ISSN: 1813-4890

MDASiam: A siamese network tracker introducing meta-learning

Junyi Wang[1, a] , Tian Zhou [2, b]

[1] School of Artificial Intelligence Hubei University, Wuhan, 430062, China.;

[2]School of Computer and Information Engineering Hubei University, Wuhan, 430062, China.

[a]752249781@qq.com, [b]2628895366@qq.com

Abstract

Siamese network has become one of the research hotspots for visual target tracking, but when it comes to complex scene changes, the existing siamese network suffers from the problem that the target feature information extraction is not abundant enough. An improved siamese network target tracking algorithm MDASiam, which adopts a deep residual network to extract target features, improve the discriminative ability of the network model, improve the backbone network module to efficiently utilize the network feature information and improve the tracking performance of the algorithm; this paper also incorporates a meta-learning network to obtain an adaptive target feature space to overcome the complex changes of target appearance. The test results on the OTB100 dataset show that the MDASiam algorithm has robust tracking performance in complex scenarios such as background interference, target deformation, in-plane rotation and scale change, among which the in-plane rotation scenario performs particularly well, with a 15.5% improvement in success rate and 12.0% improvement in accuracy compared to the benchmark SiamFC algorithm.

Keywords

Visual object tracking, Convolutional neural network, Siamese network, Meta learning.

1. Introduction

Visual object tracking has always been a fundamental and challenging task in the fields of artificial intelligence and computer vision. In recent years, visual object tracking has found extensive applications in areas such as intelligent video surveillance, drones, robotics, and more [1]. However, real-world tracking scenarios often present challenges like low resolution, background interference, target deformations, occlusions, scale changes, and lighting variations. As a result, developing real-time and accurate object tracking algorithms in the face of these challenges has become a hot topic and a formidable research task.

The core problem of object tracking is to differentiate the foreground object from a complex background. Given the initial position of any target in the first frame, the tracker aims to successfully distinguish and locate that target in subsequent frames [2]. Traditional tracking algorithms from the last century, such as Mean Shift [3] and Kalman filtering, paved the way for tracking. The CSK algorithm proposed by Okada et al. [4] introduced cyclic shift samples to increase the number of training samples and improve template efficiency. The Kernelized Correlation Filter (KCF) algorithm, developed by Henriques et al. [5], utilized HOG image features as algorithm inputs and optimized the computational efficiency using a Gaussian kernel function, achieving a minor breakthrough in correlation filter-based tracking. Subsequently, Danelljan et al. introduced the Correlation Filter with Circulant Matrix (CCOT) and the Efficient Convolution Operators (ECO) algorithms [6,7]. These algorithms aimed to reduce the impact of boundary effects and enhance tracking performance by employing more efficient kernel functions, as well as multi-feature fusion methods.

Traditional tracking algorithms have faced challenges such as insufficient feature extraction and limited precision in the network framework, making them less effective in handling complex scenarios like background interference and target deformations. In recent years, researchers have shown widespread interest in neural network frameworks with richer feature information and higher precision. One such algorithm is MDNet, proposed by Hyeonseob et al. [8], which is based on a Convolutional Neural Network (CNN) framework. MDNet uses video training to obtain a network structure capable of efficient binary classification and performs online updates. Another algorithm, TCNN, introduced by Ashutosh et al. [9], utilizes multiple convolutional networks (CNNs) to jointly represent the target state and determine the path for model updates within a decision tree. Subsequently, Siamese networks gained popularity due to their simplicity and efficiency. The SiamFC algorithm, developed by Luca Bertinetto et al. [10], is a Siamese Fully Convolutional Network. It utilizes two network branches, one for the template and the other for the target. By calculating similarity through correlation layers, it achieves good performance in terms of speed and accuracy, ranking first in the VOT2017[11] real-time challenge.

While the SiamFC Siamese network tracker has achieved a certain level of tracking accuracy, it still faces some challenges. One of its limitations is that it uses a relatively shallow network, AlexNet [12], for extracting convolutional feature information, which results in less rich feature learning. SiamFC also forgoes the time-consuming online updating approach, making it less robust in handling complex scene variations and more susceptible to interference from background information, which can lead to tracking drift.

