Improvement of Remote Sensing Target Tracking Method Based on Deep Learnin

Xuhao Wang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: sususama2@163.com

Long Ma
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: malong@xatu.edu.cn

***Abstract*—Remote Sensing Object Tracking refers to the process of detecting, recognizing, and tracking targets on the ground or at sea using remote sensing technology, particularly sensors mounted on satellite or aerial platforms to obtain high-resolution remote sensing image sequences. Current methods for remote sensing object tracking face challenges such as low tracking success rates and inefficiencies. This paper proposes a neural network for remote sensing object tracking based on SiamRPN++, which introduces an improved network structure incorporating the C3Minus module and a coordinate attention mechanism within the backbone extraction network. Furthermore, we design a feature extraction module, ResSwinT, that combines ResNet and Swin Transformer architectures to integrate local and global information obtained from feature maps as foundational features. This approach effectively addresses the aforementioned issues, and quantitative experiments demonstrate an increase in accuracy and success rates by 1.9% and 4.7%, respectively, indicating that our method effectively handles object tracking in remote sensing images.**

Keywords-Deep Learning; Object Tracking; Remote Sense

I. INTRODUCTION

In this paper, remote sensing targets refer to various objects captured in remote sensing images by satellite sensors in the visible light spectrum, specifically within the range of 0.38 to 0.76 micrometers. These images contain a wealth of detailed information, effectively reflecting the shape, color, and texture of terrestrial objects, making them easily observable by the human eye. Target recognition represents a significant area of research within the field of computer vision, with the objective of detecting and tracking objects of interest within video sequences.

The core task of target tracking involves predicting a target's future position and dimensions, starting from its initial state in the first video frame. This task comprises several crucial components: motion models, feature extraction, observation models, model updating, and ensemble methods. The motion model generates samples, feature extraction represents the target, the observation model evaluates the samples, model updating adjusts for target changes, and ensemble methods combine decisions for improved predictions.

The field of remote sensing would be incomplete without the technology of remote sensing object tracking, which plays a pivotal role in the discipline. It involves detecting and identifying specific targets or features from remote sensing images, where the targets typically refer to roads, buildings, vehicles, aircraft, and ships. With advancements in remote sensing technology, remote sensing images now possess broad perspectives and ultra-high spatial and temporal resolutions, and they are minimally affected by variations in viewing angles and lighting conditions. The development of high resolution remote sensing portraits had increasingly emphasized the importance of remote sensing object recognition in various fields such as urban planning, environmental monitoring, agriculture, military applications, and disaster management. However, this advancement has also introduced new challenges within the remote sensing domain. As illustrated in Figure 1, remote sensing object difficulty recognition significantly increases in

Figure 1. Shows remote sensing portraits. The left portrait is the original image, which contains a lot of information. The right image is a partial image captured from the left image. It can be seen that even if it is magnified many times, the information in the image is still very rich

high-resolution images due to complex backgrounds and the small size of targets relative to the overall image.

To address the aforementioned issues, this study employs deep learning techniques to improve the SiamRPN++ [1] algorithm, enhancing its resolution and feature extraction capabilities in complex backgrounds, thereby significantly improving the tracking performance of small targets in remote sensing portraits. In modified neural network architecture, we introduce the C3Minus module structure and coordinate attention mechanism [2] to optimize the feature extraction capabilities of the backbone network. Furthermore, this study design a composite feature extraction module, ResSwinT, which combines CNN [3] and Swin Transformer [4]. This module effectively integrates local and global feature information, providing a richer representation foundation for the underlying features.

II. Related Works

Remote sensing target tracking can be classified into classical tracking methods, correlation filter-based tracking methods, and deep learning-based tracking methods.

