applied sciences

MDPI

Article

SSTrack: An Object Tracking Algorithm Based on Spatial Scale Attention

Qi Mu *, Zuohui He, Xueqian Wang and Zhanli Li

College of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an 710054, China; 21208049004@stu.xust.edu.cn (Z.H.); 22208223083@stu.xust.edu.cn (X.W.); lizl@xust.edu.cn (Z.L.)
* Correspondence: muqi@xust.edu.cn

Abstract: The traditional Siamese object tracking algorithm uses a convolutional neural network as the backbone and has achieved good results in improving tracking precision. However, due to the lack of global information and the use of spatial and scale information, the accuracy and speed of such tracking algorithms still need to be improved in complex environments such as rapid motion and illumination variation. In response to the above problems, we propose SSTrack, an object tracking algorithm based on spatial scale attention. We use dilated convolution branch and covariance pooling to build a spatial scale attention module, which can extract the spatial and scale information of the target object. By embedding the spatial scale attention module into Swin Transformer as the backbone, the ability to extract local detailed information has been enhanced, and the success rate and precision of tracking have been improved. At the same time, to reduce the computational complexity of self-attention, Exemplar Transformer is applied to the encoder structure. SSTrack achieved 71.5% average overlap (AO), 86.7% normalized precision (NP), and 68.4% area under curve (AUC) scores on the GOT-10k, TrackingNet, and LaSOT. The tracking speed reached 28fps, which can meet the need for real-time object tracking.

Keywords: object tracking; Siamese network; spatial scale attention; swin transformer; potisional encoding

Citation: Mu, Q.; He, Z.; Wang, X.; Li, Z. SSTrack: An Object Tracking Algorithm Based on Spatial Scale Attention. *Appl. Sci.* **2024**, *14*, 2476. https://doi.org/10.3390/app14062476

Academic Editor: Ugo Vaccaro

Received: 8 February 2024
Revised: 8 March 2024
Accepted: 11 March 2024
Published: 15 March 2024

1. Introduction

Object tracking is one of the research hotspots in the field of computer vision, and its primary task is to specify a target object in the initial frame of a video and continuously track this object with a rectangular box in subsequent video frames to achieve target localization and scale estimation [1,2]. It finds extensive applications in various domains, such as public safety [3,4], autonomous driving [5,6], image processing [7], and sports competitions, etc. [8].

Despite the success achieved by existing tracking algorithms on simple datasets like OTB [9], the actual trackers are often disturbed or simultaneously affected by factors such as Fast Motion, Illumination Variation, and Scale Variation. The increasing video resolution also imposes specific requirements on the speed of the tracking algorithm, hindering their practical applications in real-world scenarios [10,11]. Therefore, proposing an algorithm capable of achieving real-time object tracking in large datasets containing various complex environments is of paramount significance [12].

Siamese tracking algorithms utilize dual-branch convolutional neural networks with shared weights to extract target object features, calculate the similarity between the target region and the search region, and determine the tracking target object's position. Scholars have attempted to enhance tracking accuracy and success rate by incorporating Transformers to fuse deep features or extract global features [13,14]. However, existing Siamese tracking algorithms still face limitations in tracking capability under complex environments, primarily due to:

1. Traditional Siamese tracking algorithms use convolutional neural networks or Transformer as the backbone, allowing the utilization of features from a global perspective, enhancing the ability to learn long-range feature representations. However, due to the lack of utilization of local detailed information, this type of tracking algorithm is not robust when faced with interference from complex environments such as Fast Motion [15,16].
2. As a large visual model, Siamese tracking algorithms' self-attention computation is complex, leading to poor real-time tracking performance [17].

Spurred by the above deficiencies, in order to implement a robust and real-time tracking algorithm, we propose a transformer tracker called SSTrack based on spatial scale attention. Figure 1 illustrates the comparison results of our tracking algorithm, SSTrack, with other mainstream object tracking algorithms. SSTrack achieved 71.5% AO on the GOT-10K and 68.4% AUC on the LaSOT, taking the lead.

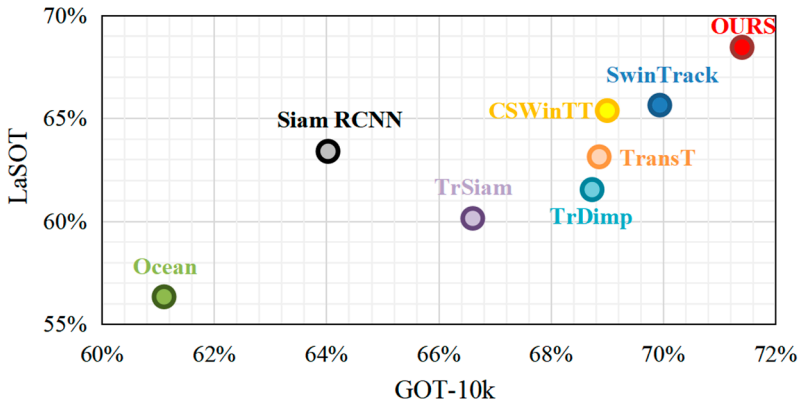

Figure 1. Comparison with other advanced object tracking algorithms on GOT-10k and LaSOT benchmark.

The main contributions of this paper are as follows:

- Addressing issue 1, we employ the Swin Transformer as the backbone and construct a spatial scale attention module. This module aims to perceive spatial and multi-scale information in the feature sequence while extracting global features.
- Addressing issue 2, we adjust the utilization of position encoding in the feature fusion stage and the computation method of self-attention. This adjustment avoids the need to retrain the Swin Transformer on ImageNet. It reduces the complexity of self-attention computation, thus improving tracking speed and meeting real-time tracking requirements.
- To validate the proposed algorithm's effectiveness, we compare SSTrack with mainstream tracking algorithms on large tracking datasets such as GOT-10k, TrackingNet, and LaSOT. The test set covers various tracking scenarios, including indoor and outdoor, and target categories, such as pedestrians and vehicles. Additionally, ablation experiments are conducted. The results demonstrate that compared to other mainstream Siamese tracking algorithms, the proposed algorithm achieves high tracking accuracy, maintains robustness in complex tracking scenarios, and satisfies the real-time tracking requirements.

