

Deep Reinforcement Learning Object Tracking Based on Actor-Double Critic Network

Jing Xin¹, Jianglei Zhou¹, Xinhong Hei² ✉, Pengyu Yue¹, and Jia Zhao¹

ABSTRACT

Aiming at the problem of poor tracking robustness caused by severe occlusion, deformation, and object rotation of deep learning object tracking algorithm in complex scenes, an improved deep reinforcement learning object tracking algorithm based on actor-double critic network is proposed. In offline training phase, the actor network moves the rectangular box representing the object location according to the input sequence image to obtain the action value, that is, the horizontal, vertical, and scale transformation of the object. Then, the designed double critic network is used to evaluate the action value, and the output double Q value is averaged to guide the actor network to optimize the tracking strategy. The design of double critic network effectively improves the stability and convergence, especially in challenging scenes such as object occlusion, and the tracking performance is significantly improved. In online tracking phase, the well-trained actor network is used to infer the changing action of the bounding box, directly causing the tracker to move the box to the object position in the current frame. Several comparative tracking experiments were conducted on the OTB100 visual tracker benchmark and the experimental results show that more intensive reward settings significantly increase the actor network's output probability of positive actions. This makes the tracking algorithm proposed in this paper outperforms the mainstream deep reinforcement learning tracking algorithms and deep learning tracking algorithms under the challenging attributes such as occlusion, deformation, and rotation.

KEYWORDS

object tracking; deep reinforcement learning; actor-double critic network

The purpose of object tracking is to determine the state of an object in subsequent video frames given the initial state of the object in the first frame. It has been widely used in unmanned system^[1-4], intelligent video surveillance^[5-7], intelligent medical services^[8,9], smart factory^[10], and other fields. In practical application, some challenging scenes, such as occlusion, deformation, and rotation of the object, often appear, which will lead to tracking drift and even tracking failure. In order to solve these problems, many solutions have been proposed, including traditional algorithms and deep learning (DL) object tracking algorithms. The DL object tracking algorithm uses the powerful perception ability of DL to extract good features of the object, and the tracking precision and robustness have been greatly improved compared with the traditional tracking algorithm, such as MDNet tracking. How to ensure tracking accuracy while maintaining good speed is still an open problem.

In recent years, with the growing application of reinforcement learning (RL) theory in areas such as UAV search^[11,12], multi-agent coordination^[13], gaming^[14], robot autonomous navigation^[15], and other fields. Researchers have begun to combine DL and RL for object tracking. Deep reinforcement learning (DRL) can effectively use the relevant information between successive frames to track, which reduces the complex calculation of template matching in DL object tracking, and well deals with challenging attribute problems such as occlusion, deformation, and motion blur, thus achieving high tracking precision, speed, and robustness. For example, two object tracking algorithms based on the actor-critic framework are proposed in Refs. [16, 17]. The

actor network outputs actions according to the state of the input image, and the critic network evaluates this action. However, the reward function designed by the algorithm is somehow rough, which is not enough to guide the output of good actions, thus reduces the overall tracking performance of the algorithm to a certain extent. Moreover, the performance of the critic network has a crucial impact on the convergence and stability of the actor-critic framework. During the training process, the critic network may be affected by oscillations, local optima, and other factors, leading to inaccurate evaluation of the value of states. These result in inaccurate feedback to the actor network, causing unstable behavioral policies and ultimately adversely affecting the performance of the entire framework, leading to issues such as unstable training and non-convergence.

According to the above analysis, we propose an improved DRL object tracking algorithm based on actor-double critic network to solve the challenging problems of object occlusion, deformation, and object rotation in video object tracking. The main contributions are as follows:

(1) An improved DRL object tracking algorithm based on actor-double critic network is proposed. During offline training, the designed double critic network is used to average its output Q value to effectively guide the actor network to improve the tracking strategy. Compared with the single critic network, the proposed double critic network can resist the poor performance of one of the critic networks, and effectively improve the convergence and stability of actor-critic method. Experiments show that the accuracy and success rate of tracking algorithm are

1 School of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, China

2 School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China

Address correspondence to Xinhong Hei, heixinhong@xaut.edu.cn

© The author(s) 2023. The articles published in this open access journal are distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>).

significantly improved in challenging scenes such as object occlusion.

(2) An effective reward function is designed. In online tracking, the designed reward function is used to evaluate the action output of the actor network. This reward function can greatly improve the output probability of positive actions, reduce useless actions, and avoid the instability of action output. Finally, the strategy can continuously locate and track the object when it turns over and deforms.

The organizational structure of this paper is as follows. Section 1 introduces the related works of deep learning object tracking algorithm and deep reinforcement learning object tracking algorithm. Section 2 introduces the principle of improved DRL object tracking algorithm based on actor-double critic network, which is divided into two parts: offline training and online tracking. Offline training includes parameters setting of DRL and the specific training process of actor-double critic network. Section 3 gives the experimental results and analysis, including quantitative analysis and qualitative analysis. Finally, Section 4 summarizes the work of this paper and points out the future work.

1 Related Work

According to the different requirements for tracking precision and speed in object tracking tasks, the current object algorithms can be divided into two categories: (1) traditional and DL object tracking and (2) DRL object tracking.

1.1 Traditional and DL object tracking

The traditional tracking method^[18, 19] has a fast tracking speed. Henriques et al.^[20] proposed KCF algorithm of kernel correlation filter, and Choi et al.^[21, 22] proposed an integrated tracking system with attention mechanism to deal with various types of correlation filters. However, it cannot complete the tracking task well for its insufficient feature representation ability. Some researchers have added deep features to correlation filters for tracking. Wang et al.^[23] constructed G network and S network on the basis of the feature maps of different layers of convolutional neural network (CNN). G network was used to distinguish the object category, and S network was used as the binary classification of the object and background. When updating online, the parameters of S network are only fine-tuned every 20 frames. This algorithm can achieve good tracking performance on a relatively simple model, but its tracking speed will decrease for complex networks. Danelljan et al.^[24] proposed a discriminant correlation filter that integrates multi resolution depth features for efficient tracking. The algorithm of adding depth features to the correlation filter is still slow because it takes a lot of time to train the filter and calculate the depth features. Xia et al.^[11] proposed a depth tracking algorithm MDNet based on multi-domain network. This algorithm uses the general feature layer to obtain the general information representation of the input image, and uses the specific feature layer to realize the binary classification of the object and background, which can finally achieve high tracking precision, but the tracking speed is only 1 frame per second (fps). In 2017, Fan and Ling^[25] proposed structure-aware network (SANet) algorithm. This algorithm combines the features of CNN and recurrent neural network (RNN) to realize tracking. The former classifies the object and background, while the latter distinguishes the object and object similarity. The perfect combination of the two in the tracking process can effectively improve the anti-jamming ability of the object. Although the algorithm is based on the improvement of MDNet, the speed is

still 1 fps. Based on the online tracking idea of bagging integration, Han et al.^[26] proposed BranchOut method, which is different from MDNet. This method consists of three conv layers and multiple fc layers. During tracking, branch subsets are randomly selected according to Bernoulli distribution to obtain tracking results, and finally the network can achieve better tracking performance without pre-training. Chi et al.^[27] proposed DNT, a tracking algorithm based on dual deep network for visual tracking. This algorithm makes full use of the features of different layers of convolutional neural network to achieve the object tracking task, but its running speed is low (5 fps).

Valmadre et al.^[28] added correlation filtering (CF) into the framework of deep learning network, and combined the effective learning ability of CF online stage with the feature discrimination ability of CNN offline stage for end-to-end training, so as to expect fast and high precision tracking performance. Jung et al.^[29] proposed RT-MDNet, a fast version of tracking algorithm MDNet, which makes feature extraction faster. By introducing a new loss function into the multi-domain network to learn a more discriminating classification model, it finally achieved a speed about 20 times faster than MDNet. Voigtlaender et al.^[30] proposed a Siam R-CNN algorithm for tracking through re-detection. This algorithm matched the proposed region with the template region to re-detect the object location in the image, and regressed the boundary box of the object. At the same time, a new trajectory-based dynamic programming method and a new difficult sample mining strategy were designed to jointly improve the robustness of Siam R-CNN to similar objects. When the backbone network is ResNet-101, the speed of this algorithm is 4.7 fps (up to 15 fps on ResNet-50). Yang et al.^[31] proposed a tracking model ROAM consisting of two parts: response generator with adjustable size and bounding box regression. Both parts use convolution filters with adjustable size to adapt to the shape change of the object, thus reducing the tracking model parameters. At the same time, a cyclic neural optimizer is trained to update the tracking model parameters to make the final tracking performance more robust. The speed of the tracking model with GPU acceleration is 13 fps.

To sum up, the object tracking algorithm based on DL uses CNNs to extract the richer appearance features of the object, and then uses the appropriate classifier to recognize the object from the background and realize tracking, which greatly improves the tracking precision^[11, 17, 24, 25, 32, 33].