The research team attempted to replace the shallow network AlexNet with deeper networks such as VGG [13] and ResNet [14]. However, the experimental results showed a decrease in performance. Upon investigation, it was found that these deeper networks were optimized for image classification tasks. In image classification tasks, the precise localization of specific objects is not highly weighted in the network. Therefore, they cannot be directly applied to tracking tasks.In neural network architecture, the receptive field [15] refers to the size of the region in the original image that is mapped to a pixel in the output feature map of each layer in a CNN. If the receptive field and the size of the target object differ significantly, it can lead to convergence difficulties and significantly impact algorithm performance.In image processing tasks, the stride of a neural network represents the sampling interval of the convolution kernel as it passes through the input feature map. The stride directly affects the accuracy of feature information processing in CNNs.

This paper, based on the SiamFC tracking algorithm, conducted numerous experiments and found that the presence or absence of edge padding in CNNs has a significant impact on tracker performance. This effect is particularly pronounced when the tracked target is very close to the boundary of the candidate search region. The presence or absence of edge feature padding can lead to substantially different results in target localization. Consequently, the study concludes that in the task of object tracking, the neural network's receptive field, stride, and edge padding are the main factors influencing whether the use of deep network structures can improve tracking performance. Furthermore, adapting to changes in the target's appearance is often challenging, and obtaining an ideal representation of target feature information is a pressing issue for researchers. Past algorithms often designed a candidate set for multi-scale template matching. When generating candidate samples in the motion model, a large number of candidate boxes with varying sizes are generated, or tracking is performed on multiple targets of different scales, producing multiple predictions. The best result among these predictions is then chosen as the final tracking target. However, in optimizing object tracking models, classifier overfitting often occurs due to the insufficient training dataset, and this can lead to difficulties in accurately recognizing target changes during tracking, resulting in target loss in such update strategies.

To address the aforementioned issues, this paper replaces the shallow network AlexNet used in SiamFC with a deeper ResNet network. This allows the neural network to receive more accurate feature information. Furthermore, the main network structure has been improved to better suit the tracking task's requirements, resulting in enhanced tracker performance.In addition, to tackle the problem of target loss due to insufficient training data and the inability to accurately identify target changes during appearance variations, this paper incorporates a meta-learning network. This meta-learning network can accurately acquire target feature information and its changes with a small amount of sample data, enabling it to predict the most suitable target position in the next frame. By adaptively learning target size parameters through the meta-learning network and optimizing the iterative size appearance of target features and search areas, an adaptive target feature space is provided to the tracking network. This substantial improvement greatly enhances the system's performance.

2. MDASiam tracker

After extensive experimental work, this paper proposes an object tracking algorithm called MDASiam, which incorporates a meta-learning module. MDASiam utilizes a deeper CIResNet-22 network as the target feature extractor, effectively enhancing the discriminative capabilities of the network model. Additionally, it employs a meta-learning network to adaptively learn target feature scale parameters, iterating to generate the most suitable feature representation space for the tracking task. This ensures that the tracker can adapt to complex changes in the target's appearance, thereby improving the algorithm's tracking performance

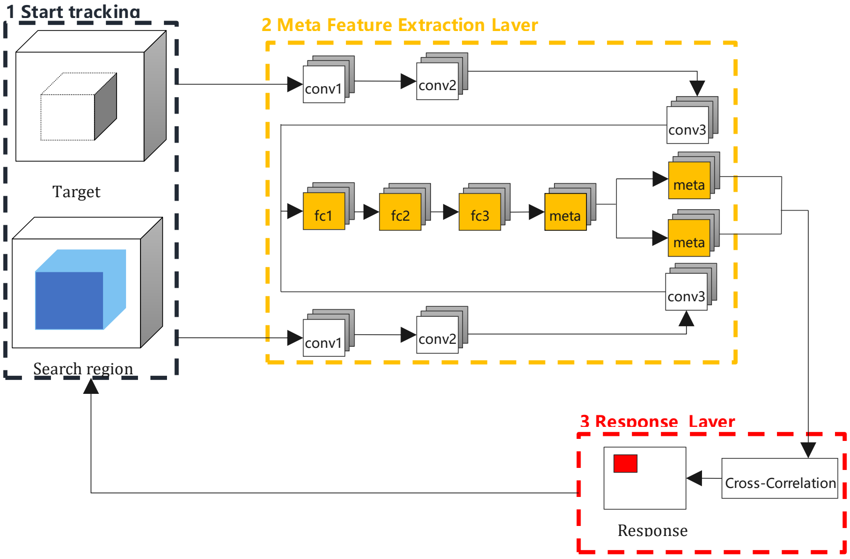


Fig.1 Framework of the MDASiam algorithm

By optimizing through the meta-network, an adaptive appearance model is obtained. Then, using the Siamese network, similarity is computed to generate the correlation response maps between the target and candidate regions. The most likely position for the target in the current frame is predicted based on the maximum value in the correlation response map. As shown in Fig.1, the MDASiam network structure consists of three main components.