The rest of the paper is structured as follows: in Section 2, we provide a brief introduction to the Siamese tracking algorithms, and Transformer tracking algorithms closely related to this paper in recent years. Next, in Section 3, we introduce the overall tracking process of SSTrack and the network structure of three sub-networks. In Section 4, extensive comparative experiments and ablation experiments are conducted on multiple datasets to evaluate SSTrack. Finally, Section 5 summarizes the conclusions and proposes plans for future research.

2. Related Works

2.1. Siamese Tracking

Siamese tracking algorithms consist of two input branches with identical network structures, and they learn the similarity between inputs through a structure with shared weights. For example, SiamFC [18] replaces correlation filters with a Siamese neural network for similarity learning, overcoming the limitation of requiring the template region to match the size of the search region. Building upon this, SiamRPN [19] introduces a region proposal network for foreground–background classification and refinement. SiamRPN++ [20] employs a deep ResNet network instead of AlexNet, increasing the depth of the algorithms. Siam-Mask [21] adds extra branches and loss functions, unifying tracking and segmentation tasks. SiamDT [22] achieves accurate feature representation of the target object by enhancing ResNet-50 and adding an additional template branch, aiming to address issues related to template failure and model drift.

In addition, some tracking algorithms further enhance accuracy by incorporating channel attention [23], adding extra template branches [24], and combining convolutional neural network (CNN) and Long Short-Term Memory (LSTM) networks [25].

However, the feature extraction networks in the above algorithms are mostly convolutional neural networks, extracting primarily local features of the target object, lacking utilization of global features. Additionally, these algorithms need more utilization of higher-order statistical quantities and fully exploit the spatial information of features, reducing the discriminability of similar objects. Therefore, tracking results are susceptible to interference from complex environmental factors such as Fast Motion, Illumination Variation, and Scale Variation.

2.2. Transformer Tracking

Transformer [26] is a model structure based on self-attention, known for capturing long-range dependencies and extracting global features. It has found extensive applications in natural language processing. With the continuous integration and development of the fields of natural language processing and computer vision [27], Vision Transformer (ViT) [28] pioneered the introduction of Transformer into computer vision, demonstrating favorable results. In 2021, TrSiam [13] combined Transformers with correlation filters, becoming the first Siamese object tracking method with a Transformer structure that enhanced the tracker's feature representation capability. TransT [14] utilized Transformers to interact with the two branch features extracted by CNNs, effectively fusing features from the target and search regions, thereby improving tracking accuracy. Swin Transformer [29] divided the input region using non-overlapping sliding windows, enabling information interaction and transmission between windows, further enhancing the versatility of the Transformer. In 2022, SwinTrack [30] employed Swin Transformer as the backbone, achieving superior tracking performance compared to CNN or hybrid CNN–Transformer methods. However, Swin Transformer lacks the utilization of scale information at a single hierarchy level and has room for improvement in the representation of local features.

2.3. Exemplar Transformer

Exemplar Transformer [17] is a Transformer structure based on a single-instance-level attention layer. Despite the success of Swin Transformer in reducing the computational complexity of attention during the feature extraction stage by dividing the window, there is still room for improvement in the efficiency of self-attention computation during the feature fusion stage. The Exemplar Transformer points out that in natural language processing, each feature represents a specific word or token, during self-attention computation, it is necessary to merge all features. However, in visual tasks, adjacent spatial locations often correspond to the same object. When tracking a single object, the attention computation process can be optimized. exemplar attention achieves this by spatial aggregation and compression, utilizing globally generated query tokens from the input image to focus attention on a single instance. It eliminates the need for a similarity function within the

sample, providing a unique perspective on spatial relationships in visual tasks. Exemplar Transformer reduces the complexity of self-attention, improves computational efficiency, and has been applied in E.T.Track, yielding significant results.

3. The Proposed Method

In response to the suboptimal tracking performance of Siamese tracking algorithms in complex scenarios such as Fast Motion, we propose an object tracking algorithm based on spatial scale attention, SSTrack. SSTrack uses Swin Transformer instead of the convolutional neural network to extract global features of the target object, and builds a spatial scale attention module to enhance the ability to express local details of the target.

The overall network structure of the SSTrack is illustrated in Figure 2, comprising three main components: the feature extraction sub-network based on Swin Transformer, the feature fusion sub-network based on exemplar attention, and the classification and regression sub-network.

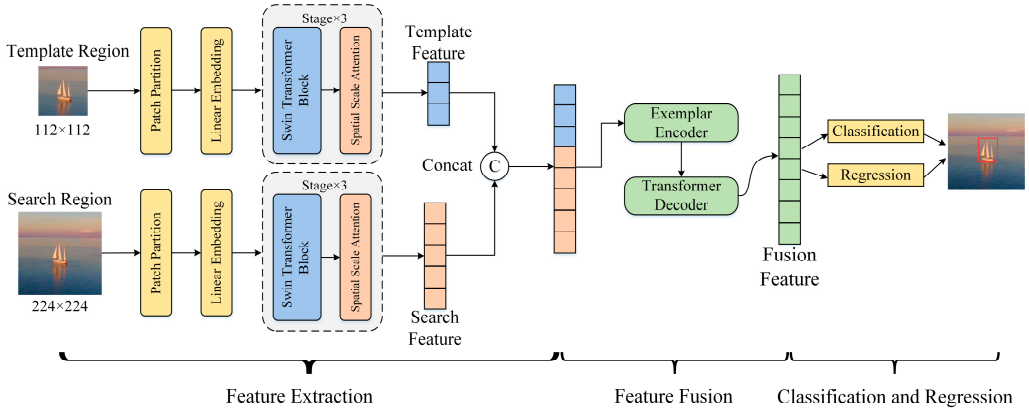

Figure 2. Architecture of SSTrack.

Firstly, when extracting global features from the input images of the template region and the search region through the Swin Transformer Block in the upper and lower branches, the constructed spatial scale attention module utilizes higher-order statistical quantities and dilated convolutions to enhance the representation capability of object local details. This module enables it to cope with complex environments. Next, the correlation between feature sequences and position information is computed separately in the feature fusion sub-network. Self-attention computation is then performed using exemplar attention, improving tracking speed and meeting the real-time tracking requirements. Finally, the designed classification and regression loss function aids in learning feature representations and position estimates, yielding the ultimate tracking results.

3.1. Feature Extraction

In the feature extraction sub-network, we embed a spatial scale attention module based on the Swin Transformer. While extracting the global features of the target object, this module helps capture detailed local features, satisfying the tracker's requirement to learn global contextual features and enhancing the perception of local features.