1.2 DRL object tracking

Although the tracking precision of the DL object tracking algorithm is generally good, the tracking speed will decrease due to the complexity of updating the object template online, which is difficult to meet the speed requirements in actual tracking tasks. Therefore, a large number of scholars consider combining RL and DL to achieve tracking to improve the shortcomings of DL object tracking. DL object tracking algorithm is divided into offline training and online tracking. Offline training is divided into network construction, RL parameter setting (state, action, reward function), and network model training. Online tracking receives the offline trained network model and updates the network parameters at the same time. Among them, the design of reward function and the strategy of updating network parameters are the key factors that affect the performance of DRL object tracking algorithm.

When using RL to train the tracking network, according to the actions output by the network and the reward function formula jointly defined by the tracking result and the real bounding box, the reward value of its actions is given correspondingly, which is

used to evaluate the actions. Yun et al.^[17] put forward the tracking algorithm of action-decision network (ADNet), which trains the action decision network through RL and rewards the actions output by the network to evaluate the actions, and finally get the accurate location of the object, but the output actions of this method are unstable.

Researchers proposed many kinds of method to solve the object tracking under the challenging scenario such as object occlusion, deformation, and object rotation in video object tracking. However, tracking accuracy and robustness are still insufficient. In order to solve this problem, an improved DRL object tracking algorithm based on actor-double critic network is proposed in this paper. Experimental results conducted on the visual tracker benchmark show that the tracking algorithm proposed in this paper outperforms the mainstream DL tracking algorithms and DRL tracking algorithms under the above challenging attributes.

2 Principle of Object Tracking Based on Actor-Double Critic Network

The proposed actor-double critic network tracking framework is shown in Fig. 1, which mainly consists of two parts: offline training and online tracking. Firstly, the actor-double critic network is trained offline using a large-scale dataset. The input of the actor network is a video sequence, the output is a 3-dimensional action vector, the double critic network evaluates the action and the output double Q value is averaged to guide the actor network to optimize the tracking strategy. The adaptive moment estimation method is used to update the Actor network's parameters. Then the well-trained AC network model is used in

online tracking. Actor network provides a dynamic search strategy to effectively locate the tracked object, while critic network evaluates the three-dimensional action vector. According to the value function, whether the boundary box of the object is accurate or not is determined, so as to achieve more robust tracking. This section is divided into two subsections, in which Section 2.1 introduces the offline training and Section 2.2 introduces the online object tracking.

2.1 Actor-double critic network offline training

This subsection will describe the offline training of the network in detail, including the setting of reinforcement learning parameters and the specific training process of actor-double critic network.

2.1.1 Parameter setting of RL

Object tracking process can be regarded as a sequential decision-making problem, so, the tracker based on RL also follows the Markov decision processes (MDP). The basic components of MDP include state s , action a , state transition function s' , and reward r . The tracker can be regarded as an agent to judge the specific location of the object to be tracked in each frame during the tracking process. The specific setting of $\{s, a, s', r\}$ are as follows.

State s : State s can be defined as the image patch within the bounding box $b = \{x, y, w, h\}$, where (x, y) , w , and h represents the center, width, and height of the bounding box, respectively. The preprocessing function $\phi = (b, F)$ is used to crop the image patch in the bounding box b in the F -th frame.

Action a : $A = \{\Delta x, \Delta y, \Delta \gamma\}$, Δx and Δy represent horizontal displacement and vertical displacement, respectively, and $\Delta \gamma$

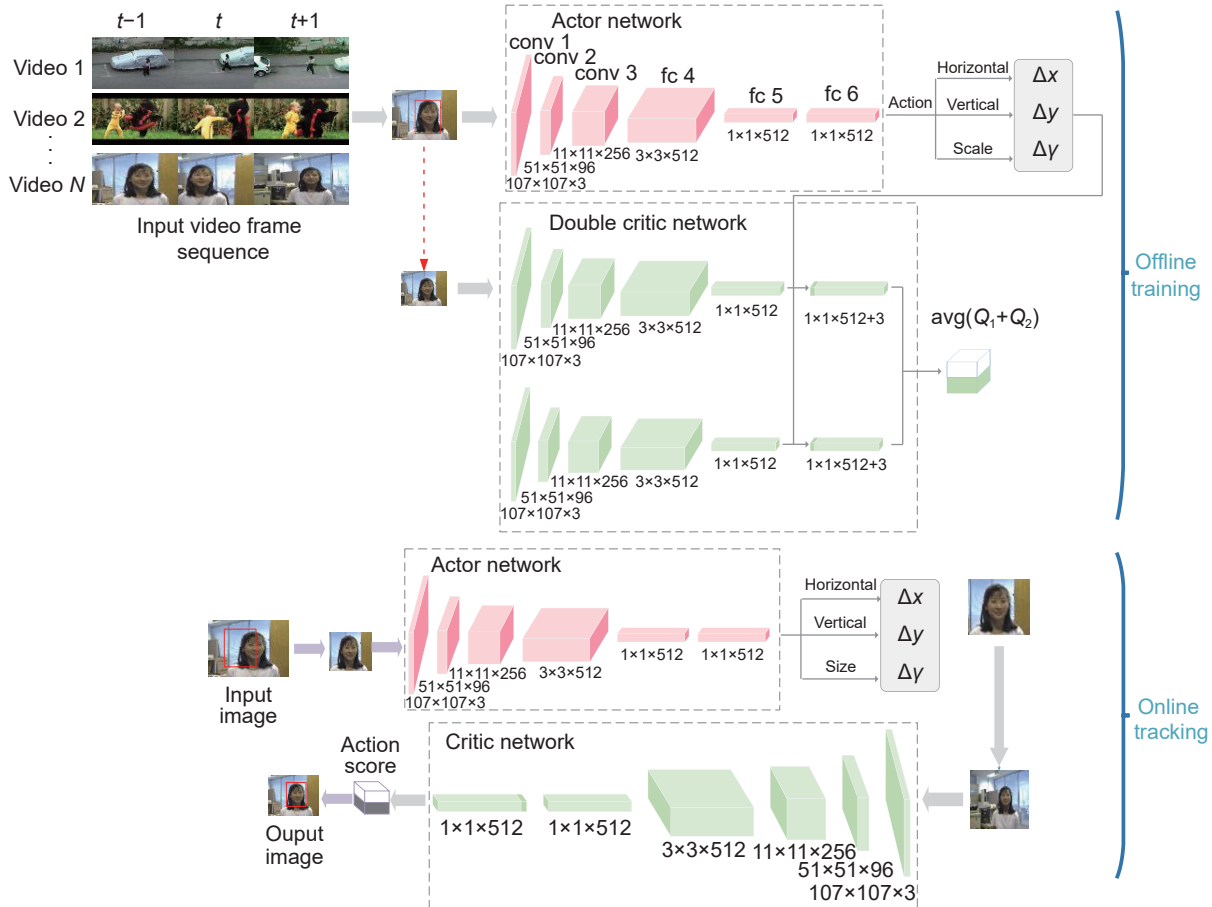


Fig. 1 Overall block diagram of deep reinforcement learning object tracking algorithm based on actor-double critic network.

represents scale change.

State transition function s' : $s' = f(s, a)$, by combining the initial boundary box with the action transition amount, a new boundary box can be obtained as $b' = \{x', y', w', h'\}$, and the calculation of the action transition amount is

$$\begin{cases} x' = x + \Delta x \times w \\ y' = y + \Delta y \times h \\ w' = \Delta w \times w \\ h' = \Delta h \times h \end{cases} \quad (1)$$

where Δx and Δy can be obtained from Laplace distribution with mean value of 0, and Δw and Δh can be obtained from Laplace distribution with mean value of 1.

Reward r : The agent obtains the reward from the state s , which is independent of the action and is only defined at the final location of each frame. In order to greatly improve the output probability of positive actions, reduce useless actions, and avoid the instability of action output, a reward function as shown in Formula (2) is designed in this paper.

$$r(s) = \begin{cases} 1 + IoU, & \text{if } IoU(b', G) \geq 0.7; \\ 1, & \text{if } 0.5 \leq IoU(b', G) < 0.7; \\ -1, & \text{otherwise} \end{cases} \quad (2)$$

If the overlap ratio of the tracking results is not less than 0.7, it means that the object is being successfully tracked, and the reward is set as $1 + IoU$ (IoU represents the overlap score of the tracked bounding box b' and the ground truth bounding box G), and if it is less than 0.5, it means that the tracking fails, the reward is set as -1 . In order to make the tracking process more robust and reduce useless actions, when the overlap ratio of tracking results meets $0.5 \leq IoU(b', G) < 0.7$, it means that the tracking performance of the tracker is stable at this time, and the reward is set as 1.