(1) The first step involves selecting the template region and the search region. "Target" represents the tracking target provided in the first frame, while the "Search Region" refers to the candidate search area determined by the algorithm based on the previous frame's target location. The size of the search candidate area is typically set to 1.5 to 2 times the size of the target region. Additionally, the algorithm chooses an appropriate search region based on the target's location.

(2) The second part of the network consists of the feature extraction layer. In this paper, an improved CIResNet-22 network is used to extract the initial target features. These target features extracted from both the target and search images are then passed to the meta-learning prediction network. This process results in obtaining more precise target features in terms of size and appearance, ensuring that the algorithm can adapt to various changes in the target's appearance.

(3) The third part of the network is the correlation calculation module. It processes the final target features obtained from the previous layers through convolution to generate response maps. The highest score in the correlation response map represents the predicted position of the target in the current frame.

2.1. MDASiam network improvement unit design

2.1.1. Cropping-Inside Residual (CIR)

In residual networks, residual units play a crucial role in enabling convolutional kernels to capture richer information representations. The residual unit is the most important module in a residual network. As shown in Fig.2(a), the original residual unit consists of three stacked convolutional layers with a skip connection.

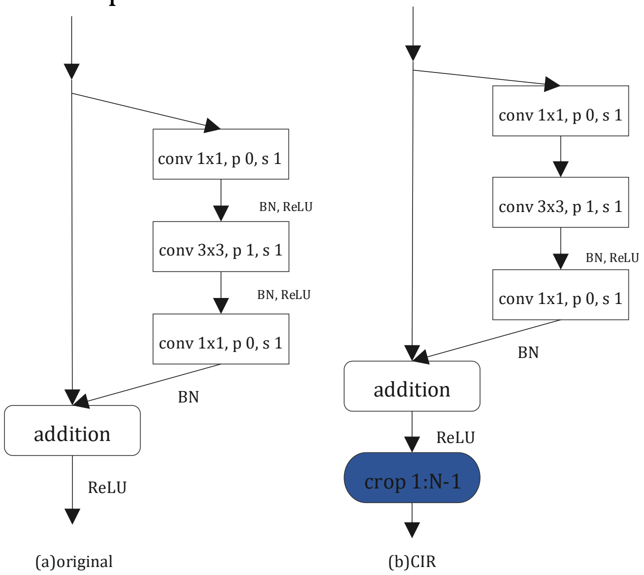
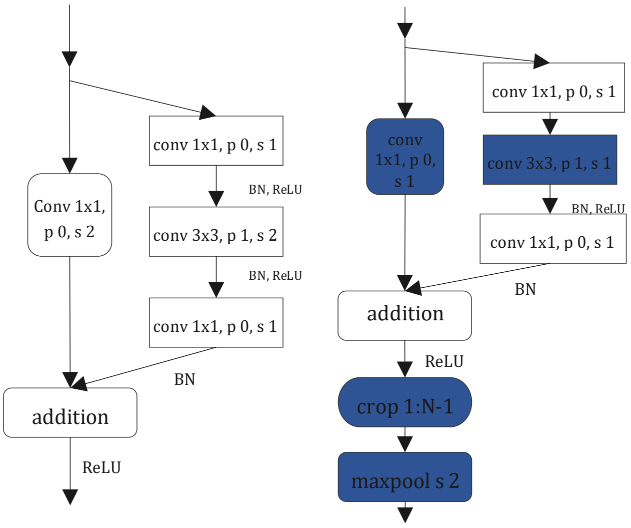


Fig.2 CIR unit of the MDASiam algorithm

Based on research analysis, it was found that the feature padding in the original ResNet network significantly affects the performance of the Siamese network-based object tracker. To address this issue, the research team decided to remove the feature padding when using a residual network as the main network for feature extraction in the tracking algorithm. They applied an improved residual unit called Cropping-Inside Residual (CIR)[16], which trims the outermost part of the residual unit affected by the feature padding while retaining other valuable feature information. As shown in Fig.2(b), this improved network structure allows the object tracker to obtain a more rich feature representation, thereby enhancing its performance.

2.1.2. Cropping-Inside Residual with Downsampling (CIR-D)



(a)original (b)CIR

Fig.3 CIR-D unit of the MDASiam algorithm

Within residual networks, there is a need for a structure that can effectively reduce the spatial size of feature maps while increasing the number of feature channels. Hence, another crucial component within residual networks, known as the Cropping-Inside Residual with Downsampling (CIR-D) block, has been introduced, as illustrated in Fig.3(a). This CIR-D unit is designed to downsample feature maps, enabling higher-level feature representations and contributing to improved network performance.Similarly, in order to mitigate the impact of feature padding on CIR-D, improvements to its network structure are required.