Classical tracking methods mainly include Kalman filtering, particle filtering, and Bayesian estimation. Kalman filtering provides optimal estimates only within Gaussian linear models. To address this limitation, Kulikova [5] proposed a NIRK-based precise continuous-discrete extended Kalman filter, while Zhou Huan et al. [6] introduced an adaptive unscented Kalman filter for target tracking in nonlinear systems involving model mismatches, which can handle divergence caused by sensor failures or model mismatches during the tracking process. Particle filtering is a Monte Carlo-based method that models the target state as a set of particles, where each particle represents a possible estimate of the target state. The particles propagate over time according to a process model and update their weights based on measurement information. Particle filtering has been applied to target tracking in remote sensing images by modeling the target state as a set of variables and updating the particle weights according to remote sensing data. Bayesian estimation is a probabilistic method for target tracking that models the target state as a probability distribution. It updates the distribution based on measurement information and former knowledge of the target condition. Bayesian calculation represents the target state as a set of variables and updates the distribution according to remote sensing data. The Smooth Variable Structure Filter (SVSF) [7] was the model-based estimation method

suitable for smoothing nonlinear dynamic systems. It accounts for sources of uncertainty and ensures stability the maximum limits on uncertainty and noise levels, with performance improvements achievable through finer definitions of parameter variations or uncertainty bounds.

Methods based on correlation filtering begin with initializing a target template, which is defined according to the initial target position. This template typically encompasses the target and surrounding pixels within a region of interest (ROI), and it is updated throughout subsequent video frames. S. Xuan et al. [8] developed a correlation filter based on embedded motion estimation to address the tracking of rapidly moving targets in satellite videos. Y. Liu [9] proposed a novel method employing an embedded multi-feature fusion and motion trajectory compensation kernelized correlation filter for tracking fast-moving targets in satellite videos. This approach utilizes a multi-feature fusion strategy to comprehensively describe the target, thus addressing issues of tracking accuracy caused by inaccurate target localization. Additionally, it incorporates adaptive Kalman filtering to compensate for the kernel correction in the KCF tracker, reducing boundary box drift of the objects.

In recent years, with the development of neural networks, an increasing number of remote sensing image target tracking methods have employed convolutional neural networks (CNNs), recurrent neural networks (RNNs), and attention mechanisms. Some studies utilize pre-trained CNNs, such as VGG [10] and ResNet [11], to extract features from remote sensing images, while others focus on training CNNs from scratch using large datasets. Target tracking networks based on Siamese networks have also been applied by Bertinetto et al. [12] They introduced SiamFC, which integrates Siamese networks into target tracking. However, SiamFC relies on exhaustive multi-scale search to regress the target bounding box, resulting in low efficiency and accuracy. Based on SiamFC, Li et al.[13] proposed SiamRPN, which integrates previous correlation filtering methods. It uses the first frame as a detection template while improving the output of SiamFC by enhancing the twin network's output layer. This enhancement allows for the extraction of proposed regions and scoring, resulting in more accurate target bounding boxes. Zhu et al.[14] introduced DaSiamRPN, which focuses more on semantic interference factors and incorporates an interference-aware module for incremental learning. Building on SiamRPN, Li et al.[1] developed SiamRPN++, which employs multi-layer aggregation to fuse shallow and deep features during feature extraction. This approach leverages modern deep neural networks' capability to capture features, significantly enhancing the model's accuracy. Wang et al. [15] presented SiamMask, which combines segmentation concepts with the twin network, adding a mask branch during the regression phase to compute the loss of the segmentation network. Xu et al. [16] proposed SiamFC++, which employs an anchor-free design, allowing the network to directly classify and regress candidate boxes for each position of the corresponding features, thus avoiding the impact of predefined anchors on network speed. In the field of remote sensing object tracking, Yan et al. [17] proposed a novel search algorithm, LightTrack, which encodes all possible frames into a BackBone supernet and a head supernet, significantly reducing inference time and thus speeding up the overall tracking process. Cao et al. [18] introduced online temporal adaptive convolution and an adaptive temporal transformer. The former dynamically adjusts convolutional weights using temporal information from previous frames to enhance spatial features, while the latter employs efficient temporal encoding to adjust similarity maps accurately, thereby improving the network's temporal awareness. Zhou et al. [19] integrated computer vision with natural language processing by unifying visual localization and object tracking into a single task. This framework leverages a multi-source relational module in the Transformer to effectively build a multimodal network structure. Hong et al. [20] developed a unified visual object tracking framework, OneTracker, which can handle multiple tracking tasks, including conventional RGB and RGB+X tracking, where X represents additional information, such as natural language descriptions, depth, thermal imaging, or event maps. OneTracker abstracts the common characteristics of tracking tasks and extends based on them to adapt to various tracking scenarios.