2.1.2 Training process of actor-double critic network

The offline training block diagram is shown in Fig. 2. Actor-double critic networks are based on VGG-19 model with three convolution layers conv 1, conv 2, conv 3, 3 fully connected layers conv 1, conv 2, conv 3, and 3 fully connected layers fc 4, fc 5, and

fc 6, respectively. The full connection layers fc 4 and fc 5 of actor network have 512 outputs respectively, and then combined with the ReLU activation function, the last full connection layer fc 6 generates action vector $(\Delta x, \Delta y, \Delta w, \Delta h)$. The structure of the double critic network is similar to that of the actor network. The input of the double critic network is the action vector $(\Delta x, \Delta y, \Delta w, \Delta h)$, which is the output of the actor network. During offline training, the average value of Q corresponding to the action is predicted according to the current state, so as to obtain a more accurate Q value. Finally, this average value Q can guide the actor and the critic network model to jointly learn. This improvement improves the stability and convergence of actor-double critic network, and makes the tracking network achieve higher success rate and accuracy in challenging scenarios. Deep deterministic policy gradient (DDPG) algorithm is used to iteratively train actor and double critic network. The specific steps are as follows:

Step 1. Initialize actor network $\mu(s|\theta^\mu)$ and critic network $Q(s, a)$. Initialize target networks μ' and Q' . Randomly select the input video frames, get the initial State s_1 , and start the training.

Step 2. When the t -th frame arrives, use the state s_{t-1} of the previous frame to get the current state s_t , select and extract action $a_t = \mu(s_t|\theta^\mu)$ according to Eq. (3), compute reward r_t according to Formula (2), and then get the next state s'_t , then store the generated state-action pair (s_t, a_t, r_t, s'_t) in the experience playback pool R .

Step 3. In the training process, the learning of actor network depends heavily on the evaluation results of critic network, which reduces the stability and convergence of the algorithm. For this reason, we propose the algorithm architecture of double critic network. The proposed algorithm uses the double critic network to obtain the Q_1 and Q_2 values respectively during offline training, and then take the average of Q_{y_1} and Q_{y_2} to get the best output predicted value y_t . For challenging scenarios, even when one critic network does not perform well, the network structure of double critic can weaken this influence to some extent. N state-action pairs are sampled from R , and then the mean square error is used to minimize the error between y_t , Q_{y_1} , and Q_{y_2} respectively, so as to update the parameters of the double critic network, that is the critic network parameters. The equations are shown in Eqs. (3) and (4). The way of updating actor network parameters is

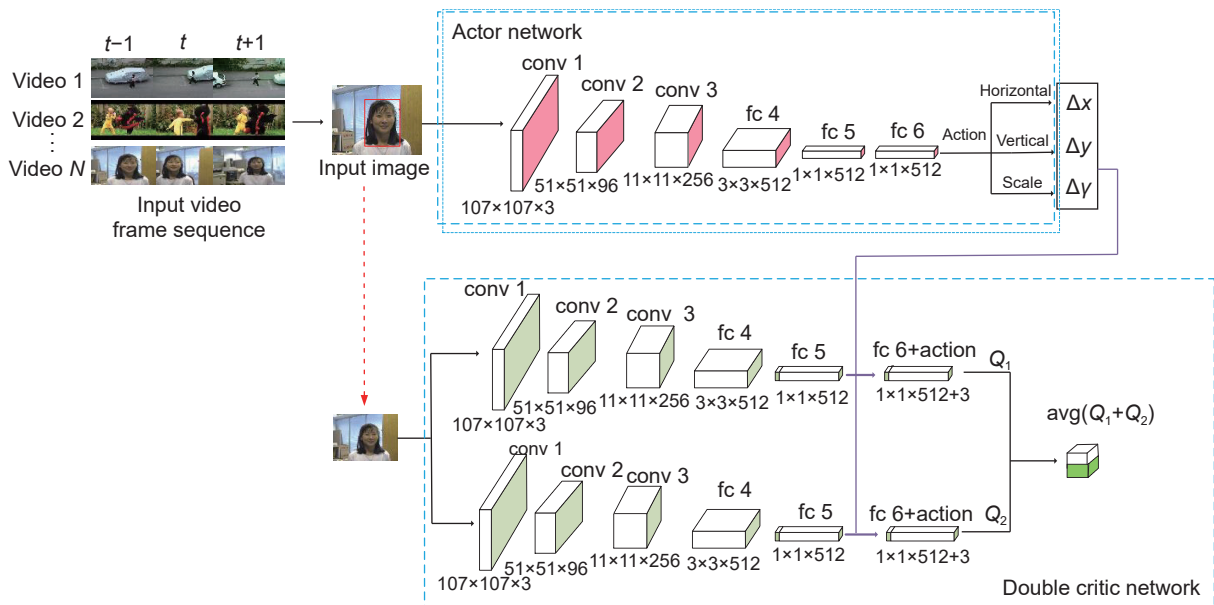


Fig. 2 Offline training of deep reinforcement learning object tracking based on actor-double critic network.

determined by the gradient direction of critic network updating parameters. The action generated by actor network is evaluated by the value functions Q_1 and Q_2 , and the higher the Q value, the better the action taken. Therefore, the chain rule (Eq. (5)) is adopted to update the actor network parameter μ .

$$L_1 = \frac{1}{N} \sum_i (y_i - Q_1(s_i, a_i | \theta^{Q_1}))^2 \quad (3)$$

$$L_2 = \frac{1}{N} \sum_i (y_i - Q_2(s_i, a_i | \theta^{Q_2}))^2 \quad (4)$$

$$\nabla_{\theta_\mu} J = \frac{1}{N} \sum_i \nabla_a Q(s, a | \theta^Q) \Big|_{s=s_i, a=\mu(s_i)} \nabla_{\theta_\mu} \mu(s | \theta^\mu) \Big|_{s=s_i} \quad (5)$$

where $y_i = r_i + \gamma Q'(s'_i, \mu'(s'_i | \theta^{\mu'}) | \theta^{Q'})$ is the predicted Q value of the output, and i is the i -th frame in the video sequence. J is the expected return from the start distribution of the “actor” model. By applying the chain rule, it will be updated by the model parameters $\nabla_{\theta_\mu} J$, where $\nabla_a Q(s_i, a | \theta^Q) | a = \mu(s_i)$ represents the gradient of critic network parameters and $\nabla_{\theta_\mu} \mu(s | \theta^\mu) | s = s'_i$ represents the gradient of actor network parameters.

Step 4. Use Formula (6) to update the target network μ' and Q' .

$$\begin{aligned} \theta^{Q'} &\leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'} \\ \theta^{\mu'} &\leftarrow \tau \mu^Q + (1 - \tau) \mu^{\mu'} \end{aligned} \quad (6)$$

Repeat the above steps until the reward output is stable.

2.2 Online tracking

After the offline training of the network, the trained actor-critic (AC) network model is further used in online tracking. In the current frame, critic network, as an evaluation network, evaluates the action generated by actor network, and obtains its corresponding score according to the reward Formula (2). The score can be used to evaluate the goodness of the action for more robust tracking.

Specifically, in the $(t-1)$ -th frame, the pre-processing function $\phi(b'_{t-1}, F_t)$ is first used to calculate the state s_t , where b'_{t-1} represents the optimal object location in the $(t-1)$ frame and F_t is the image in the t -th frame. Then, the state s_t is used as the input of the actor network, and the actor network takes action a_t according to $\mu(s_t | \theta^\mu)$ and b'_{t-1} to get the new location b'_t and its state s'_t in the current frame. Then critic network $v(s'_t | \theta^v)$ is used to evaluate the action of the state. If the score calculated by the

network according to Formula (2) is not less than 0.7, it means that the object is being tracked, and the reward at this time is set as $1+IoU$. In order to make the tracking process more robust and avoid repeated actions, when the overlapping rate of tracking results is between 0.5 and 0.7, it means that the tracking performance of the tracker is stable, and the reward is set to 1 at this time. If it is less than 0.5, it means that the tracking fails, and the reward is set as -1 . At this time, an effective online update strategy is adopted, which is shown in Fig. 3. Critic network is used to evaluate the N samples around the b'_{t-1} , which b'_{t-1} represents the optimal object location in the $(t-1)$ -th frame. The sample b'_t with the highest score evaluated by the critic network is used as the optimal location.

3 Experiment

In order to verify the effectiveness of the proposed improved DLR object tracking algorithm, we have conducted several groups of comparative experiments with 4 mainstream object tracking algorithms on the standard tracking data set OTB100. The four algorithms are ACT^[16], ADNet^[17], CFNet^[28], and RT-MDNet^[29], among which the former two are DLR algorithms and the latter two are DL algorithms. The experimental results of overall tracking and tracking based on 7 challenge attributes show that the tracking precision and speed of the proposed algorithm are better than the mainstream tracking algorithms. The detailed experimental results and analysis are as follows.