As shown in Fig.3(b), similar to the Inner Cropping Residual Unit, cropping is performed at the end of the downsampling residual network to remove the outermost part while retaining other feature information. To ensure that even the outermost layers of the network can receive feature information, this paper designs the downsampling operation as a maximum pooling operation[17]. Additionally, the convolutional strides for the skip connections and bottleneck layers are set to 1 in order to maintain the stability of the network's internal structure performance. Through these operations, it is ensured that as the network depth increases, effective feature information is also collected, thus ensuring the improvement of network performance.

2.1.3. CIResNet-22 network architecture design

Table 1 Improved algorithm network structure

Stage	CIRes22
Conv1	7×7,64, stride 2
Conv2	2×2, max pool, stride 2 [1x1,64 3x3,64 x3 1x1,256]
Conv3	[1x1,128 3x3,128 x4 1x1,512] [1x1,128] x4
Response	cross correlation

In this paper, an improved CIResNet-22[18] network is used as the backbone network for extracting target features. As shown in the table above, the network structure consists of three main parts and includes 22 convolutional layers. The first part consists of a 7x7 convolutional layer, following the original residual structure 2(a), while the rest are composed of CIR units 2(b). In the second part, the network's stride is set to 4, and cropping layers with a size of 2 are introduced to reduce the impact of feature padding on the network structure. In the first two parts, the research team employed the original ResNet structure for downsampling. In the third part, CIR-D units from 3(b) are used to perform feature downsampling, with the network's stride set to 8. Additionally, the receptive field size of the last layer of the residual network is set to 70% of the sample image size. This design ensures an organic combination of the receptive field, network stride, and feature padding. By using the improved residual network as the backbone network for feature extraction in the twin-object tracking algorithm, this paper successfully establishes a deeper and higher-performing target tracking algorithm network.

2.2. Meta-learning networks in the MDASiam

In tracking tasks, real-world scenarios often involve unpredictable changes in the appearance of the target. This gives rise to two challenges in tracking algorithms: how to obtain more suitable feature information and how to adaptively cope with changes in the target's appearance. Many tracking algorithms have been improved along these two directions. As researchers have become more adept at optimizing CNN-based applications, we can efficiently extract more accurate target feature information. However, adapting to changes in the target's appearance remains a pressing issue. In light of this, this paper introduces meta-learning networks to provide the tracking network with more appropriate feature dimensions for the target to be tracked and the search area.

The feature extraction architecture of the MDASiam algorithm primarily consists of two components: the matching network and the meta-learning network. The matching network conveys meta-information to the meta-learning network, while the meta-learning network, in turn, provides the matching network with the adaptive target feature space required for tracking. The matching network is structured as a convolutional Siamese network, having two inputs: z, which represents the sample image (the target to be tracked), and x, which represents the search region image in the current frame, selected based on the target's previous frame location. The matching network employs a deep feature extractor through a CNN network to extract feature functions $\emptyset_w(\cdot)$ from the input images and, by employing cross-correlation operations between these feature functions, ultimately obtains the response map $f_w(z,\mathrm{x})$, as shown in Formula (1):

$$f_w(z,\mathrm{x}) = \emptyset_w(z) * \emptyset_w(x) \tag{1}$$

In the formula, the $*$ symbol represents the cross-correlation operator between the two feature functions, and $w = \{w_1, w_2, \cdots, w_N\}$ represents the set of trained kernel weights for the various layers of the feature extractor CNN.

2.2.1. Meta-learning networks and gradient information

Gradients have found increasingly wide applications in image processing. As shown in Fig.4, the yellow box represents the tracking target. In the first column, the research team attempted to mask the target region with a black rectangle. In the second column, the deep red color represents regions with higher gradient values, and it is evident that there are significant gradient magnitudes in the occluded region of the target.To further investigate the role of gradients, in the third column, this paper conducted experiments in a scenario with a similar background interference, and it was found that even when the target and the background are similar, there are still significant differences in the gradients between them. Therefore, the research team concluded that gradient information can effectively represent changes in the target and the spatial relationship between the target and the background[19].

