the bird video sequence, our model exhibits smaller center position errors than most models. However, when the target object's scale changes or becomes occluded, the bounding box fails to fully cover the target. Consequently, while our model accurately tracks the target, coverage significantly decreases in the middle portion of the video sequence.

In the car2_s video sequence, our model exhibited some drift and target loss due to the target's rapid motion and environmental interference. In the group2 video sequence, our model achieved the smallest center position error, demonstrating its ability to accurately track even when the target shares similar colors with the surroundings or is occluded.

Table 4: Average Coverage Rate(%)

	DSST	CLRST	Staple	MDNET	DAT	The proposed
bird	80.2	46.8	72.1	49.6	9.7	70.1
car2_s	38.8	51.2	43.4	70.2	1.3	71.2
group2	36.2	38.9	20.2	66.5	20.5	80.4
person1	60.4	46.7	55.1	60.2	58.7	82.1
UAV2	23.5	23.4	19.6	9.7	13.4	33.6
wakeboard	58.9	73.2	39.7	66.4	39.6	72.4

Table 5: Average Center Position Error(pixel)

	DSST	CLRST	Staple	MDNET	DAT	The proposed
bird	7.3	23.5	8.7	15.2	98.8	6.9
car2_s	13.2	9.6	24.1	7.7	70.2	13.2
group2	38.1	33.2	90.4	45.1	80.6	30.1
person1	20.7	18.9	17.7	23.6	94.5	11.4
UAV2	192.5	198.3	196.1	423.4	695.7	165.4
wakeboard	61.6	21.7	72.8	145.3	80.2	24.6

4 Conclusion

This paper investigates the application of artificial intelligence in drone target recognition and tracking. Through model design and experimental validation, it systematically addresses the challenges of target perception and continuous tracking in complex dynamic environments.

On the VisDrone2023 public test set, the object detector proposed in this paper achieves excellent performance of AP@0.5=0.395 and mAP=0.254, and the recognition accuracy results in 10 categories are above 0.6. Our method achieves near-optimal performance at approximately 310 epochs, whereas other approaches require 400 epochs or more. Compared to alternative loss functions, convergence speed improves by 22.5%. Among eight state-of-the-art tracking methods, SiamFC++ demonstrates superior performance. Across three comprehensive evaluation metrics, it achieves a MOTA score of 35.6, a MOTP score of 85.2, and an IDF1 score of 50.6. Anchor-free methods consistently outperformed anchor-based approaches. Increasing the stride from 8 to 16 boosted the MOTP score to 77.2, while MOTA and IDF1 scores decreased.

Under both overlap rate and center error evaluation metrics, the proposed model demonstrated superior overall tracking performance compared to other models. Particularly for person1, the average coverage rate reached 82.1%, with a center position error of only 11.4 pixels.

References

[1] Li, J., Ye, D. H., Chung, T., Kolsch, M., Wachs, J., & Bouman, C. (2016, October). Multi-target detection and tracking from a single camera in Unmanned Aerial Vehicles (UAVs). In 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 4992-4997). IEEE.

[2] Wang, S., Jiang, F., Zhang, B., Ma, R., & Hao, Q. (2019). Development of UAV-based target tracking and recognition systems. IEEE Transactions on Intelligent Transportation Systems, 21(8), 3409-3422.

[3] Ramachandran, A., & Sangaiah, A. K. (2021). A review on object detection in unmanned aerial vehicle surveillance. International Journal of Cognitive Computing in Engineering.

[4] Fu, C., Lu, K., Zheng, G., Ye, J., Cao, Z., Li, B., & Lu, G. (2023). Siamese object tracking for unmanned aerial vehicle: A review and comprehensive analysis. Artificial Intelligence Review, 56(Suppl 1), 1417-1477.

[5] Pertuack, S., & Latue, P. C. (2023). Geographic Artificial Intelligence and Unmanned Aerial Vehicles Application for Correlation Analysis of Settlement Density and Land Surface Temperature in Panggang Island Jakarta. Buana Jurnal Geografi, Ekologi Dan Kebencanaan, 1(1), 39-47.

[6] Bayramov, A. A., & Hashimov, E. G. (2020). Application SMART for small unmanned aircraft system of systems. In Handbook of Research on Artificial Intelligence Applications in the Aviation and Aerospace Industries (pp. 193-213). IGI Global Scientific Publishing.

[7] Sun, N., Zhao, J., Shi, Q., Liu, C., & Liu, P. (2024). Moving target tracking by unmanned aerial vehicle: A survey and taxonomy. IEEE Transactions on Industrial Informatics, 20(5), 7056-7068.

[8] Laghari, A. A., Jumani, A. K., Laghari, R. A., Li, H., Karim, S., & Khan, A. A. (2024). Unmanned aerial vehicles advances in object detection and communication security review. Cognitive Robotics, 4, 128-141.

[9] Ye, D. H., Li, J., Chen, Q., Wachs, J., & Bouman, C. (2018). Deep learning for moving object detection and tracking from a single camera in unmanned aerial vehicles (UAVs). Electronic Imaging, 30, 1-6.

[10] Choi, S. Y., & Cha, D. (2019). Unmanned aerial vehicles using machine learning for autonomous flight; state-of-the-art. Advanced Robotics, 33(6), 265-277.

