Review on Target Tracking Algorithm Based on Deep Learning

Shihao Tao

School of Information and Communication Engineering Nanjing Institute Of Technology Nanjing, China

x00208201127@njit.edu.cn

Abstract. Target tracking is a crucial research area in computer vision, with increasing demand for real-time monitoring and intelligent interaction. Algorithms for target tracking have evolved from traditional feature matching methods to modern deep neural networks, reinforcement learning, and model-based approaches. These algorithms are not only applied in video surveillance but also promote the development of drone tracking, autonomous driving, human-computer interaction, behavior analysis, and other fields. However, target tracking still faces challenges, such as interference from lighting intensity, environmental objects, and cluttered backgrounds, as well as the need for real-time tracking and accurate identification of multiple similar targets. This study focuses on the problem of target tracking and proposes a solution. Specifically, we adopt deep neural networks and reinforcement learning to improve the accuracy and real-time performance of target tracking. Experimental results indicate that our algorithm can accurately track targets in various scenarios and exhibits good robustness and real-time performance.

Keywords : Object tracking; Deep Learning; Deep Reinforcement Learning

1 Introduction

Object tracking has important research value and wide application prospect in the field of computer vision [1]. With the improvement of the performance of computer hardware, the development of image processing algorithms and the increasing demand for

B. H. Ahmad (ed.), *Proceedings of the 2023 International Conference on Data Science, Advanced Algorithm and Intelligent Computing (DAI 2023)*, Advances in Intelligent Systems Research 180,
https://doi.org/10.2991/978-94-6463-370-2_31

real-time monitoring and intelligent interaction, object tracking has become one of the hot spots in the field of computer vision. In recent years, with the continuous improvement of the world's scientific and technological level, target tracking algorithms have also made significant progress [2], from traditional target tracking algorithms based on feature matching to modern methods based on deep neural networks, reinforcement learning and models. These algorithms can not only be applied in video surveillance, but also play a role in promoting the development of drone tracking, automatic driving, human-computer interaction, behavior analysis and other fields.

However, there are still some challenges and difficulties in the research of target tracking algorithm. The first is the most fundamental problem in target tracking - the appearance and motion characteristics of the target in the image may be interfered with by a variety of factors, such as changes in the intensity of light, the occlusion of environmental objects, and the clutter of the background. These will cause the appearance and shape of the target to be altered by interference, thus increasing the difficulty of target tracking. Secondly, in recent years, the demand for target tracking is getting higher and higher, not only need to invest a complete video and wait for a complete result, but also have certain requirements for real-time, that is, need to be able to input video while output results. Secondly, it is necessary for target tracking to accurately distinguish multiple similar targets in the same scene, which requires the target tracking algorithm to have good differentiation ability and robustness.

The main work of this paper is to introduce the development of existing mainstream target tracking algorithms, and evaluate and compare their advantages and disadvantages. In addition, we will explore the challenges of goal tracking and future research directions to provide researchers with a comprehensive and accurate understanding. Through the research and improvement of target tracking algorithm, we can better cope with the needs of practical applications in life.

2 Target Tracking Principle

2.1 Basic process of target tracking

There is no clear and unique definition of target tracking, but many scholars have different explanations: "Tracking is the process of identifying the area of interest in the video sequence [3]", and "tracking is to estimate the state of the target in the subsequent frame through the given state of the target in a certain frame of the video [4]". In the

field of computer vision, generally speaking, the target of tracking is generally a video frame or an area or an object in the image. Target tracking is to detect the target through target detection first, and target tracking is to predict the size and position of the target in the subsequent frame under the condition of the target size and position of the initial frame of a given video sequence.

The traditional basic process of target tracking is to first enter an initial target box, generate many candidate boxes in the next frame, extract the features of these candidate boxes, and then score these candidate boxes, and finally select the candidate box with the highest score among these candidate boxes as the predicted target.