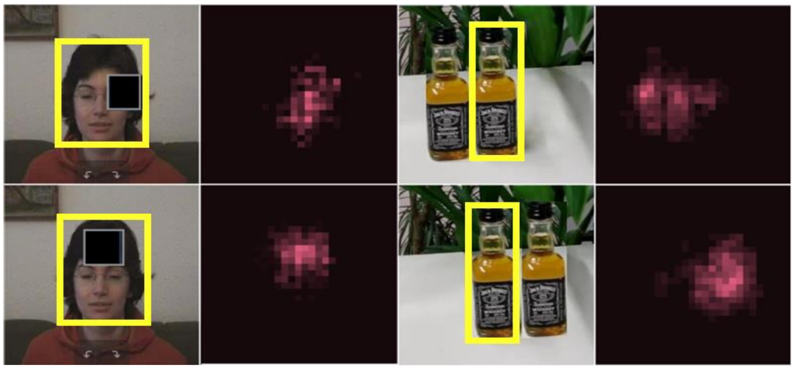


Fig.4 Gradient information

The meta-learning network provides specific target weights to the matching network, given the tracking target z and the context patch $x_\delta = \{x_1, \dots, x_M\}$ cropped around the target. In order to adapt the weights to the tracking target, this paper utilizes the iterative update based on the average negative gradient of the last layer's loss function in the matching network. The average negative gradient δ[20] is computed as follows：

$$\delta = \sum_{I=1}^{M} -\frac{1}{M}\frac{\partial l(f_W(z,x_i),\hat{y}_i)}{\partial w_N} \tag{2}$$

Where $\hat{y}$ is the generated binary correlation response graph assuming that the target is located at the correct location z_i within the context patch. The design of the meta-learning network is based on the fact that the features of δ in the meta-learning network vary with the learning object. Then, given as input, the meta-learning network $g_\theta(\cdot)$ can generate target-specific weights corresponding to the input w^{target} :

$$w^{target} = g_\theta(\delta) \tag{3}$$

Where θ is the parameter to be optimized in the meta-learning network. The new weights are used to update the original weights of the matching network, as shown in Equation (4):

$$f_{w^{adapt}}(z,\mathrm{x}) = \emptyset_{w^{adapt}}(z) * \emptyset_{w^{adapt}}(x) \tag{4}$$

Where $w^{adapt} = \{w_1, w_2, \dots .[w_N, w^{target}]\}$, connects w^{target} to the last layer w_N for feature extraction. The meta-learning network also generates sigmoid attention weights for each channel of the feature mapping to further adjust the feature representation space, and these weights can be applied by channel multiplication.

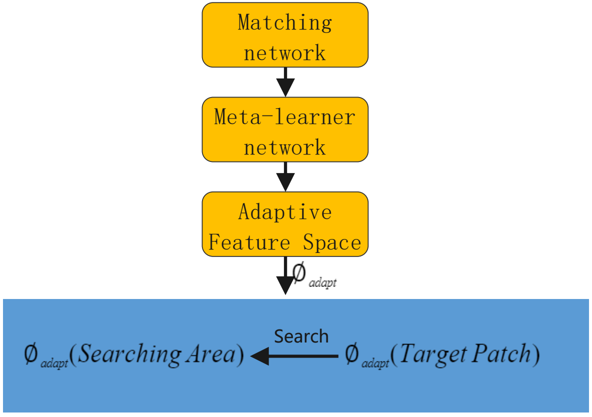


Fig.5 Schematic diagram of the meta-learning algorithm

In Fig.5, the target gradient δ from the final layer of the matching network is transferred to the meta-learning network, obtaining adaptive target features and search candidate region feature blocks. Through the meta-learning network, input weight calculations are performed without any iterative optimization, avoiding overfitting. With just a single forward pass, a rapidly adaptive target feature space can be constructed [21].

3. Experimentation and Analysis

The tracking task was carried out on the OTB100 dataset to assess the performance of the MDASiam algorithm. Additionally, the research team compared the test metric results of the MDASiam algorithm with those of relevant mainstream algorithms to verify whether its tracking performance meets expectations. As illustrated in Fig.6, MDASiam performs well on the OTB dataset. The videos from left to right include basketball, bird1, boy, and carscale. The tracked target is denoted by the green box, and the predicted target position in the current frame is indicated by the yellow box. The yellow number in the top left corner of each frame represents the frame number in the video sequence.