3.1 Experimental platform and parameter setting

Under Ubuntu 14.04 system, NVIDIA GeForce GTX 980Ti GPU is used to accelerate its offline training and online testing process. The model training and testing framework built in the experiment is a DL framework based on Pytorch (version 1.4.0), written by Python (version 3.6.2). CUDA and CUDNN versions are CUDA 8.0 and CUDNN 7.5, respectively. The training dataset used in the experiment is ImageNet-ILSVRD2015^[34] large scale video image dataset, and the test dataset is OTB100^[35]. When pre-training actor networks and critic networks offline, 20–40 image frames are randomly selected from a video sequence in this dataset during each iterative training, and the learning rates of the two networks are set to 10^{-6} and 10^{-5} , respectively. Firstly, the actor network is initialized, that is, 72 positive samples are collected around the first frame image, and the overlap ratio IoU of these positive samples relative to the object real frame is all greater than 0.7. The learning rate of the network in this initialization stage is 10^{-4} . During offline training, actor network adopts greedy strategy search method when selecting actions. The initial value is set to $\varepsilon = 0.5$, and it

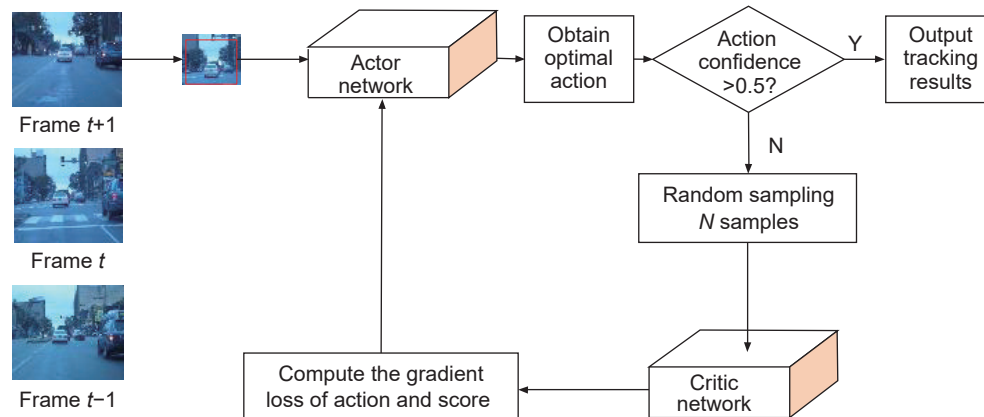


Fig. 3 Online update strategy.

decreases by 0.05 after every 10 000 iterations of the network.

3.2 Dataset

The ImageNet-ILSVRC2015 dataset includes long-time tracking video sequences of various scenes. The OTB100 dataset includes OTB100^[35] and OTB50^[36]. Both datasets have 11 challenging attributes, such as illumination change, out-of-plane rotation, deformation, occlusion, fast motion, motion blur, background clutter, in-plane rotation, out-of-view, and low resolution. In the experiment, ImageNet-ILSVRC2015 dataset is used for training and OTB100 is used for testing.

3.3 Experimental result and analysis

3.3.1 Quantitative evaluation

The one pass evaluation (OPE) criterion of precision and success rate is adopted as the quantitative evaluation index. In the experiment, the location error threshold range of the precision plots is set to [0, 50]. The threshold range of the success rate curve is set to [0, 1]. The precision plots and success rate plots for overall performance of five tracking algorithms on OTB100 are shown in Fig. 4. And Fig. 5 is the performance plots based on seven challenging attributes: illumination change (IV), out of plane rotation (OPR), occlusion (OCC), deformation (DEF), in plane rotation (IPR), fast motion (FM), and out of the view (OV).

It can be seen from Figs. 4 and 5, compared with the other four mainstream tracking algorithms, the proposed tracking algorithm (Ours) has a better tracking performance whether based on the overall or seven attributes. The overall precision and success rate of Ours rank first among the five algorithms, with 92.2% and 81.4% respectively. Especially, the precision and success rate scores of Ours are the first in OCC, OPR, and IPR, and the precision scores of IV, DEF, and FM are the second, which is higher than ACT, RT-MDNet, and CFNet. Although the success rate score of the proposed algorithm in FM and OV attributes is not higher than that of ADNet and RT-MDNet, the success rate score of Ours is higher than that of the baseline ACT, which shows the effectiveness of the improved algorithm strategy. In addition to the precision plots and the success rate plots, we can also use the 20-pixel error threshold in the precision plots and the area under curve (AUC) in the success rate plots to draw the score histogram of the tracking results of the five tracking algorithms. The scores of the overall and seven attributes performance of IV, OPR, OCC, DEF, FM, IPR, and OV performance of the 5 tracking algorithms are shown in Fig. 6.

In Table 1, Y represents GPU acceleration. According to the

comparison of the tracking speed of the five tracking algorithms in Table 1, the speed of Ours is 25 fps, which is faster than the DRL tracking algorithms ADNet and ACT, and is close to the speed of the DL tracking algorithm RT-MDNet. Although the speed is not faster than that of CFNet tracking algorithm based on correlation filtering and DL, the precision and success rate of Ours are always higher than that of CFNet. At the same time, although the speed of our algorithm is equal to that of the DL tracking algorithm RT-MDNet, the overall precision and success rate of our algorithm are indeed higher than that of RT-MDNet. To sum up, the proposed tracking algorithm achieves a good balance between tracking precision and tracking speed, and it achieves a satisfactory tracking speed on the premise of ensuring the precision. If it is applied to practice, it can also accomplish the tracking task well in the face of complex environment.

In order to comprehensively evaluate the performance of the proposed tracking algorithm, the performance of Ours and two mainstream DRL tracking algorithms of ADNet and ACT, were tested and evaluated on 6 video sequences: Basketball, BlurFace, Football, Jump, Skater2, and Subway from two aspects of center location error and overlap rate. The test video sequence is selected from the OTB100, and its attributes are shown in Table 2, IV, OPR, SV, OCC, DEF, MB, FM, OV, BC, and LR in the table have been described above. The test and evaluation results of the three algorithms are shown in Fig. 7.

As shown in Fig. 7, the red plots, green plots, and blue plots, in the figure represent the center error and overlap rate of Ours, ADNet, and ACT, respectively. The goal we expect to achieve is that the smaller the pixel error value on any sequence, the better, and the larger the overlap rate, the better. In the video sequence basketball, near the 10th frame and the 480th frame, the pixel error plots of ADNet suddenly increase, while the overlap ratio suddenly decrease, all of which have peaks, local instability, and large errors, while the average errors of Ours and ACT are small. In the video sequence BlurFace, after 120 frames, ADNet and ACT have been unstable, with large deviation. However, Ours is stable from the beginning to the end of the frame, without big drift. In the video sequence football, the pixel error plots of ADNet suddenly increases in the 60th frame, and then changes in the form of triangular wave, which is very unstable. This is mainly due to the serious interference of the background similar to the object and the serious occlusion of the object. However, the pixel error plots of Ours and ACT always tend to be stable, and the overlap ratio plots of ACT are slightly unstable around 160 and 330 frames. In the video sequence Skater2, the pixel error of ADNet is larger at the 10th and 110th frames, and the error of Ours and

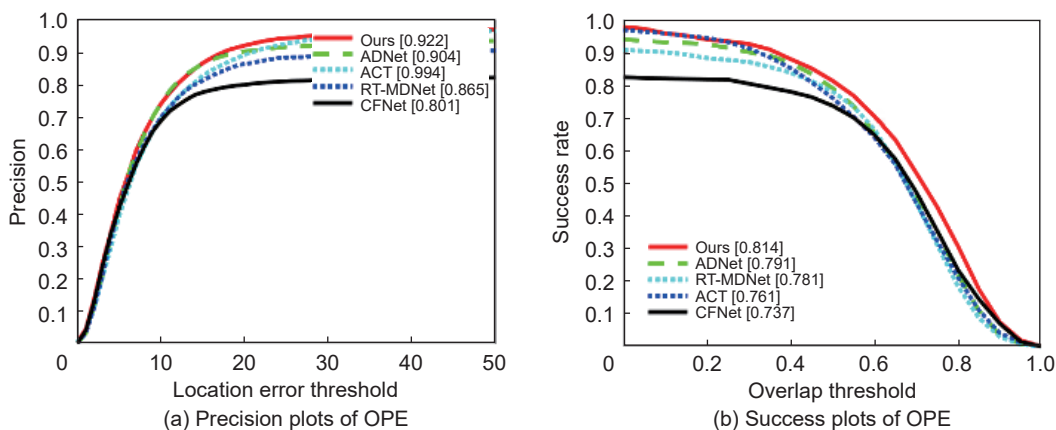
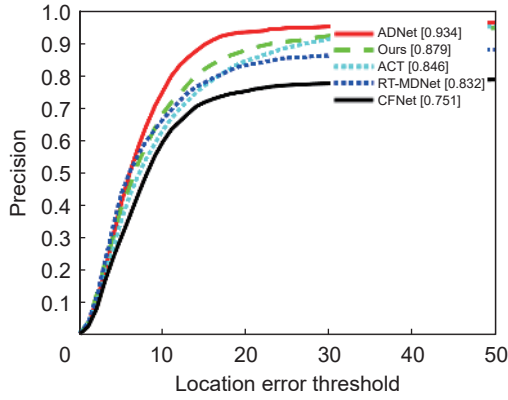


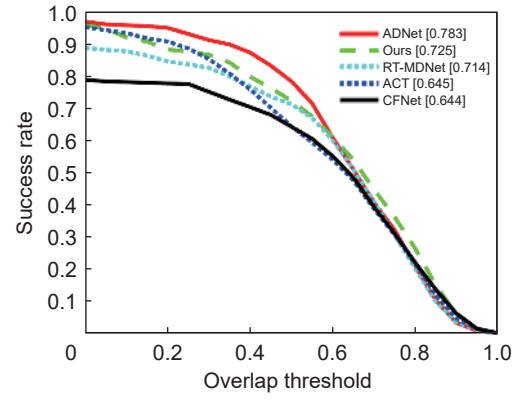
Fig. 4 Precision plots and success rate plots for overall performance.