[11] Matthew, U. O., Kazaure, J. S., Onyebuchi, A., Daniel, O. O., Muhammed, I. H., & Okafor, N. U. (2021, February). Artificial intelligence autonomous unmanned aerial vehicle (UAV) system for remote sensing in security surveillance. In 2020 IEEE 2nd International Conference on Cyberspac (CYBER NIGERIA) (pp. 1-10). IEEE.

[12] Fan, B., & Zhang, R. (2023). Unmanned aircraft system and artificial intelligence. Geomatics and Information Science of Wuhan University, 42(11), 1523-1529.

[13] Tymochko, O., Trystan, A., Matiushchenko, O., Shpak, N., & Dvulit, Z. (2022). Method of controlling a group of unmanned aircraft for searching and destruction of objects using artificial intelligence elements. Mathematical modeling and computing, (9, Num. 3), 694-710.

[14] Keneni, B. M., Kaur, D., Al Bataineh, A., Devabhaktuni, V. K., Javaid, A. Y., Zaientz, J. D., & Marinier, R. P. (2019). Evolving rule-based explainable artificial intelligence for unmanned aerial vehicles. IEEE Access, 7, 17001-17016.

[15] Puente-Castro, A., Rivero, D., Pazos, A., & Fernandez-Blanco, E. (2022). A review of artificial intelligence applied to path planning in UAV swarms. Neural Computing and Applications, 34(1), 153-170.

[16] Yin, R., Li, W., Wang, Z. Q., & Xu, X. X. (2020, November). The application of artificial intelligence technology in UAV. In 2020 5th international conference on information science, computer technology and transportation (ISCTT) (pp. 238-241). IEEE.

[17] McGee, J., Mathew, S. J., & Gonzalez, F. (2020, September). Unmanned aerial vehicle and artificial intelligence for thermal target detection in search and rescue applications. In 2020 International Conference on Unmanned Aircraft Systems (ICUAS) (pp. 883-891). IEEE.

[18] Imanian, A., Zhang, S., Fan, C., Ayhan, B., WhiteSell, A., Sayed, S., & Wanke, C. (2024). Safe and scalable collision avoidance model for small unmanned aircraft systems: An artificial intelligence approach. In AIAA SCITECH 2024 Forum (p. 1081).

[19] Zagorski, N. (2021). Analysis of the military application of unmanned aircraft and main direction for their development. Aerospace Research in Bulgaria, 33, 237-250.

[20] Whelan, J., Almehmadi, A., & El-Khatib, K. (2022). Artificial intelligence for intrusion detection systems in unmanned aerial vehicles. Computers and Electrical Engineering, 99, 107784.

[21] Kurtpınar, E. Ö. (2024). Privacy's Sky-High Battle: The Use of Unmanned Aircraft Systems for Law Enforcement in the European Union. Journal of Intelligent & Robotic Systems, 110(3), 99.

[22] Shmelova, T., Sterenharz, A., & Dolgikh, S. (2020). Artificial intelligence in aviation industries: methodologies, education, applications, and opportunities. In Handbook of research on artificial intelligence applications in the aviation and aerospace industries (pp. 1-35). IGI Global Scientific Publishing.

[23] Lahmeri, M. A., Kishk, M. A., & Alouini, M. S. (2021). Artificial intelligence for UAV-enabled wireless networks: A survey. IEEE Open Journal of the Communications Society, 2, 1015-1040.

[24] Aibin, M., Aldiab, M., Bhavsar, R., Lodhra, J., Reyes, M., Rezaeian, F., ... & Taer, M. (2021). Survey of RPAS autonomous control systems using artificial intelligence. IEEE Access, 9, 167580-167591.

[25] Jia, J., & Duan, H. (2017). Automatic target recognition system for unmanned aerial

vehicle via backpropagation artificial neural network. Aircraft Engineering and Aerospace Technology, 89(1), 145-154.

[26] Gonzalez, L. F., Montes, G. A., Puig, E., Johnson, S., Mengersen, K., & Gaston, K. J. (2016). Unmanned aerial vehicles (UAVs) and artificial intelligence revolutionizing wildlife monitoring and conservation. Sensors, 16(1), 97.

[27] Cosar, M. (2023). Artificial intelligence technologies and applications used in unmanned aerial vehicle systems. The Eurasia Proceedings of Science Technology Engineering and Mathematics, 26, 1-12.

[28] Soni, S., Chakraborty, M., & Raj, A. B. (2022, November). AI based small unmanned aerial vehicle (SUAV) targets detection and tracking techniques. In 2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS) (pp. 43-49). IEEE.

[29] Alamin, M. K. N. (2021). Detection and tracking of uav targets using deep learning. Journal of Karary University for Engineering and Science.

[30] Sai, S., Garg, A., Jhawar, K., Chamola, V., & Sikdar, B. (2023). A comprehensive survey on artificial intelligence for unmanned aerial vehicles. IEEE Open Journal of Vehicular Technology, 4, 713-738.

[31] López-Muñoz, P., San Frutos, L. G., Abarca, C., Alegre, F. J., Calle, J. L., & Monserrat, J. F. (2024, December). Hybrid Artificial-Intelligence-Based System for Unmanned Aerial Vehicle Detection, Localization, and Tracking Using Software-Defined Radio and Computer Vision Techniques. In Telecom (Vol. 5, No. 4, pp. 1286-1308). MDPI.