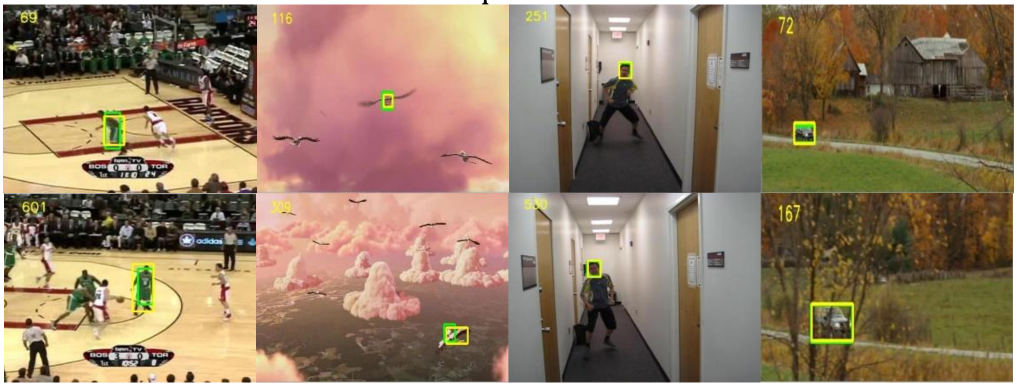


Fig.6 Tracking results of the MDASiam on OTB video

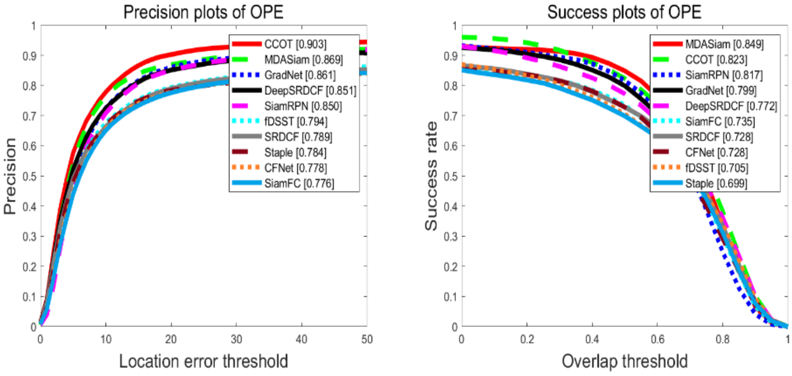
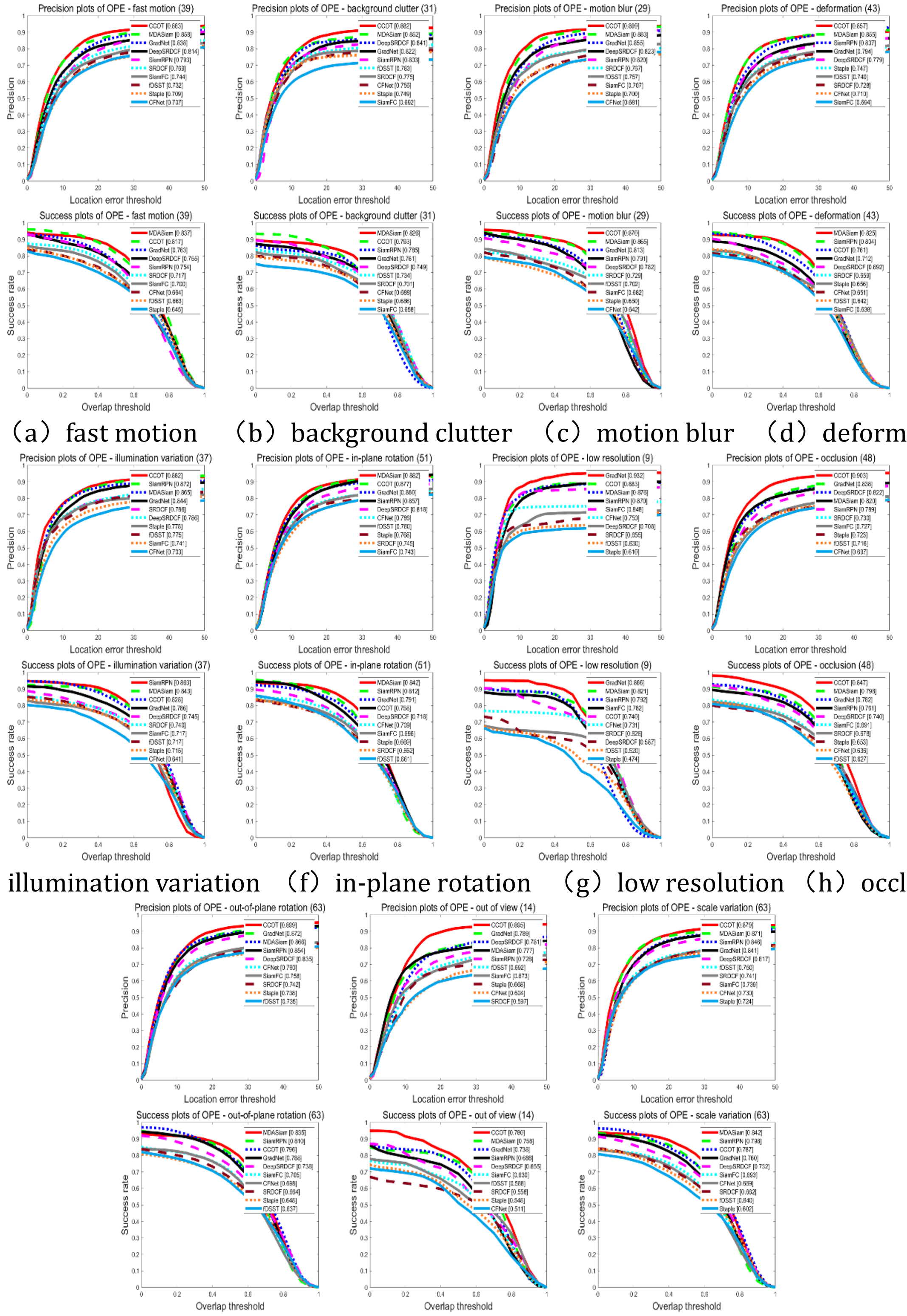