Patch Merging plays a role in pooling and downsampling by dividing and laying the feature map. However, it only utilizes low-level information of the image, limiting the further expression of target object features. As shown in Figure 3, the spatial scale attention module is constructed based on the original Patch Merging.

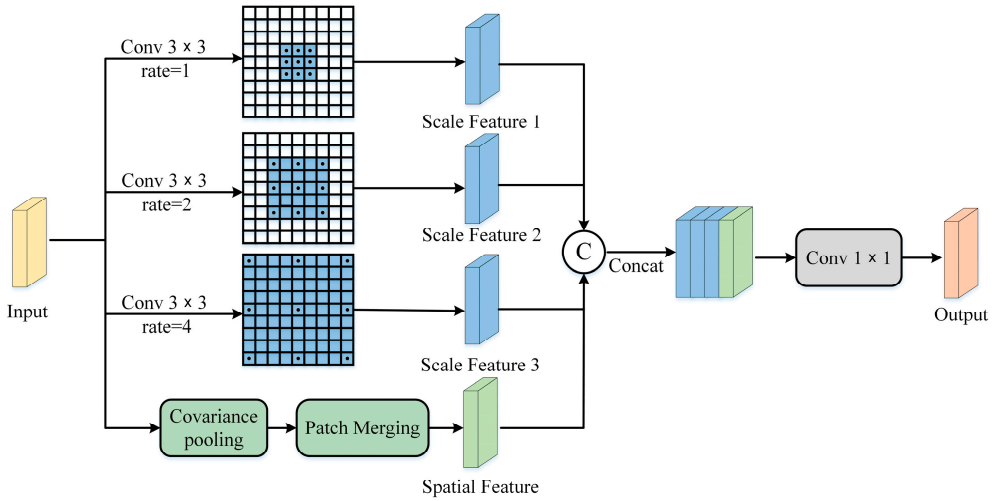

Figure 3. Architecture of the spatial scale attention module.

We introduce a layer of covariance pooling before Patch Merging. Compared to regular pooling, covariance pooling considers the covariance relationships between features, selects a value that represents the distribution of a feature map by calculating the covariance matrix of the feature map, which is the second-order information, enhancing the expressive capability of spatial information [31]. This approach better captures the correlations within input features. Dilated convolution [32] increases the receptive field by introducing dilation in the convolutional kernel, aiding the network in more effectively capturing a broader range of contextual information. Inspired by the dilated pyramid in image segmentation, [33] three dilated convolution branches with 1, 2, and 4 dilation rates are incorporated, forming a hierarchical Transformer. This design achieves the utilization of multiscale information within a single hierarchy. Finally, to fuse the dilated features with global spatial features, the outputs of each branch are concatenated and dimensionally reduced. This module outputs a feature sequence with the exact dimensions as the input, providing a feature sequence containing more spatial scale information for subsequent structures.

The input image (H × W × 3) is divided into a sequence of patch tokens through the Patch Partition, undergoes a linear embedding, and enters three consecutive stages. In each stage, the Swin Transformer Block first extracts global features from the feature sequence of the current hierarchy. Subsequently, the spatial scale attention module is applied to extract and fuse local detailed information, and the result is fed into the next stage for further processing. The output of the third stage serves as the final output of the feature extraction sub-network.

3.2. Feature Fusion

In the feature fusion sub-network, we adjust the utilization of position encoding by separating the traditional position encoding from the binding relationship with the feature sequence. This separation allows for a more comprehensive interaction of concatenated features during the feature fusion stage [34]. The exemplar attention module implements visual motion representation learning in the encoder–decoder structure. The encoder is responsible for the feature fusion of the concatenated features. At the same time, the decoder is utilized for visual motion representation learning, enhancing the understanding of the tracked target images. The overall structure is illustrated in Figure 4.

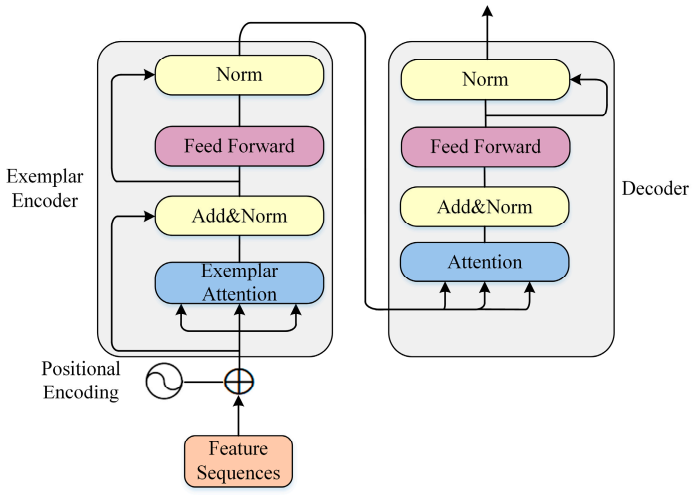

Figure 4. Architecture of feature fusion sub-network.

3.2.1. Exemplar Attention

Exemplar attention [17] is employed instead of standard attention as the self-attention calculation layer in the encoder. This is to reduce computational complexity and improve tracking speed.

Traditional feature fusion networks rely on standard attention to calculate the intrinsic relationships of features, as shown in Formula (1):

$$Attention(X) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V = softmax\left\{\frac{(xW_Q)(W_k^T x^T)}{\sqrt{d_k}}\right\}(xW_V) \quad (1)$$

In the formula, Q represents the query vector, K represents the key vector, and V represents the value vector. By calculating the dot product of the Q and K, scaling with a scaling factor $\sqrt{d_k}$, and normalizing through the softmax function, the final weighted sum of the V effectively captures crucial information at each position in the input sequence.

In natural language processing, each feature represents a specific word or token, necessitating the calculation of interrelationships among all input sequences. However, in visual tasks like object tracking, adjacent spatial locations often represent the same object, leading to a corresponding reduction in the number of feature vectors. It enables the construction of a coarser visual representation, thereby reducing computational complexity.

Exemplar attention, as shown in Figure 5, constructs the query vector Q by compressing the spatial dimensions of the input feature map X through average pooling to size S, followed by linear mapping. The key vector K relies on learning a small set of data information as exemplars, not solely dependent on relationships within the samples. Furthermore, the value vector V undergoes convolutional operations at the local level.