III. METHODS

This study proposes an improved overall network structure based on SiamRPN++, aiming to enhance tracking accuracy in the context of remote sensing images. The architecture of the entire network is illustrated in the figure below.

The network architecture employed in this research builds upon the foundational framework of SiamRPN++, incorporating significant modifications to several key modules. The architecture is systematically divided into three distinct components. The first component is the feature extraction module. In this study, we have modified the ResNet backbone, enhancing its enhancing feature extraction by incorporating the C3Minus and CA modules. These enhancements are designed to bolster the network's ability to extract relevant features from remote sensing images effectively. The second component consists of the ResSwinT module, which consolidates image patches from deeper layers to construct hierarchical feature maps. This approach significantly augments the network's modeling capacity, enabling it to capture complex representations within the input data. Finally, the third component is the regression head, which facilitates the precise localization of the target within the search space. This component is critical for ensuring that the network can accurately identify and track the target across varying conditions in remote sensing applications.

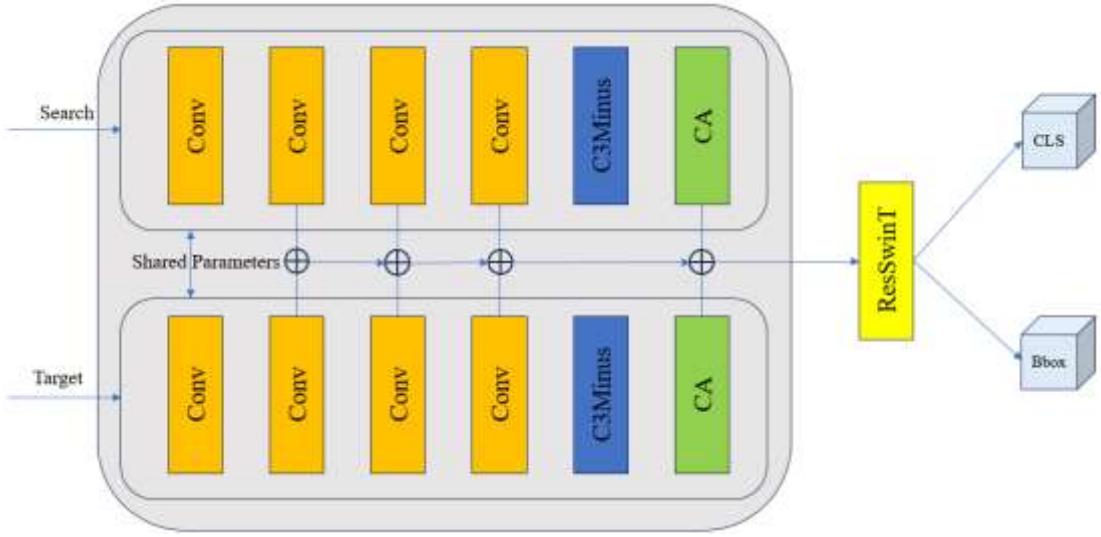

Figure 2. Network Structure. The proposed network's architecture is organized into three main components: the Backbone, the ResSwinT module, and the regression head. The Backbone is constructed upon the ResNet-50 architecture, with significant enhancements that include the incorporation of C3Minus and CA modules. These additions are designed to enhance the ability to extract features. The network effectively fuses the outputs from three distinct convolutional layers along with the outputs from the CA module. This fused information is subsequently passed to the ResSwinT module, which processes the data to generate hierarchical feature representations. Finally, the output from the ResSwinT module is directed to the regression head, which is responsible for accurately locating the target object in the image.

A. C3Minus

C3Minus module represents a significant advancement over the traditional CSPBottleneck block. It incorporates several key enhancements that contribute to improved performance and efficiency in convolutional neural networks. The specific improvements offered by C3Minus are as follows:

- **Efficiency**: C3Minus optimizes the convolutional operations by combining multiple individual convolutions into a single convolution layer. This strategic integration reduces the overall number of operations required during the forward pass, leading to reduced memory consumption and enhanced convolution efficiency. Consequently, this allows the network to process inputs more swiftly while maintaining performance.