Fig.7 Comparison of precision rate and success rate of each algorithm on OTB100

The performance of the tracking algorithm is evaluated using two metrics: Precision Rate and Success Rate. Precision Rate is defined as the success tracking rate within a given 20-pixel threshold for the Euclidean distance error between the predicted box and the ground truth box center. Success Rate is calculated by measuring the overlap of pixels between the algorithm's predicted box and the ground truth box region.Firstly, a one-time analysis is employed to quantitatively assess the distance precision and threshold success rate of the tracking algorithm MDASiam, comparing it with mainstream trackers on the OTB100 dataset.

The experiment selected nine tracking algorithms for comparison on the OTB100 dataset, including the baseline algorithm SiamFC, deep convolutional neural network algorithms such as SiamRPN[22], DeepSRDCF, and correlation filter-based methods like Staple[23], CCOT, CFNet, FDSST, SRDCF, and the gradient feature-based GradNet. As shown in Fig. 7, the MDASiam algorithm outperforms the comparison algorithms in tracking success rate.For this algorithm, at an overlap rate of 0.5 and a center error of 20, the success rate and precision rate are 0.849 and 0.869, respectively, both higher than most comparison algorithms. Compared to the SiamFC algorithm that uses AlexNet as the backbone network, the success rate has increased by 15.5%, and the precision rate has improved by 12.0%. This further validates the superior tracking performance of the MDASiam algorithm.

In order to further investigate the strengths and weaknesses of the algorithm, a specific evaluation was conducted to assess the performance of the MDASiam target tracking algorithm in various complex scenarios, comparing it with relevant mainstream algorithms. Fig.8 illustrates the success rate and precision rate of each algorithm in 11 complex scenarios, including fast motion, background interference, motion blur, and target deformation. In these diverse and challenging scenarios, the MDASiam algorithm demonstrates excellent tracking performance, highlighting its capability to handle complex tracking situations effectively.



（a）fast motion （b）background clutter （c）motion blur （d）deformation

（e）illumination variation （f）in-plane rotation （g）low resolution （h）occlusion

（i）out-of-plane rotation （j）out of view （k）scale variation

Fig.8 Tracking comparison of each algorithm on OTB100

As shown in the figure, the MDASiam tracking algorithm exhibits significantly improved performance over the baseline SiamFC algorithm in the mentioned 11 scenarios. Moreover, it performs optimally or sub-optimally in scenarios involving fast motion, background interference, motion blur, target deformation, in-plane rotation, and scale changes. Particularly noteworthy is its outstanding performance in handling in-plane rotation. This further validates the strong discriminative capability of the MDASiam tracking algorithm, showcasing its ability to discern and track targets effectively in a variety of complex scenarios.

4. Conclusion

To enhance the accuracy of the Siamese network-based target tracking algorithm, a Siamese network tracking algorithm called MDASiam is proposed, incorporating a meta-learning module. It employs a deeper CIResNet-22 network to extract target features, providing primary features to the matching network, thus effectively improving the discriminative capability of the network model. Simultaneously, the meta-learning network is utilized to adaptively learn target feature scale parameters, iteratively generating a feature representation space that best suits the tracking task. This ensures that the tracker can adapt to the complex variations in the appearance of the target.