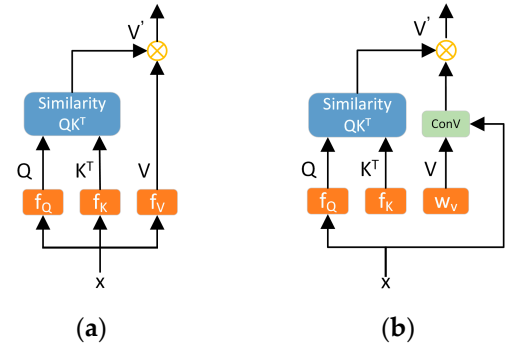

Figure 5. Standard attention (**a**) and exemplar attention (**b**).

The calculation of Q, K, and V in exemplar attention is as shown in Formula (2):

$$\begin{aligned} Q &= \Psi_S(X)W_Q \\ K &= \hat{W}_K \\ V &= W_V \otimes X \end{aligned} \tag{2}$$

$\Psi_S(X)$ represents the operation of mapping the feature map X to dimensions $S \times S$ using a two-dimensional adaptive average pooling layer and obtaining the query vector Q by multiplying it with a parameter matrix W_Q. $\hat{W}_K$ represents the key vector K obtained after learning from the dataset, and $W_V \otimes X$ represents the value vector V obtained by refining the input representation locally through convolution instead of the projection operation on the feature map X. From Formulas (1) and (2), the calculation of exemplar attention can be deduced, as shown in Formula (3).

$$Attention(X) = softmax\left\{\frac{(\Psi_S(X)W_Q)(\hat{W}_K^T)}{\sqrt{d_k}}\right\}(W_V \otimes X) \tag{3}$$

3.2.2. Positional Encoding

We separate the binding relationship between input and positional encoding, and introduces relative positional bias, ensuring model flexibility while avoiding excessive computations.

In the encoder, positional encoding represents the positional relationships between input sequences. Traditional encoders treat position information and feature sequences as a whole, performing self-attention calculations. However, this approach introduces redundant information and limiting tracking performance. This paper adopts the TUPE [33] structure, conducting correlation calculations separately for the feature sequence and positional information using parameter matrices. This reduces redundancy caused by binding. The calculation of the TUPE is illustrated in Formula (4):

$$\alpha_{ij} = \frac{1}{\sqrt{2d}}(x_i^l W^{Q,l})(x_j^l W^{K,l})^T + \frac{1}{\sqrt{2d}}(p_i U^Q)(p_j U^K)^T + b_{j-i} \tag{4}$$

α_{ij} is a learnable parameter and can be viewed as the embedding of the relative position. x_i^l, x_j^l are the input to the self-attention module in the l-th layer. $W^{Q,l}$, $W^{K,l}$ are their corresponding parameter matrices. U^Q and U^K map the positional sequences p_i, p_j to the parameter matrices used for Q and K. Dividing by $\sqrt{2d}$ is carried out to maintain a_{ij} on the same scale, ensuring the importance of features is preserved. b_{j-i} represents the bias term related to relative position.

3.3. Classification and Regression

The classification and regression sub-network consists of two parts: the classification branch and the regression branch. In this paper, the classification loss function and regression loss function are designed based on varifocal loss L_{vfl} [35] and Generalized Intersection over Union (GIoU) loss L_{GIoU} [36]. This aids in learning feature representations and position estimation suitable for object tracking tasks, ultimately obtaining accurate tracking results.

3.3.1. Classification Branch

The classification branch is primarily responsible for predicting the classification response map. The IoU-Aware Classification Score (IACS) is used as the training objective for the classification loss function. Additionally, the varifocal loss L_{vfl} is employed as the training loss function. L_{vfl} is a dynamically scaled binary cross-entropy loss that utilizes an asymmetric training sample weighting approach. This involves reducing the weight of negative samples and increasing the weight of high-quality positive samples to address the

issue of class imbalance during training. The classification and varifocal loss functions are expressed as shown in Formula (5) [35]:

$$L_{cls} = L_{vfl}(p, IoU(b, \hat{b}))$$
$$L_{vfl}(p, q) = \begin{cases} -q(qlog(p) + (1-q)log(1-p)) & q > 0 \\ -\alpha p^{\gamma} log(1-p) & q = 0 \end{cases} \quad (5)$$

where p represents the predicted IACS, q represents the target score calculated from the intersection over union (IoU) of b and $\hat{b}$. b represents the predicted bounding box position, and $\hat{b}$ represents the actual bounding box position.

3.3.2. Regression Branch

The regression branch is responsible for predicting the bounding box regression map. In the regression branch, this paper employs L_{GIoU} as the regression loss function. Compared to the traditional IoU metric, L_{GIoU} extends the overlap evaluation between the predicted and actual boxes from rectangular boxes to shapes of any form. Therefore, L_{GIoU} is considered a more comprehensive target box evaluation metric, providing a more accurate regression loss for object tracking tasks. The regression loss function is expressed as shown in Formula (6) [36]:

$$L_{reg} = L_{GIoU}(b, \hat{b}) \quad (6)$$

The SSTrack's final classification and regression loss function is expressed as shown in Formula (7):

$$L_{loss} = \lambda_{cls} L_{Vfl}\ (p, IoU(b, \hat{b})) + \lambda_{reg} L_{GIoU}(b, \hat{b}) \quad (7)$$

where λ_{cls} and λ_{reg} represent the weight parameters for the classification loss and regression loss, with values of 1.2 and 1.2.

4. The Experiments

4.1. Implementation Details

The training environment of SSTrack is Ubuntu 18.04, Python version 3.8, Pytorch version 1.11.0, CUDA version 11.3, and employs an RTX 3090 GPU. The backbone is loaded with a pre-trained model of Swin V2 [37] on ImageNet-1k, and joint training is conducted using the GOT-10k, TrackingNet, and LaSOT training sets.

The algorithm's tracking capabilities and reliability are validated to comprehensively simulate tracking performance in natural and complex environments. The article conducts quantitative and qualitative comparisons with existing mainstream Siamese tracking algorithms on GOT-10k, TrackingNet, and LaSOT test sets. The test sets encompass various target categories, such as pedestrians, vehicles, and animals, and include diverse scenarios like indoor, road, and outdoor settings, making the test results more convincing.