- **Simplified Architecture**: By streamlining the structure of the network, C3Minus eliminates redundant convolutions that do not contribute significantly to feature extraction. This simplification results in a more compact architecture, which not only facilitates faster training times but also eases the deployment of the model in resource-constrained environments.
- **Reduced Computational Cost**: The C3Minus module effectively integrates convolutions, which leads to a decrease in both the number of parameters and the overall computational costs associated with the model. This reduction in complexity not only shortens the training duration but also enhances the model's scalability, allowing it to be more adaptable to various tasks and datasets.

The C3Minus module is of paramount importance in enhancing the performance of network by streamlining architecture, improving operational efficiency, and minimizing computational demands.

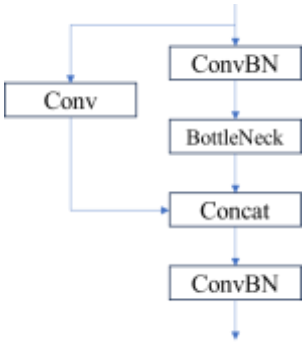

Figure 3. C3Minus network structure. The network consists of three convolution layers and one BottleNeck layer. ConvBN refers to the Batch Normalization and activation function, and Concat is a short circuit

B. CA

The CA module, which stands for Coordinate Attention, introduces a novel attention mechanism specifically designed for lightweight neural networks. Proposed by Hou et al.[2], this mechanism effectively integrates positional information into the channel attention framework, improving the model's capacity to detect critical spatial correlations and extended dependencies in the input data.

The CA module operates through two fundamental steps:

- **Coordinate Information Fusion**: In this initial phase, the module incorporates spatial coordinate information, which serves as a crucial component in understanding the spatial arrangement of features. By fusing this positional information with the channel-wise features, the CA module creates a more comprehensive representation that reflects the significance of different spatial locations relative to the features being processed. This fusion allows the network to prioritize certain features based on their spatial context, thereby improving its ability to discern important patterns.
- **Coordinate Attention Generation**: Following the fusion of coordinate information, the module generates coordinate attention maps. These maps are designed to emphasize relevant features across various spatial coordinates, effectively guiding the network to focus on critical areas within the input image. The generation of coordinate attention maps enables the network to adaptively weigh features based on their spatial context, enhancing the overall representation capabilities of the model.

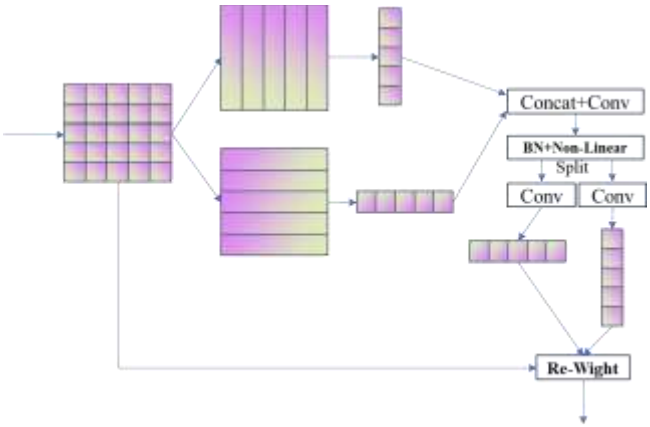

Figure 4. CA network structure. The entire module performs average pooling in the horizontal and vertical directions, then uses Transform to encode the spatial information, and finally fuses the spatial information by weighting it on the channel, making the network's overall perception of space more profound.

C. ResSwinT

In this study, the ResSwinT module is developed as an advanced extension of the ResNet block, integrating Swin Transformer layers to leverage the strengths of both convolutional neural networks (CNNs) and transformers. Inclusion of Swin Transformer layers allows for the construction of hierarchical feature representations, which are essential for capturing complex patterns and relationships within the data.

The ResSwinT module operates by utilizing the local and global attention mechanisms inherent in the Swin Transformer architecture. This enables the network to effectively learn contextual information across various scales, improving its capacity for discern intricate details in the input images. Specifically, hierarchical representations derived from the transformer layers facilitate the construction of detailed feature maps, which are crucial for accurately identifying and tracking targets.

One of the key advantages of the ResSwinT module is its capability to enhance local detail capture at lower stages of the network. By focusing on fine-grained features at these early stages, the module ensures that essential information is preserved and emphasized throughout the subsequent processing layers. This focus on local details is particularly beneficial in remote sensing applications, where variations in target appearance and background complexity can pose significant challenges.