ACT is larger at the 205th frame. In the overlap plots, ADNet and ACT are basically in an unstable state all the time, and the peak of Ours only appears at the 205th frame. In the 40th frame of the

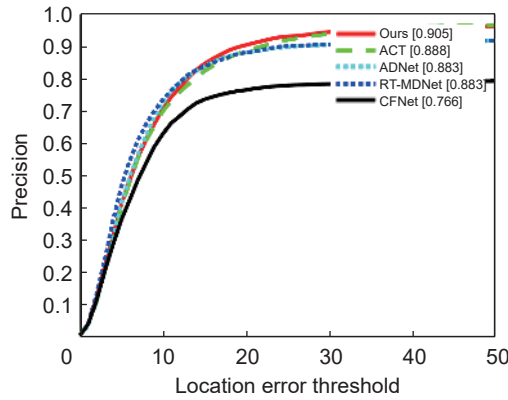
video sequence subway, all three algorithms have peaks, among which ADNet has the highest peak, so the error is the largest. The main reason is that the object is suddenly completely blocked,



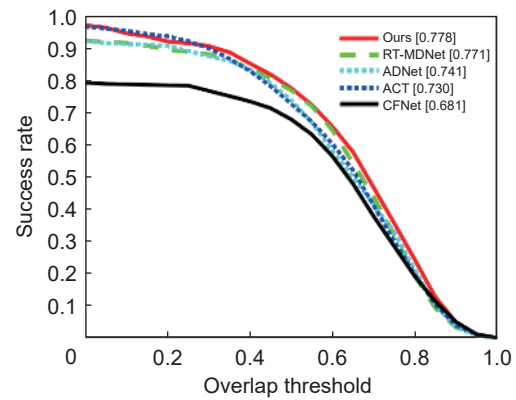
(a) Precision plots of OPE-IV (10)



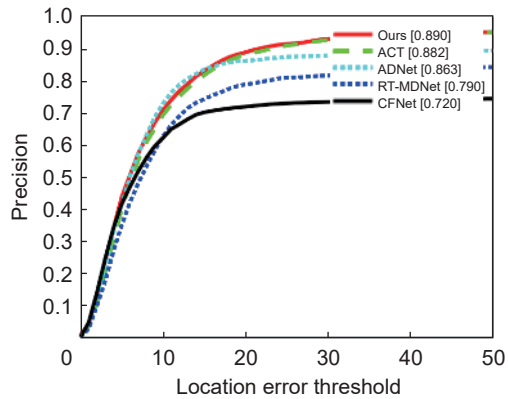
(b) Success plots of OPE-IV (10)



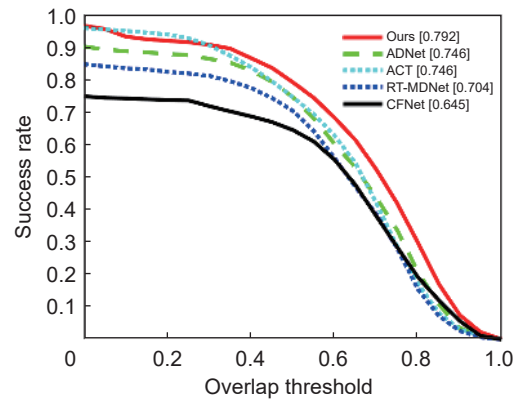
(c) Precision plots of OPE-OPR (15)



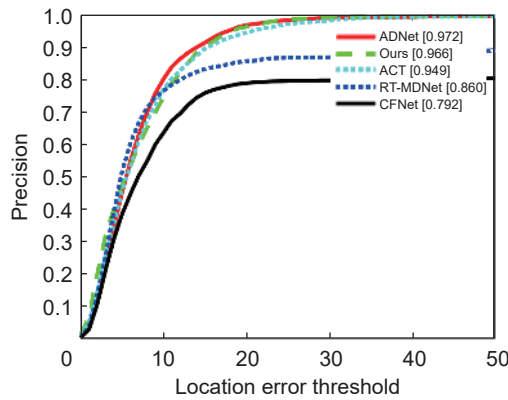
(d) Success plots of OPE-OPR (15)



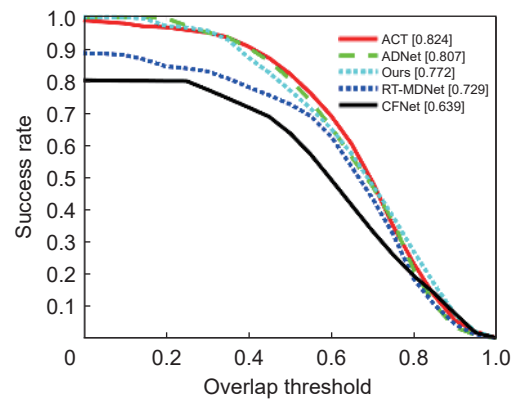
(e) Precision plots of OPE-OCC (12)



(f) Success plots of OPE-OCC (12)



(g) Precision plots of OPE-DEF (8)



(h) Success plots of OPE-DEF (8)

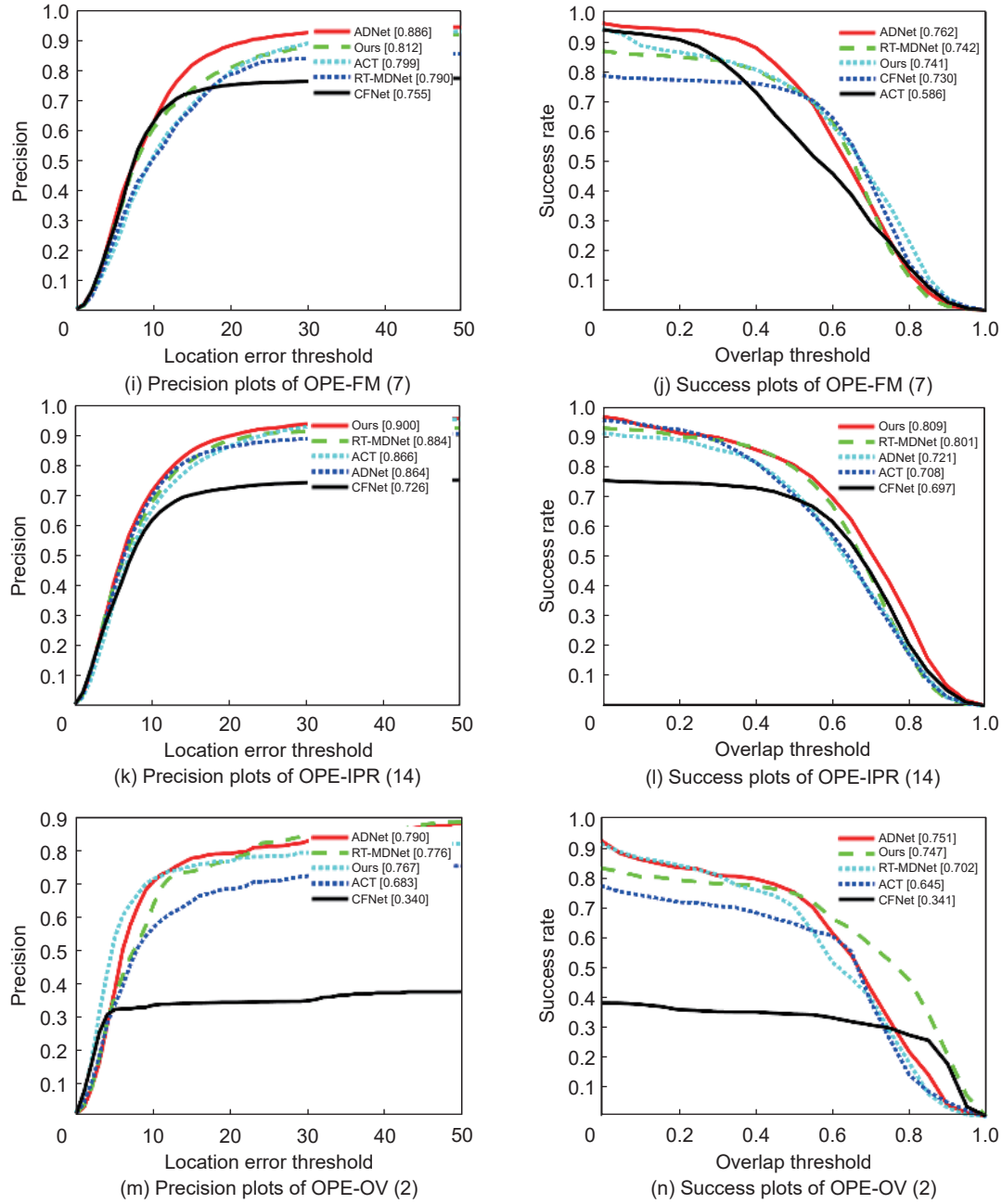


Fig. 5 Precision plots and success rate plots for 7 challenge attributes performance.