4.2. Comparison with the Popular Trackers

4.2.1. GOT-10K

GOT-10k [38] proposed by the Chinese Academy of Sciences, is a large-scale tracking dataset comprising over 10,000 video sequences, 563 target categories, 87 sports modes (e.g., running, swimming, skiing, crawling) with 180 sequences, 84 target categories, and 32 sports modes in the test set. The average length of the test set sequence is 127 frames. Results are required to be submitted to the official platform for online evaluation, making it one of the mainstream datasets in the field of object tracking due to its relatively fair and accurate assessment process.

To avoid evaluation errors caused by imbalances in the number of categories, GOT-10k introduces class-balanced metrics, namely mAO and mSR, as evaluation criteria. Typically, thresholds of 0.5 and 0.75 are used for mSR evaluation.

The computation process for mAO and mSR is illustrated in Formula (8).

$$\begin{aligned} \text{mAO} &= \frac{1}{C}\sum_{c=1}^{C}\left(\frac{1}{|S_c|}\sum_{i\in S_c}\text{AO}_i\right) \\ \text{mSR} &= \frac{1}{C}\sum_{c=1}^{C}\left(\frac{1}{|S_c|}\sum_{i\in S_c}\text{SR}_i\right) \end{aligned} \tag{8}$$

Experimental results are shown in Table 1 for the GOT-10k. After fusing spatial scale information, our tracker achieves the highest accuracy while maintaining a real-time tracking speed of 28 fps. The mAO, $\text{mSR}_{0.5}$, and $\text{mSR}_{0.75}$ of SSTrack are 71.5%, 81.9%, and 65.5%, which are 2.6%, 2.2%, and 4% ahead of TransT using CNN as the backbone. Compared to SwinTrack, which uses Swin Transformer, it is 1.6%, 1.8%, and 1.6% ahead.

Table 1. Experiments and comparisons on GOT-10k.

Method	AO (%)	$\text{SR}_{0.5}$ (%)	$\text{SR}_{0.75}$ (%)	Speed/fps
OURS	71.5	81.9	65.5	28
SwinTrack [15]	69.9	80.1	63.9	45
CSwinTT [39]	69.6	80.0	63.1	6
TransT [14]	68.9	79.7	61.5	15
TrDimp [13]	68.8	80.4	58.6	12
TrSiam [13]	67.3	78.7	58.5	14
Siam RCNN [40]	65.0	72.8	59.8	4
Ocean [41]	61.1	71.6	49.2	34

In terms of speed, although Ocean has fast speed, its accuracy is not high enough. SwinTrack, which employs self-attention computation through sliding window partitioning, significantly improves tracking speed. In this paper, SSTrack introduces a spatial scale attention module based on the Swin Transformer. However, the tracking speed is somewhat reduced at the cost of improved accuracy. TransT requires complex self-attention computation, CSwinTT introduces Circular Shift Attention Mechanism, increasing computational complexity and resulting in lower speed and accuracy compared to our tracker.

4.2.2. TrackingNet

TrackingNet [42], the first large-scale object tracking dataset, consists of video sequences exclusively captured in real outdoor scenarios. The training dataset is exceptionally rich, encompassing over 30,000 video sequences and 14 million annotated bounding boxes. The test set comprises 511 sequences. Like GOT-10k, tracking results are evaluated online, providing a relatively fair assessment standard for object tracking algorithms. The evaluation metrics include precision (P), normalized precision (NP), and success rate (SUC). Experimental results are presented in Table 2.

Table 2. Experiments and comparisons on TrackingNet.

Method	P (%)	NP (%)	SUC (%)
OURS	79.9	86.7	81.9
CSWinTT [39]	79.5	86.6	81.7
SwinTrack [15]	78.4	85.7	81.1
TransT [14]	73.1	83.3	78.4
TrDimp [13]	73.1	83.3	78.4
TrSiam [13]	72.7	82.9	78.0
SiamPRN++ [20]	73.3	69.4	80.0

Leveraging the powerful feature extraction capabilities of the Swin Transformer, SwinTrack outperforms other tracking algorithms across various evaluation metrics. However,

due to the lack of utilization of spatial scale features, SwinTrack lags behind the SSTrack by 1.5%, 1%, and 0.8% on the three respective metrics.

4.2.3. LaSOT

LaSOT [43] comprises 1400 video sequences, with a total frame count exceeding 3.5 million and an average sequence length of 2512 frames. Utilizing precision and success rate as evaluation metrics, LaSOT's longer video sequences better reflect the algorithm's robustness.

As shown in Figure 6, for the LaSOT dataset, our algorithm achieves a precision AUC of 68.4% and a success AUC of 68.9%. In comparison to other Transformer tracking algorithms, it holds a leading position.