Moreover, the integration of the Swin Transformer layers introduces a flexible windowing mechanism that enables the model to adaptively adjust its concentrate on different regions from the input image. This adaptability not only improves the model's efficiency but also enhances its performance in diverse scenarios, making it robust against various conditions encountered in real-world tracking tasks.

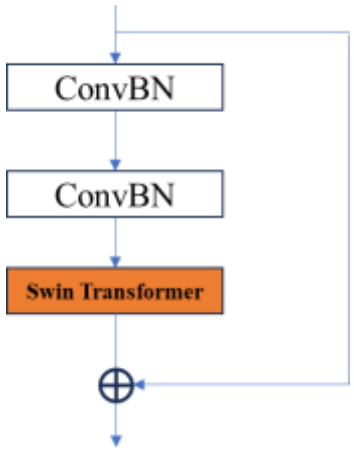

Figure 5. ResSwinT network structure, the overall structure uses a RestNet module as the basis, adds a Swin Transformer layer, and further extracts and fuses image features.

D. Detection Head

In this study, the detection head processes the outputs from the ResSwinT module to produce two distinct outputs: a binary classification output and a regression output. The main goal of binary classification is to distinguish the target object from the background within the search area. Meanwhile, the regression output is planned determine the accurate location of the target.

Regression head employs deep cross-correlation convolution to assess the relationship between the search area and the target template. This operation begins with feature maps secured since both the template and search branches, which were processed in batches to ensure they have the same number of channels. Subsequently, these two feature maps undergo a channel-wise correlation operation (essentially a convolution operation) to produce a result that maintains same number of channels. Finally, resulting feature maps are normalized to effectively merge the outputs from different channels. To finalize the process, an extra convolutional layer is included to produce the ultimate classification and regression outcomes.

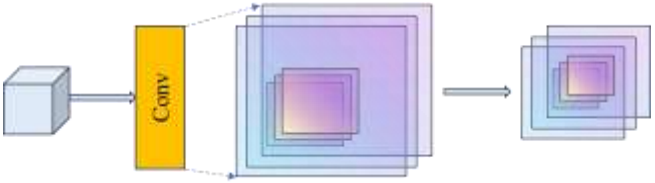

Figure 6. Depth-wise Cross Correlation

E. Loss Function and Optimizer

Since the network employed in this study outputs both classification and regression results, a mixed loss function[21] is used to measure the discrepancy between each branch output and the ground truth. The overall loss function is defined as:

$$Loss = \alpha L_{cls} + \beta L_{reg} \quad (1)$$

Where α and β are the weight coefficients for balancing the classification and regression loss components, set to 0.3 and 0.7 in this study, respectively. L_{cls} and L_{reg} represent the classification loss and regression loss functions. As the Backbone used in this study is based on ResNet-50, a model with strong classification capabilities, we assign a higher weight (0.7) to the classification component.

The classification output is responsible for distinguishing pixels within the search region as either background or target of interest, essentially a binary classification task. Therefore, we adopt the binary cross-entropy loss function:

$$L_{cls} = -y \times log(\hat{y}) - (1-y) \times log(1-\hat{y}) \quad (2)$$

where y denotes the ground truth, and $\hat{y}$ represents the predicted binary output from the network.

For the regression task, the output represents four absolute coordinates within the search region (upper left, upper right, lower left, and lower right), relative to the lower left corner of the region. Given that this task is essentially a regression problem, we apply the L_1 loss function[22], which is commonly used in regression tasks for its interpretability and tendency to produce sparse solutions:

$$L_1(x, \hat{x}) = \frac{1}{m} \sum_{i=1}^{m} \left| x_{(i)} - \hat{x}_{(i)} \right| \quad (3)$$

where x represents the ground truth, $\hat{x}$ is the network's regression output, and mmm is the number of coordinates in each sample.

The optimizer used in this study is Stochastic Gradient Descent (SGD). The SGD optimizer requires parameters such as the list of trainable parameters, momentum, and weight decay. It enhances training efficiency by accelerating the updates along relevant directions and reducing oscillations in irrelevant directions. The update process is given by:

$$V(t) = \gamma V(t-1) + \varepsilon \nabla Loss(\theta) \quad (4)$$

where t is the iteration count, ε is the learning rate, and $\nabla Loss(\theta)$ is the gradient of the loss with respect to the model parameters θ at the current iteration. Finally, parameters are updated as:

$$\theta = \theta - V(t) \quad (5)$$

Here, γ represents the momentum term, typically set to 0.9.