The experiments on the OTB dataset demonstrate that the MDASiam tracking algorithm exhibits excellent tracking performance and maintains robustness across various complex scenarios. However, it involves a relatively deep network framework, which may consume significant computational resources during runtime. If applied to small devices such as drones, it demands high-performance onboard computers. Additionally, the algorithm's training dataset is singular, which may lead to overfitting. In future work, the plan is to validate the tracker on different datasets to enhance its performance.

References

[1] J.H.Tan, W.Yin, L.Liu, et al.DenseNetsiamese network with global context feature module for object tracking[J]. Journal of Electronics & Information Technology, 2021, 43(01): 179–186.

[2] Z.W.He, J.H.Nie, C.J.Du, et al.Siamese Object Tracking Based on Key Feature Information Perception and Online Adaptive Masking [J]. Journal of Electronics & Information Technology, 2022, 44(05):1714-1722.

[3] Irene Anindaputri Iswanto, Tan William Choa, Bin Li.Object tracking based on meanshift and particle-kalman filt er algorithm with multi features[J].Procedia Computer Science , 2019.

[4] Okada M, Nada S, Yamanashi Y, et al. CSK: a protein-tyrosine kinase involved in regulation of src family kinases[J]. Journal of Biological Chemistry, 1991, 266(36): 24249-24252.

[5] Henriques J F, Caseiro R, Martins P, et al. High-speed tracking with kernelized correlation filters[J]. IEEE transactions on pattern analysis and machine intelligence, 2014, 37(3): 583-596.

[6] Danelljan M, Robinson A, Khan F S, et al. Beyond correlation filters: Learning continuous convolution operators for visual tracking[C] //European conference on computer vision. Springer, Cham, 2016: 472-488.

[7] Danelljan M, Bhat G, Shahbaz Khan F, et al. Eco: Efficient convolution operators for tracking[C] //Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 6638-6646.

[8] H. Nam and B. Han, Learning Multi-domain Convolutional Neural Networks for Visual Tracking, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, 4293-4302.

[9] A. Pandey and D. Wang, "TCNN: Temporal Convolutional Neural Network for Real-time Speech Enhancement in the Time Domain," ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, 6875-6879.

[10] Bertinetto L, Valmadre J, Henriques J F, et al. Fully-convolutional siamese networks for object tracking[C]. In: European conference on computer vision. Springer, Cham, 2016: 850-865.

[11] Kristan M, Leonardis A, Matas J, et al. The visual object tracking VOT2017 challenge results. In Proceedings - 2017 IEEE International Conference on Computer Vision Workshops,2017.

[12] Alom M Z, Taha T M, Yakopcic C, et al. The history began from alexnet: A comprehensive survey on deep learning approaches[J]. arXiv preprint arXiv:1803.01164, 2018.

[13] Yu W, Yang K, Bai Y, et al. Visualizing and comparing AlexNet and VGG using deconvolutional layers[C]//Proceedings of the 33 rd International Conference on Machine Learning. 2016.

[14] Wu Z, Shen C, Van Den Hengel A. Wider or deeper: Revisiting the resnet model for visual recognition[J]. Pattern Recognition, 2019, 90: 119-133.

[15] Traore B B, Kamsu-Foguem B, Tangara F. Deep convolution neural network for image recognition[J]. Ecological Informatics, 2018, 48: 257-268

[16] Zhang Z, Peng H. Deeper and wider siamese networks for real-time visual tracking[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 4591-4600.

[17] Yang K, Song H, Zhang K, et al. Deeper Siamese network with multi-level feature fusion for real-time visual tracking[J]. Electronics Letters, 2019, 55(13): 742-745.

[18] Zhang D, Zheng Z. Joint Representation Learning with Deep Quadruplet Network for Real-Time Visual Tracking[C]//2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020: 1-8.

[19] Li P, Chen B, Ouyang W, et al. Gradnet: Gradient-guided network for visual object tracking[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 6162-6171.

[20] Asadi Soodabeh,Povh Janez. A Block Coordinate Descent-Based Projected Gradient Algorithm for Orthogonal Non-Negative Matrix Factorization[J]. Mathematics,2021,9(5).

[21] Choi J, Kwon J, Lee K M. Deep meta learning for real-time target-aware visual tracking[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 911-920.

[22] Li B, Yan J, Wu W, et al. High performance visual tracking with siamese region proposal network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8971-8980.

[23] Valmadre J, Bertinetto L, Henriques J, et al. End-to-end representation learning for correlation filter based tracking[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2805-2813.