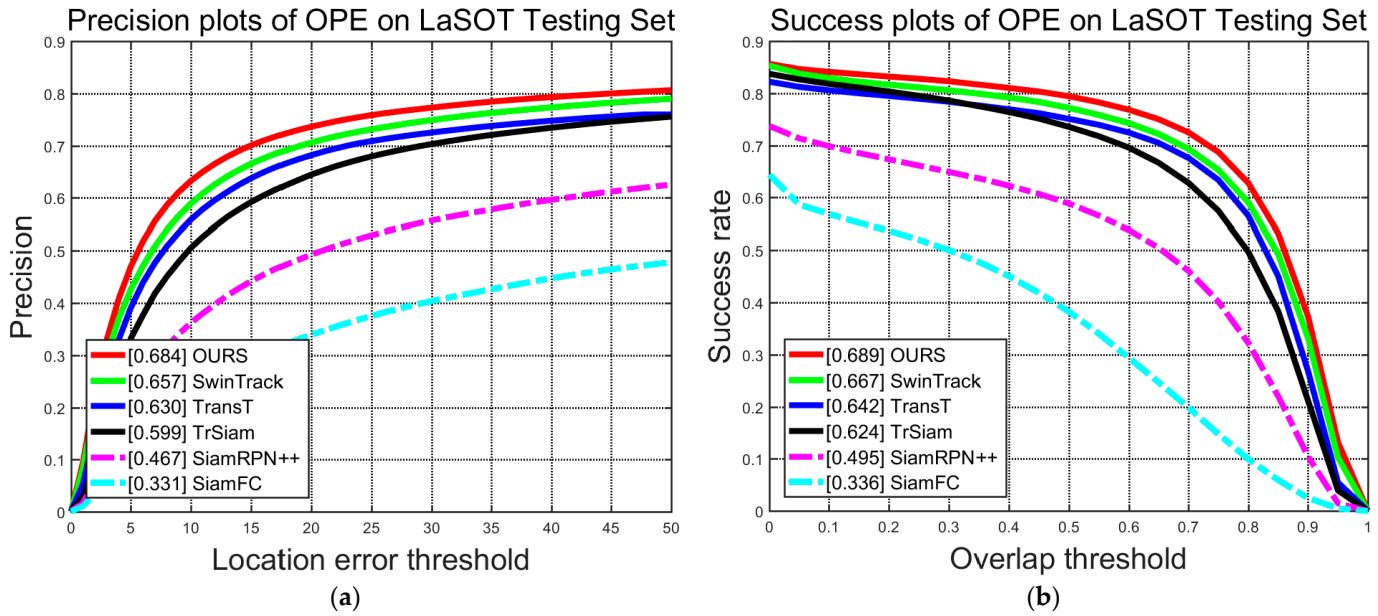

Figure 6. (**a**) SSTrack's precision curve on LaSOT; (**b**) SSTrack's success curve on LaSOT.

Figure 7 shows the precision and success rates of SSTrack and five other popular trackers across three different video sequence attributes such as Fast Motion, Illumination Variartion, and Scale Variation. SSTrack achieves the top ranking in success rate across these three attributes. Compared to the second-ranked Swin Track, SSTrack shows improvements in tracking precision by 3.5%, 2.7%, and 2.9%, and an increase in tracking success rate by 3.6%, 2.3%, and 2.2%. Compared to the Transformer algorithm, TrSiam, SSTrack demonstrates improvements in tracking precision by 7.4%, 8.2%, and 8.5%, and an increase in tracking success rate by 6.3%, 5.9%, and 6.4%. This indicates that incorporating a spatial scale attention module into the feature extraction network can effectively mitigate the issue of feature degradation caused by complex environmental conditions, outperforming a conventional Swin Transformer structure.

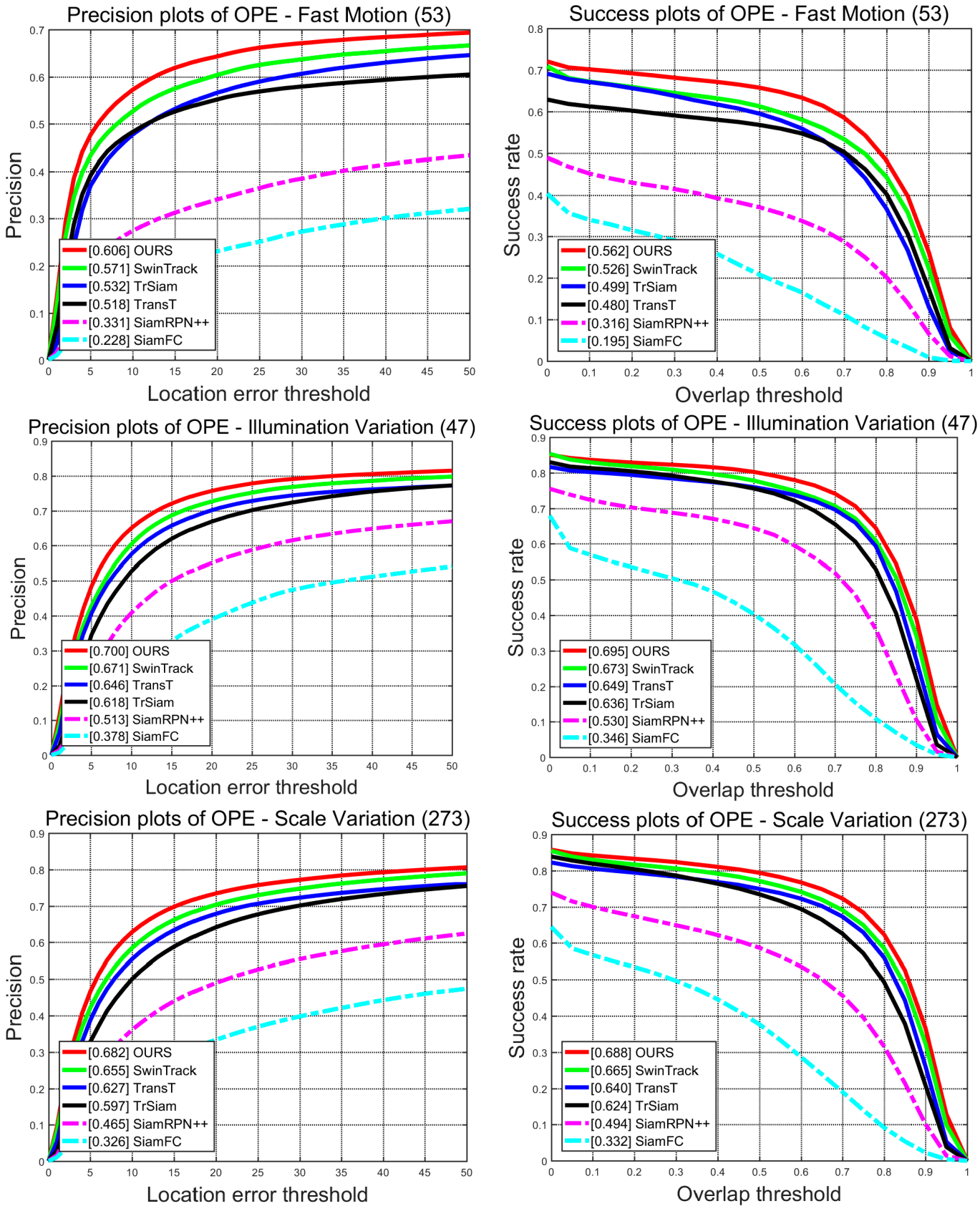

Figure 7. Experiments and comparisons on Fast Motion, Illumination Variation, and Scale Variation.

4.3. Ablation Study and Analysis

To validate the effectiveness of the SSTrack, this study conducted ablation experiments on the GOT-10k test set. The experimental environment and the main hyperparameters of the model were consistent, and the results are presented in Table 3.

Table 3. Ablation experiments.