IV. EXPERIMENTS

A. Experimental Environment & Dataset

TABLE I. EXPERIMENTAL ENVIRONMENT

Experimental Environment	Version
CPU	Intel Xeon E5-2698
GPU	NVIDIA Tesla V100 32G
Language	Python 3.8
Framework	Pytorch

The hardware configuration for this study consists of an Intel Xeon E5-2698 CPU, paired with four NVIDIA Tesla V100 GPUs. The system environment is Ubuntu 18.04, and the model is built using the PyTorch framework with Python version 3.8.

The dataset used in this study is based on DIOR and UCAS-AOD, which have been fused and processed together for joint training to enhance the model's robustness.

The performance of the model is quantified using accuracy and success rate. Accuracy is defined as the proportion of frames in which the target's center position error (Δ) falls below a specified threshold, compared to the total number of frames. In previous studies, this threshold is

typically set at 20. The method for determining accuracy can be expressed as follows:

$$Precision = \frac{Count(\Delta < 20)}{Count(\Delta_{(all)})} \quad (6)$$

Where Δ is:

$$\Delta = \sqrt{\left(x_p - x_r\right)^2 + \left(x_p - y_r\right)^2} \quad (7)$$

The success rate refers to the ratio of the number of frames in which the overlap between the successfully tracked target detection box and the ground truth box exceeds a predefined threshold to the total number of frames. It measures the intersection-over-union (IoU) of the computed box with the ground truth box, and the formula is as follows:

$$IoU = \frac{Area\ of\ \ Interserction\ of\ \ two\ boxes}{Area\ of\ Union\ of\ \ two\ boxes} \quad (8)$$

$$Sucess = \frac{Count(IoU > 0.5)}{Count(IoU_{all})} \quad (9)$$

B. Experimental Results

In this study, we utilized a custom-built test set specifically designed for the evaluation of our proposed tracking method. The results of our experiments are illustrated in Figure 7, which showcases representative examples of the tracking performance achieved by our approach.

Figure 7. Experimental results. The red part is the model result and the green part is the real frame.

As depicted in the figure, the proposed method exhibits remarkable efficacy in tracking small targets within remote sensing images. Notably, it maintains a high level of accuracy in both tracking and recognition, even under challenging conditions such as occlusion. This level of robustness is essential in real-world scenarios where targets might be partially hidden by environmental elements or other objects.

To further validate the effectiveness of our approach, we conducted a comparative analysis against five state-of-the-art tracking models from recent years: SiamRPN [13], SiamRPN++ [1], SiamMask [15], SiamBAN [23], and SiamCar [24]. These models were selected based on their prominence in the field of object tracking, ensuring a comprehensive evaluation of our method's performance. The results are summarized in the table below:

TABLE II. THE SUCCESS RATE AND ACCURACY OF THIS METHOD ARE COMPARED WITH THE SOTA METHOD. THE RED VALUE IN THE TABLE IS HIGHEST, AND GREEN VALUE IS SECOND HIGHEST.

Models	Years	*Precision*	*Success*
SiamRPN	2018	0.753	0.342
SiamRPN++	2018	0.435	0.261
SiamMask	2019	0.569	0.278
SiamBAN	2020	0.784	0.497
SiamCar	2022	0.769	0.502
Ours	-	0.803	0.549

The experimental results clearly indicate that the enhancements introduced in this study have led to a significant improvement in the model's performance. Specifically, our proposed method achieves an impressive accuracy increase of 1.9%, reflecting a more precise capability in target tracking within remote sensing images. Additionally, we observe a notable improvement in the success rate, which has risen by 4.7%.

This increase in success rate signifies a substantial advancement in the model's ability to consistently and reliably track targets, even under challenging conditions such as occlusion and varying backgrounds. The enhancements not only demonstrate the effectiveness of the modifications made to the network architecture but also underscore the potential of our approach in real-world applications where high accuracy and robustness are paramount.