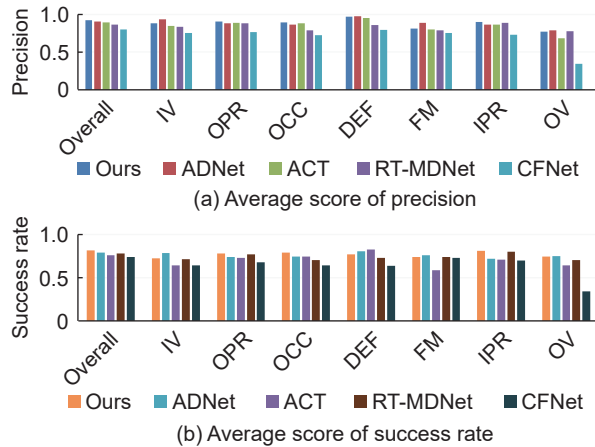


Fig. 6 Histogram of average score of precision (20 pixels) and success rate based on the overall performance and 7 attributes.

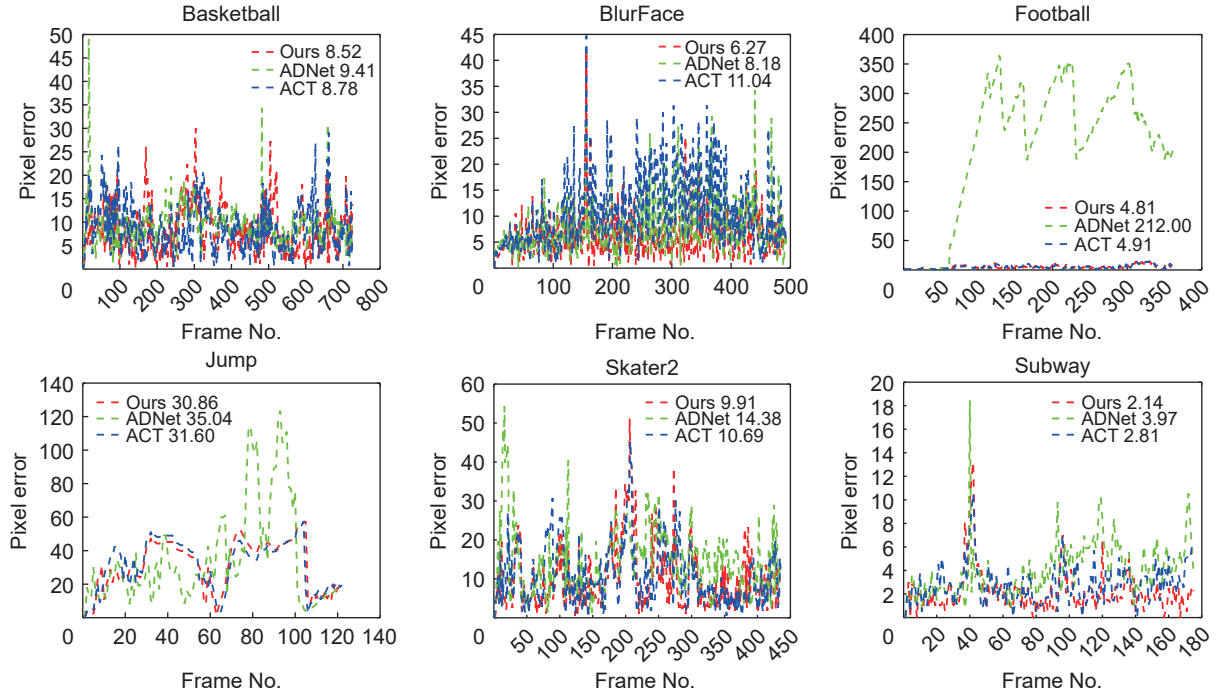
Table 1 Speed comparison of 5 algorithms.

Method	Speed (fps)	GPU
Ours	25	Y
ADNet ^[31]	3	Y
ACT ^[16]	18	Y
RT-MDNet ^[28]	26	Y
CFNet ^[27]	43	Y

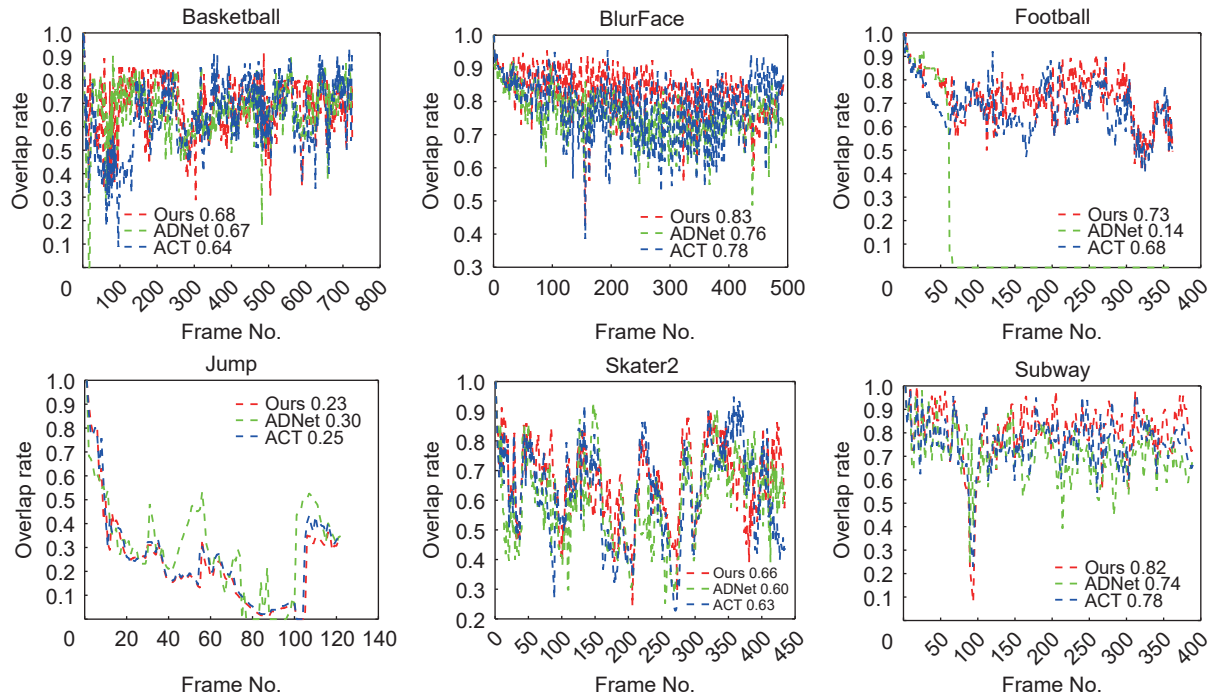
which leads to local drift in the tracking of all three algorithms. Generally speaking, Our method has a better tracking effect than the other two algorithms, and it shows a relatively small error. The ACT ranks second, and the tracking performance of ADNet is poor, which shows the precision and stability of our algorithm. Table 3 shows the pixel error and overlap rate values of three algorithms on 6 video sequences.

Table 2 Attributes of 6 test video sequences.

Sequence	Number of Frames	Image pixel size	IV	OPR	SV	OCC	DEF	MB	FM	IPR	OV	BC	LR
Basketball	725	576×432	√	√		√	√					√	
BlurFace	493	640×480						√	√	√			
Football	362	640×480		√		√				√		√	
Jump	122	416×234		√	√	√	√	√	√	√			
Skater2	435	320×240		√	√		√		√	√			
Subway	175	360×240				√	√					√	



(a) Plots of center distance error



(b) Plots of overlap rate

Fig. 7 Plots of center distance error (pixel error) and overlap rate on the 6 sequences.

Table 3 Pixel error and overlap rate of 3 algorithms on 6 video sequences.

Sequence	Ours		ADNet ^[31]		ACT ^[16]	
	Pixel error	Overlap rate	Pixel error	Overlap rate	Pixel error	Overlap rate
Basketball	8.52	0.68	9.41	0.67	8.78	0.64
BlurFace	6.27	0.83	8.18	0.76	11.04	0.78
Football	4.81	0.73	212.0	0.14	4.91	0.68
Jump	30.86	0.23	35.04	0.30	31.60	0.25
Skater2	9.91	0.66	14.38	0.60	10.69	0.63
Subway	2.14	0.83	3.97	0.74	2.81	0.78

In order to verify the effectiveness of the improved algorithm proposed in this paper, this section does two groups of ablation experiments on video sequences with specific attributes with the improved DRL algorithm and ACT.