Feature Extraction		Feature Fusion		Suc (%) ↑
ResNet	**Swin Transformer**	**Spatial Scale Attention**	**Exemplar Transfomer**	
		√	√	71.5
		√		71.1
	√			70.4
√				67.6

Note: "↑" means that the larger the Suc, the better the tracking effect, "√" represents the use of this method.

Firstly, based on the SSTrack, a success rate of 71.5% was achieved on the GOT-10k dataset. Subsequently, replacing the Exemplar Transformer used in the feature fusion stage with a regular Transformer led to a decrease in success rate to 71.2%. Following that, removing the spatial scale feature enhancement module in the feature extraction stage resulted in a success rate drop to 70.4%. This suggests that the covariance pooling and dilated convolutions, which provide local detailed information, contribute to obtaining more accurate feature representations of the target.

Finally, replacing the backbone network with ResNet-50 led to a success rate decrease to 67.6%, demonstrating that the global feature extraction capability of Swin Transformer effectively enhances tracking performance.

4.4. Qualitative Analysis

To qualitatively illustrate the SSTrack's superiority, we visualized tracking results on selected video sequences from LaSOT. These results were compared with those of mainstream tracking algorithms to intuitively assess the tracking performance of each algorithm. The study chose video sequences in complex environments, such as common scenarios involving Fast Motion (Tiger), Illumination Variation (Person), and Scale Variation (Truck), aiming to recreate realistic tracking scenes. Experimental results, as depicted in Figure 8, demonstrate that SSTrack can still achieve accurate target state estimation, yielding high-quality tracking results even in complex tracking environments.

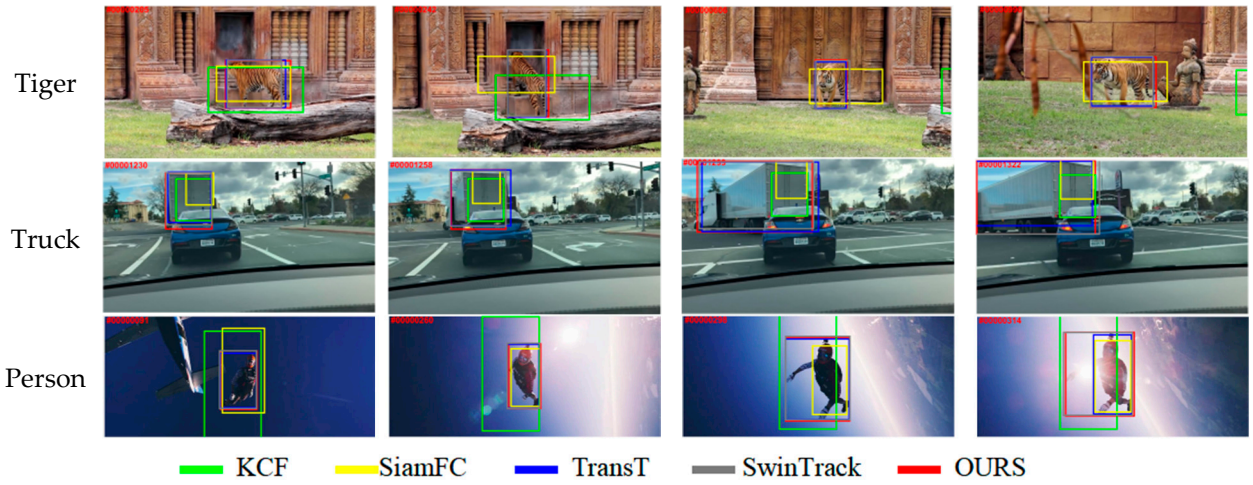

Figure 8. Visualization of tracking results.

(1) Impact of Fast Motion: In the Tiger sequences, the tiger experiences significant position changes starting from frame 220 due to its rapid movement. The traditional KCF algorithm exhibits poor feature extraction capabilities, resulting in tracking drift. Siamese tracking algorithms like SiamFC focus on the local features of the target, often including background information within the tracking box. However, Transformer tracking algorithms, represented by the approach in this paper, can accurately track the target. The tracking box in this method closely adheres to the target, achieving superior tracking performance.

(2) Impact of Scale Variation: In the Truck sequences, the truck undergoes scale changes from frame 1230 to 1322 due to a turn, introducing challenges for tracking. The proposed algorithm, incorporating spatial scale features, possesses richer semantic information during the tracking process, leading to more precise object localization than other algorithms.

(3) Impact of Illumination Variation: In the Person sequences, the parachutist is continuously affected by background light from frame 91 to 314. This interference makes algorithms like TransT unable to locate the target's outline accurately. However, SSTrack consistently achieves accurate object tracking, demonstrating that the feature extraction and feature fusion sub-networks proposed in this paper enhance the expressive power of features.

5. Conclusions

SSTrack enhances the tracking performance in complex environments such as Fast Motion, Illumination Variation, and Scale Variation, reducing issues like tracking drift and target loss while enabling real-time tracking. The main reasons for this improvement are as follows: 1. Swin Transformer possesses excellent global context information extraction capabilities, contributing to robust tracking performance. 2. Incorporating covariance pooling and multi-scale features enhances the algorithm's ability to capture detailed information about the target's local context. 3. The rational utilization of positional encoding and exemplar attention improves the effectiveness of feature fusion. Experimental results indicate that compared to Siamese tracking algorithms like SwinTrack, SiamFC and earlier KCF, SSTrack achieves performance improvements on GOT-10k, TrackingNet, and LaSOT. Compared with the Transformer tracking algorithms, like TrSiam and TransT, SSTrack meets the real-time requirements. However, SSTrack still has some limitations. As an object tracking algorithm based on Siamese networks, it only utilizes the initial frame as a template for tracking. Due to the dynamic nature of tracked objects, the initial frame may not adequately represent the current state of the target object. SSTrack may struggle in scenarios involving occlusion and similar challenges. Future improvements should consider using intermediate frames as tracking templates and incorporating a template updating mechanism to facilitate learning the evolving characteristics of the tracked object, thereby enhancing tracking robustness.