V. Conclusion

This study investigates target tracking methods for remote sensing imagery and presents several innovations by the authors. Based on the SiamRPN++ framework, a series of enhancements were introduced. Firstly, the C3Minus and CA modules were incorporated into the backbone network, significantly improving feature fusion and extraction capabilities. These additions allow the network to capture richer feature information when processing remote sensing images, resulting in enhanced tracking accuracy, especially in challenging scenarios with complex backgrounds and changing target appearances. Additionally, this study introduces the novel RestSwinT module, which combines the strengths of the Swin Transformer and ResNet to bolster the network's spatiotemporal feature extraction capabilities. In target tracking tasks, the effective integration of spatiotemporal information enables the network to capture dynamic target changes more accurately. By incorporating the RestSwinT module, the network achieves more effective spatiotemporal feature fusion, further enhancing target identification and tracking performance.

With ongoing advancements in deep learning and computer vision, future research in this domain could explore methods for the effective integration of multimodal data (such as infrared and synthetic aperture radar imagery) to enhance tracking accuracy and reliability. Optimizing network architectures and algorithms to enable more efficient real-time tracking performance is also a promising direction.

References

[1] Li, Bo, et al. "Siamrpn++: Evolution of siamese visual tracking with very deep networks." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.

[2] Hou, Qibin, Daquan Zhou, and Jiashi Feng. "Coordinate attention for efficient mobile network design." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

[3] Rakhlin, A. "Convolutional neural networks for sentence classification." GitHub 6 (2016): 25.

[4] Liu, Ze, et al. "Swin transformer: Hierarchical vision transformer using shifted windows." Proceedings of the IEEE/CVF international conference on computer vision. 2021.

[5] Kulikova, Maria V., and G. Yu Kulikov. "NIRK-based accurate continuous–discrete extended Kalman filters for estimating continuous-time stochastic target tracking models." Journal of Computational and Applied Mathematics 316 (2017): 260-270.

[6] Zhou, Huan, et al. "Adaptive unscented Kalman filter for target tracking in the presence of nonlinear systems involving model mismatches." Remote Sensing 9.7 (2017): 657.

[7] Habibi, Saeid. "The smooth variable structure filter." Proceedings of the IEEE 95.5 (2007): 1026-1059.

[8] Xuan, Shiyu, et al. "Object tracking in satellite videos by improved correlation filters with motion estimations." IEEE Transactions on Geoscience and Remote Sensing 58.2 (2019): 1074-1086.

[9] Liu, Yaosheng, et al. "Object tracking in satellite videos based on correlation filter with multi-feature fusion and motion trajectory compensation." Remote Sensing 14.3 (2022): 777.

[10] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[11] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[12] Bertinetto, Luca, et al. "Fully-convolutional siamese networks for object tracking." Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14. Springer International Publishing, 2016.

[13] Li, Bo, et al. "High performance visual tracking with siamese region proposal network." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[14] Zhu, Zheng, et al. "Distractor-aware siamese networks for visual object tracking." Proceedings of the European conference on computer vision (ECCV). 2018.

[15] Wang, Qiang, et al. "Fast online object tracking and segmentation: A unifying approach." Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. 2019.

[16] Xu, Yinda, et al. "SiamFC++: Towards robust and accurate visual tracking with target estimation

guidelines." Proceedings of the AAAI conference on artificial intelligence. Vol. 34. No. 07. 2020.

[17] Yan, Bin, et al. "Lighttrack: Finding lightweight neural networks for object tracking via one-shot architecture search." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[18] Cao Z, Huang Z, Pan L, et al. TCTrack: Temporal contexts for aerial tracking[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 14798-14808.

[19] Zhou, Li, et al. "Joint visual grounding and tracking with natural language specification." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023.

[20] Hong, Lingyi, et al. "Onetracker: Unifying visual object tracking with foundation models and efficient tuning." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.

[21] Mao, Anqi, Mehryar Mohri, and Yutao Zhong. "Cross-entropy loss functions: Theoretical analysis and applications." International conference on Machine learning. PMLR, 2023.

[22] Barron, Jonathan T. "A general and adaptive robust loss function." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.

[23] Chen, Zedu, et al. "Siamese box adaptive network for visual tracking." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[24] Guo, Dongyan, et al. "SiamCAR: Siamese fully convolutional classification and regression for visual tracking." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.