In the first group of experiments, in order to verify that the improved double critic network (ACT_Double_C, Our method) can better deal with the occlusion problem than the single critic network (ACT_C), the precision and success rate of sequence with occlusion attributes are compared. The experimental results are shown in Fig. 8 and Table 4.

It can be seen from Fig. 8 and Table 4 that compared with ACT_C, the precision of ACT_Double_C is increased by 5.4% and the success rate is increased by 8.2%, which shows that the improved double critic network proposed in this paper can improve the tracking performance of the algorithm when the object is blocked better than the single critic network.

In the second group of experiments, in order to verify that the improved three-stage reward function (ACT_Tri_R, Our method) can better deal with the deformation and rotation of the object than the two-stage reward function (ACT_Sec_R), the precision and success rate of the sequence with deformation and rotation were compared. The experimental results are shown in Fig. 9 and Table 5.

As can be seen from Fig. 9 and Table 5, it is concluded that compared with ACT_Sec_R, the precision of ACT_Tri_R (Our method) has increased by 4.5% and the success rate has increased by 3.3%, which indicates that the improved three-stage reward function proposed in this paper can improve the tracking performance when the object is deformed and rotated better than the two-stage reward.

3.3.2 Qualitative evaluation

To further illustrate the performance of the proposed tracking algorithm, we will qualitatively evaluate the proposed algorithm and four mainstream object tracking algorithms on 6 challenging sequences. Table 2 shows the corresponding challenging attributes of 6 video sequences (six sequences have more than four challenging attributes such as occlusion and background clutter), and the partial tracking results of video sequences are shown in Fig. 10.

As shown in Fig. 10, the red, green, blue, black, and purple rectangular bounding boxes represent the tracking effects of Ours, ADNet, ACT, RT-MDNet, and CFNet, respectively. It can be seen from the tracking results in Fig. 10 that the tracking effect of Ours on six sequences is good. Specific ally, in the 16th frame of the basketball sequence, the object is blocked by more similar backgrounds, which causes the tracking frames of ACT and CFNet to drift, and then follow the wrong object. However, Ours and the other two algorithms can still track the object. In frame 631, when the object suddenly turns, RT-MDNet directly loses the object, and the tracking boundary box of ADNet also drifts a little, but Ours can still track the object accurately. In the video sequence blurFace, at the 230th, 341st, and 468th frames, the object's face is blurred due to the rapid turning of the head. At the 468th frame, the object's face changes. Compared with the other three algorithms, Ours and ADNet have better tracking performance. In the sequence Football, from the 60th frame to the end of the 74th frame, because the similar background is more and more messy and the object rotates, ACT moves from tracking drift to tracking failure, while the bounding box of the other four algorithms can accurately track the surrounding object, and Ours has the highest tracking robustness. From the 35th frame to the 95th frame in Jump, Ours, ADNet, and ACT can locate the object location, while ACT, RTMDNet, and CFNet have completely

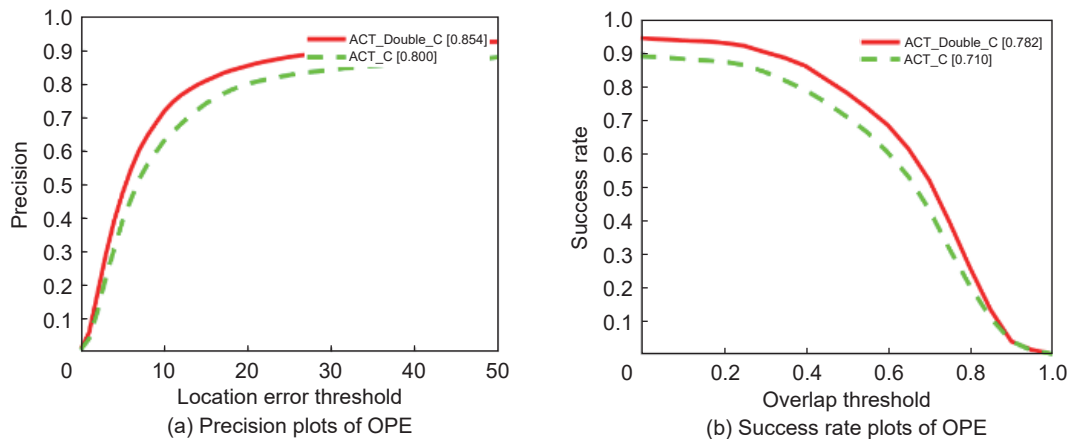


Fig. 8 Precision and success rate curve (occlusion sequence).

Table 4 Comparison of precision and success rate between improved double critic network.

Method	Precision	Success rate
ACT_C	0.800	0.710
ACT_Double_C (Ours)	0.854	0.782

Table 5 Comparison of precision and success rate between ACT_Sec_R and ACT_Tri_R.

Method	Precision	Success rate
ACT_Sec_R	0.768	0.637
ACT_Tri_R (Ours)	0.813	0.670

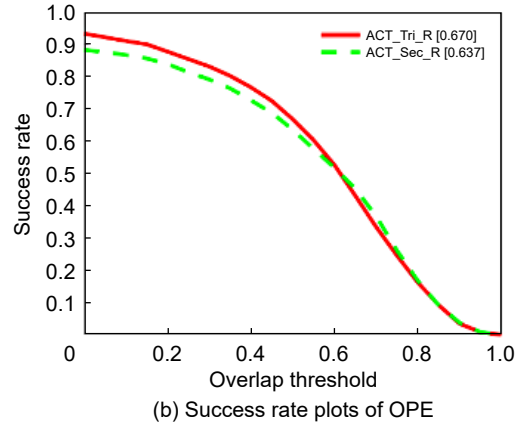
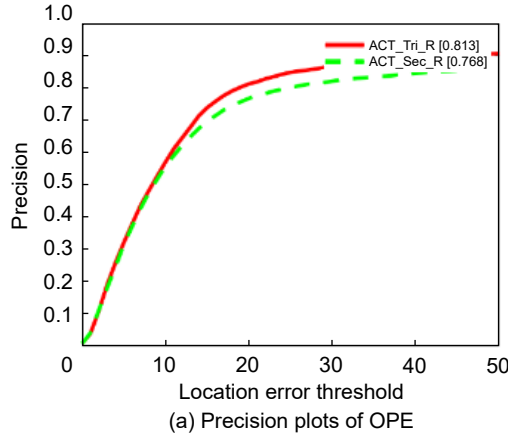


Fig. 9 Curve of precision and success rate (deformation and rotation sequence).

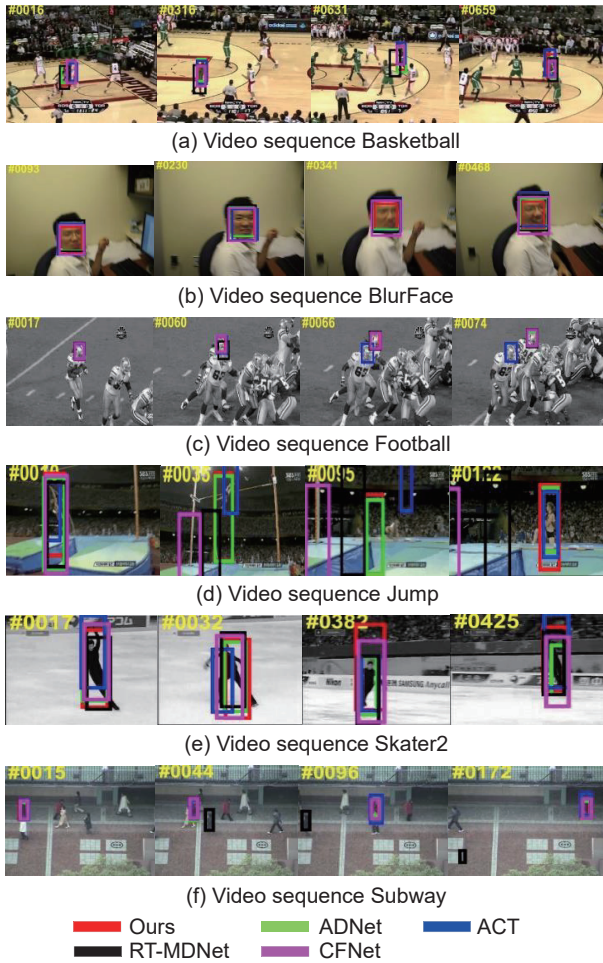


Fig. 10 Partial tracking results on the video sequences.

failed. At the 32nd frame of Skater2 sequence, the object is deformed, and the tracking boxes of RT-MDNet and CFNet drift. At the 425th frame, the object has a large jump action, where the tracking effect of ACT and CFNet is poor, while the tracking effect

of Ours and ADNet is better, and the tracking boundary box of Ours is the most accurate and robust. In the 44th frame of the video sequence Subway, when the object is completely blocked by the similar background, RT-MDNet fails to track the object directly. From the 41st frame to the 96th frame, when the object encounters the background interference again, RTMDNet still cannot track the object, and the tracking boundary box of ACT also has some tracking drift. In the last 172nd frame, except RT-MDNet, the other three tracking algorithms can track the object. In a word, the proposed algorithm has been in a stable state, and the tracking performance is the best.