Author Contributions: Conceptualization, Q.M. and Z.H.; methodology, Z.H. and X.W.; software, Z.H. and X.W.; validation, Q.M. and Z.L.; writing—original draft preparation, Q.M. and Z.H.; writing—review and editing, X.W. and Z.L.; supervision, Q.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China, grant number 2022YFB3304401.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These datasets can be found here: LaSOT: https://vision.cs.stonybrook.edu/~lasot/, accessed on 1 March 2024; GOT-10K: http://got-10k.aitestunion.com/downloads, accessed on 1 March 2024; TrackingNet: https://github.com/SilvioGiancola/TrackingNet-devkit, accessed on 1 March 2024.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Chen, F.; Wang, X.; Zhao, Y. Visual object tracking: A survey. *Comput. Vis. Image Underst.* **2022**, *222*, 1455–1471. [CrossRef]
2. Zhang, Y.; Wang, T.; Liu, K. Recent advances of single-object tracking methods: A brief survey. *Neurocomputing* **2021**, *455*, 1–11. [CrossRef]
3. Huang, K.Q.; Chen, X.T.; Kang, Y.F. Intelligent Visual Surveillance: A Review. *Chin. J. Comput.* **2015**, *38*, 1093–1118.
4. Liang, J.; Jiang, L.; Niebles, J.C. Peeking into the future: Predicting future person activities and locations in videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5725–5734.
5. Liu, C.H.; Zhang, L.; Huang, H. Visualization of Cross-View Multi-Object Tracking for Surveillance Videos in Crossroad. *Chin. J. Comput.* **2018**, *1*, 221–235.
6. Li, P.; Chen, X.; Shen, S. Peeking into the future: Stereo r-cnn based 3d object detection for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7644–7652.
7. Xie, J.C.; Xi, R.; Chang, D.F. Mask wearing detection based on YOLOv5 target detection algorithm under COVID-19. *Acadlore Trans. AI Mach. Learn.* **2022**, *1*, 40–51. [CrossRef]
8. Lu, H.C.; Fang, G.L.; Wang, C. A novel method for gaze tracking by local pattern model and support vector regressor. *Signal Process.* **2010**, *90*, 1290–1299. [CrossRef]
9. Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
10. Meng, L.; Yang, X. A Survey of Object Tracking Algorithms. *Acta Autom. Sin.* **2019**, *45*, 1244–1260.
11. Lu, H.C.; Li, P.X.; Wang, D. Visual Object Tracking: A Survey. *Pattern Recognit. Artif. Intell.* **2018**, *32*, 61–76.
12. Hou, Z.Q.; Guo, F.; Yang, X.L. Transformer Visual Object Tracking Algorithm Based on Mixed Attention. *Control Decis.* **2022**, *39*, 739–748.
13. Wang, N.; Zhou, W.; Wang, J.; Li, H. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 1571–1580.
14. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8126–8135.
15. Wei, X.; Bai, Y.; Zheng, Y.; Shi, D.; Gong, Y. Autoregressive visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 9697–9706.
16. He, K.; Zhang, C.; Xie, S. Target-Aware Tracking with Long-term Context Attention. In Proceedings of the Association for the Advancement of Artificial Intelligence, Washington, DC, USA, 7–14 February 2023.
17. Blatter, P.; Kanakis, M.; Danelljan, M. Efficient visual tracking with Exemplar Transformers. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–7 January 2023; pp. 1571–1581.
18. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision Workshops, Amsterdam, The Netherlands, 7–10 October 2018; pp. 850–865.
19. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-aware siamese networks for visual object tracking. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 101–117.
20. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4282–4291.
21. Wang, Q.; Zhang, L.; Bertinetto, L. Fast online object tracking and segmentation: A unifying approach. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4282–4291.
22. Liu, M.; Shi, J.; Wang, Y. Dual-Template Siamese Network with Attention Feature Fusion for Object Tracking. *Radioengineering* **2023**, *32*, 371–381. [CrossRef]
23. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
24. Cui, Y.; Jiang, C.; Wang, L.; Wu, G. Mixformer: End-to-end tracking with iterative mixed attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13608–13618.
25. Sanagavarapu, K.S.; Pullakandam, M. Object Tracking Based Surgical Incision Region Encoding using Scalable High Efficiency Video Coding for Surgical Telementoring Applications. *Radioengineering* **2022**, *31*, 231–242. [CrossRef]
26. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
27. Deore, S.P.; Bagwan, T.S.; Bhukan, P.S.; Rajpal, H.T.; Gade, S.B. Enhancing Image Captioning and Auto-Tagging through a FCLN with Faster R-CNN Integration. *Inf. Dyn. Appl.* **2023**, *3*, 12–20. [CrossRef]

28. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 × 16 words: Transformers for image recognition at scale. In Proceedings of the Ninth International Conference on Learning Representations, Virtual, 3–7 May 2021; pp. 1–22.
29. Liu, Z.; Lin, Y.; Cao, Y. Swin Transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 568–578.
30. Lin, L.; Fan, H.; Zhang, Z.; Xu, Y.; Ling, H. SwinTrack: A simple and strong baseline for transformer tracking. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; pp. 16743–16754.
31. Dai, T.; Cai, J.; Zhang, Y. Second-order attention network for single image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 11065–11074.
32. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. In Proceedings of the International Conference on Learning Representations 2016, San Juan, Puerto Rico, 2–4 May 2016; pp. 1–13.
33. Mehta, S.; Rastegari, M.; Caspi, A. ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation. In Proceedings of the IEEE/CVF International Conference on European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 552–568.
34. Ke, G.; He, D.; Liu, T.Y. Rethinking positional encoding in language pre-training. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 3–7 May 2021; pp. 13608–13618.
35. Zhang, H.; Wang, Y.; Dayoub, F. Varifocalnet: An iou-aware dense object detector. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 8514–8523.
36. Rezatofighi, H.; Tsoi, N.; Gwak, J.Y. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
37. Liu, Z.; Hu, H.; Lin, Y. Swin transformer v2: Scaling up capacity and resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12009–12019.
38. Huang, L.; Zhao, X.; Huang, K. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1562–1577. [CrossRef] [PubMed]
39. Mayer, C.; Danelljan, M.; Bhat, G.; Paul, M.; Paudel, D.P.; Yu, F.; Van Gool, L. Transforming model prediction for tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8731–8740.
40. Voigtlaender, P.; Luiten, J.; Torr, P.H.S. Siam r-cnn: Visual tracking by re-detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6578–6588.
41. Zhang, Z.; Peng, H.; Fu, J.; Li, B.; Hu, W. Ocean: Object-aware anchor-free tracking. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 771–787.
42. Muller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; Ghanem, B. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 300–317.
43. Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; Ling, H. LaSOT: A high-quality benchmark for large-scale single object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5369–5378.