To sum up, the deep reinforcement learning video object tracking algorithm based on actor-double critic network can track the object robustly under most challenging attributes.

4 Conclusion

In this paper, an improved deep reinforcement learning object tracking algorithm based on actor-double critic network is proposed. The proposed tracking network can output the continuous action for the input video sequence, and at the same time, an effective online update fine-tuning strategy is adopted to have a high tracking accuracy even when the object has a drastic change in appearance. The experimental results show that, compared with the mainstream trackers, our tracker can well deal with the problems that may lead to tracking failure in the scene of object occlusion, object deformation, rotation, and object disappearance in the field of view, and finally achieve accurate and robust tracking. In the future, we will combine the proposed tracking algorithm with unmanned aerial vehicle (UAV) control to achieve autonomous object tracking.

Acknowledgment

This work was supported in part by the National Key R&D Program of China (No. 2022YFB2602203), and in part by the National Natural Science Foundation of China (Nos. U20A20225 and 61873200) and Shaanxi Provincial Key Research and Development Program (No. 2022-GY111).

Article History

Received: 16 December 2022; Revised: 6 April 2023; Accepted: 28 April 2023

References

- [1] P. Li, X. Chen, and S. Shen, Stereo R-CNN based 3D object detection for autonomous driving, in *Proc. 2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2020, pp. 7636–7644.
- [2] D. Liu, J. Zhao, A. Xi, C. Wang, X. Huang, K. Lai, and C. Liu, Data augmentation technology driven by image style transfer in self-driving car based on end-to-end learning, *Comput. Model. Eng. Sci.*, vol. 122, no. 2, pp. 593–617, 2020.
- [3] I. Paulo Canal, M. Martín Pérez Reimbold, and M. de Campos, Ziegler–Nichols customization for quadrotor attitude control under empty and full loading conditions, *Comput. Model. Eng. Sci.*, vol. 125, no. 1, pp. 65–75, 2020.
- [4] J. Yang, S. Liu, H. Su, and Y. Tian, Driving assistance system based on data fusion of multisource sensors for autonomous unmanned ground vehicles, *Comput. Netw.*, vol. 192, p. 108053, 2021.
- [5] D. Guo, Q. Yang, Y. D. Zhang, G. Zhang, M. Zhu, and J. Yuan, Adaptive object tracking discriminate model for multi-camera panorama surveillance in airport apron, *Comput. Model. Eng. Sci.*, vol. 129, no. 1, pp. 191–205, 2021.
- [6] W. Tian, M. Lauer, and L. Chen, Online multi-object tracking using joint domain information in traffic scenarios, *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 374–384, 2020.
- [7] M. Adimoolam, S. Mohan, A. John, and G. Srivastava, A novel technique to detect and track multiple objects in dynamic video surveillance systems, *Int. J. Interact. Multimed. Artif. Intell.*, vol. 7, no. 4, p. 112, 2022.
- [8] A. A. Gumbs, I. Frigerio, G. Spolverato, R. Croner, A. Illanes, E. Chouillard, and E. Elyan, Artificial intelligence surgery: How do we get to autonomous actions in surgery? *Sensors*, vol. 21, no. 16, p. 5526, 2021.
- [9] S. Ipsen, S. Böttger, H. Schwegmann, and F. Ernst, Target tracking accuracy and latency with different 4D ultrasound systems—a robotic phantom study, *Curr. Dir. Biomed. Eng.*, vol. 6, no. 1, p. 20200038, 2020.
- [10] B. Yang, X. Liu, X. Tang, and X. Chen, AGV Multi-target tracking under smart factory, *Electron. Sci. Technol.*, vol. 32, no. 11, pp. 23–27, 2019.
- [11] Z. Xia, J. Du, J. Wang, C. Jiang, Y. Ren, G. Li, and Z. Han, Multi-agent reinforcement learning aided intelligent UAV swarm for target tracking, *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 931–945, 2022.
- [12] S. Zhang, Y. Li, and Q. Dong, Autonomous navigation of UAV in multi-obstacle environments based on a Deep Reinforcement Learning approach, *Appl. Soft Comput.*, vol. 115, p. 108194, 2022.
- [13] D. E. Asher, A. Basak, R. Fernandez, P. K. Sharma, E. G. Zaroukian, C. D. Hsu, M. R. Dorothy, T. Mahre, G. Galindo, L. Frerichs, et al., Strategic maneuver and disruption with reinforcement learning approaches for multi-agent coordination, *J. Def. Model. Simul.*, doi: 10.1177/15485129221104096.
- [14] S. Liu, J. Cao, Y. Wang, W. Chen, and Y. Liu, Self-play reinforcement learning with comprehensive critic in computer games, *Neurocomputing*, vol. 449, pp. 207–213, 2021.
- [15] S. Wen, Z. Wen, D. Zhang, H. Zhang, and T. Wang, A multi-robot path-planning algorithm for autonomous navigation using meta-reinforcement learning based on transfer learning, *Appl. Soft Comput.*, vol. 110, p. 107605, 2021.
- [16] B. Chen, D. Wang, P. Li, S. Wang, and H. Lu, Real-time ‘actor-critic’ tracking, in *Proc. 15th European Conf. Computer Vision (ECCV)*, Munich, Germany, 2018, pp. 328–345.
- [17] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, Action-decision networks for visual tracking with deep reinforcement learning, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 1349–1358.
- [18] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg, Accurate scale estimation for robust visual tracking, in *Proc. 2014 British Machine Vision Conference*, Nottingham, UK, 2014, doi: 10.5244/C.28.65.
- [19] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, Multi-Store Tracker (MUSTer): A cognitive psychology inspired approach to object tracking, in *Proc. 2015 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 749–758.
- [20] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, High-speed tracking with kernelized correlation filters, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, 2015.
- [21] J. Choi, H. J. Chang, J. Jeong, Y. Demiris, and J. Y. Choi, Visual tracking using attention-modulated disintegration and integration, in *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 4321–4330.
- [22] J. Choi, H. J. Chang, S. Yun, T. Fischer, Y. Demiris, and J. Y. Choi, Attentional correlation filter network for adaptive visual tracking, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 4828–4837.
- [23] L. Wang, W. Ouyang, X. Wang, and H. Lu, Visual tracking with fully convolutional networks, in *Proc. 2015 IEEE Int. Conf. Computer Vision (ICCV)*, Santiago, Chile, 2016, pp. 3119–3127.
- [24] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, ECO: Efficient convolution operators for tracking, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 6931–6939.
- [25] H. Fan and H. Ling, SANet: structure-aware network for visual tracking, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, USA, 2017, pp. 2217–2224.
- [26] B. Han, J. Sim, and H. Adam, BranchOut: regularization for online ensemble tracking with convolutional neural networks, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 521–530.
- [27] Z. Chi, H. Li, H. Lu, and M. H. Yang, Dual deep network for visual tracking, *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 2005–2015, 2017.
- [28] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, End-to-end representation learning for correlation filter based tracking, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 5000–5008.
- [29] I. Jung, J. Son, M. Baek, and B. Han, Real-time MDNet, in *Proc. 15th European Conf. Computer Vision (ECCV)*, Munich, Germany, 2018, pp. 89–104.
- [30] P. Voigtlaender, J. Luiten, P. H. S. Torr, and B. Leibe, Siam R-CNN: Visual tracking by re-detection, in *Proc. 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, pp. 6577–6587.
- [31] T. Yang, P. Xu, R. Hu, H. Chai, and A. B. Chan, ROAM: Recurrently optimizing tracking model, in *Proc. 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, pp. 6717–6726.
- [32] Z. Teng, J. Xing, Q. Wang, C. Lang, S. Feng, and Y. Jin, Robust object tracking based on temporal and spatial deep networks, in *Proc. 2017 IEEE Int. Conf. Computer Vision (ICCV)*, Venice, Italy, 2017, pp. 1153–1162.
- [33] C. Ma, J. B. Huang, X. Yang, and M. H. Yang, Hierarchical convolutional features for visual tracking, in *Proc. 2015 IEEE Int. Conf. Computer Vision (ICCV)*, Santiago, Chile, 2016, pp. 3074–3082.
- [34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., ImageNet large scale visual recognition challenge, *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
- [35] Y. Wu, J. Lim, and M. H. Yang, Online object tracking: A

- benchmark, in *Proc. 2013 IEEE Conf. Computer Vision and Pattern Recognition*, Portland, OR, USA, 2013, pp. 2411–2418.
- [36] Y. Wu, J. Lim, and M. H. Yang, Object tracking benchmark, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, 2015.