2.2 Object Detection

Object detection usually follows a common framework that covers the entire process from the input image to the final object detection result. The following is a typical object detection framework:

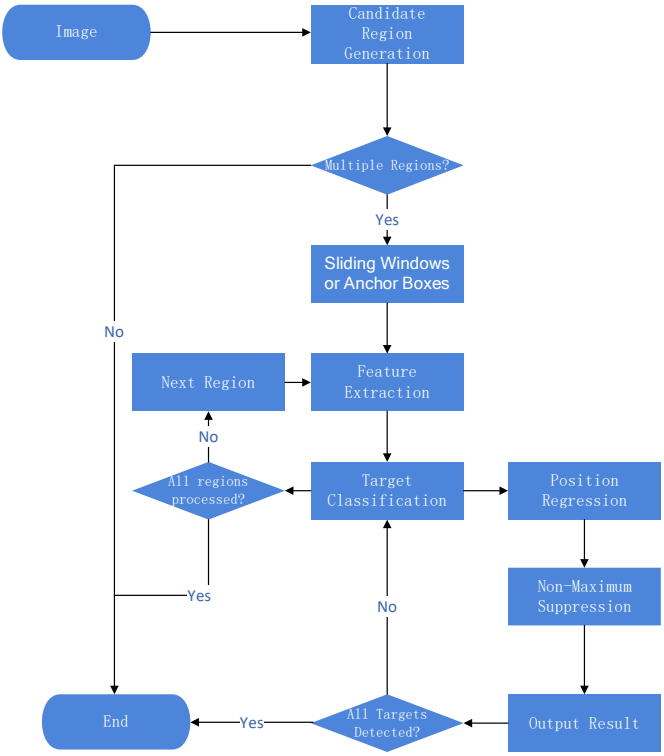

Fig. 1. Object Detection Framework（Photo/Picture credit :Original ）

Object detection is a multi-step process that involves several stages. First, candidate regions are generated using techniques like sliding windows or anchor boxes to identify areas in the image that may contain targets. Then, feature extraction is performed on each candidate region to capture visual information using methods like convolutional neural networks (CNN).

Once the features are extracted, they are passed through a classifier to determine whether each candidate region contains a target. The classifier can be binary, classifying targets as either present or not, or it can be multi-classification, distinguishing between different classes of targets.

For the candidate regions classified as targets, their boundary box positions are adjusted through position regression to more accurately surround the targets. Next, non-maximum suppression is applied to remove overlapping candidate boxes. This involves selecting the box with the highest confidence and suppressing other highly overlapping boxes to reduce redundant detections.Finally, the output includes information about the detected target category and its location, typically represented as a bounding box.

So far, the mainstream algorithms for target detection include Faster R-CNN, YOLO, CornerNet, CenterNet, etc. All these algorithms have unique advantages and characteristics and can be applied to different application scenarios and needs, but the basic algorithm flow of these algorithms is roughly the same. A typical object detection algorithm begins by obtaining an input image. Subsequently, candidate target regions are generated using techniques like sliding window and anchor boxes. These regions then undergo feature extraction, employing methods such as CNN to derive relevant features for each candidate region. Through target classification, the features are analyzed to identify potential targets, while position regression refines the boundary box positions for the regions classified as targets. To ensure accuracy, highly overlapping candidate boxes are suppressed through non-maximum suppression, retaining only the boxes with the highest confidence scores. Finally, the outcome of the process entails providing information about the detected target's category and its precise location within the image.

2.3 Target Tracking

Target tracking can be simplified to use the model to estimate the trajectory of the moving target in the scene, that is, the model needs to mark the target in all the image sequences, and provide more target information and features to the model, so that the model can automatically estimate the state and position of the target object in the subsequent frames through these feature information. A typical target tracking framework is shown in Figure 2:

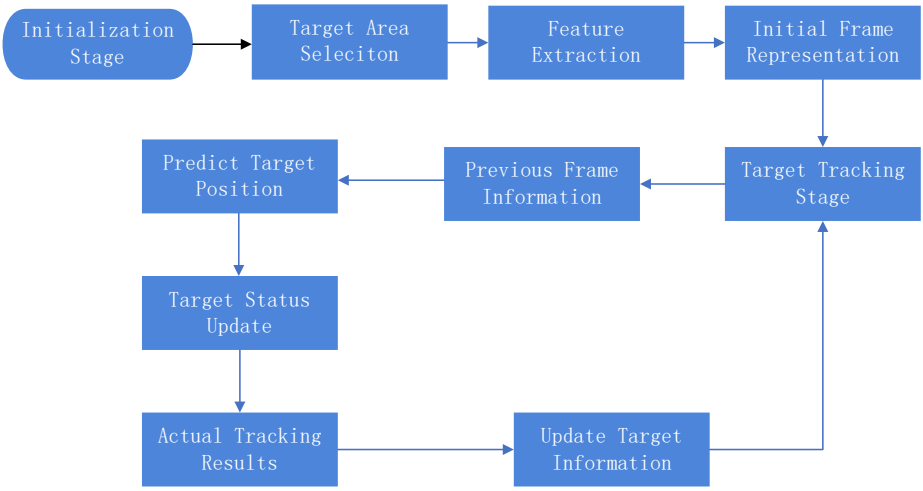

Fig. 2. Target Tracking Framework（Photo/Picture credit :Original ）

In the initialization stage, the target area to be tracked is chosen in the initial frame of the video, either through user input or automatic methods, and its feature representation is extracted. During the target tracking stage, for each subsequent frame, information and characteristics from previous frames are utilized to predict the target's position in the current frame. This enables the continuous tracking of the target. In the target status update stage, based on the actual tracking outcomes, the target's status information is refreshed. This includes its location, speed, size, and other attributes, facilitating a more precise prediction of the target's state in the next frame. Through the collaborative operation of these stages, the system effectively achieves stable tracking of the target within the video.

So far, the mainstream target tracking algorithms include SORT algorithm, DeepSORT algorithm, CenterTrack and FairMOT, etc., which basically combine detection and tracking, and can effectively select and track targets. In the typical target

tracking algorithm flow, the process begins with the manual or automatic selection of the target area, followed by the extraction of its feature representation. In each frame, features like color, texture, shape, and motion are extracted. Based on previous frame information, predictive models such as Kalman filters or neural networks forecast the target's position. This prediction is updated with the observed target position, often using Kalman filter status updates. Tracking results are evaluated for accuracy and robustness to determine success. Relocation or correction mechanisms may be applied for tracking failures or occlusion. The target's status information, like location and speed, is updated according to current tracking. This entire process is then iterated for continuous and consistent target tracking across successive frames.

3 Target Tracking Algorithm

3.1 Development Course

In the early days, the research task of target tracking was only single target tracking. This kind of research can solve the problem of finding the target again in the following video after the initial position of a frame target is given. After that, Comaniciu D et al. proposed to use Mean Shift to calculate the possible position of the object in the next frame [5], and then continue to iterate from the new position. Later, Basar T proposed to apply Kalman filter to predict the position of the object in the next frame of the video [6], and corrected the calculated result by using the result of human observation, and iterated continuously until the predicted result was closer and closer to the observation result. Since Kalman Filter is an equation used to describe linear systems and cannot be applied to nonlinear systems, Einicke et al proposed to approximate nonlinear systems as linear systems by using first-order Taylor expansion [7], and Extend Kalman Filter. Julier, S.J. The Unscented Kalman Filter is proposed to estimate the probability density of the nonlinear function with a series of determined samples [8]. Later, in modern times, Isard et al. proposed Particle Filter to estimate the posterior probability of an object state by means of Monte Carlo random sampling [9].

The Two-Shot algorithms are SORT and DeepSORT. This algorithm introduces a target detector to detect each frame of the video and sends the detection results to the tracker, which assigns a unique ID to each detected target, called the Re-ID model. SORT algorithm uses Kalman filter to predict the possible position of the target in the next frame, and compares the predicted result with the observed result [10]. Then the

Hungarian algorithm is used to assign the detected result. If there is no corresponding predicted result in the detected result, a new ID is assigned to the detected result. If one of the targets is blocked, SORT algorithm will lose this target. To solve this problem, DeepSORT algorithm is proposed and a feature extraction network is introduced to calculate the features of the monitoring frame and then match [11].

Examples of One-Shot algorithms include FairMOT, JDE, and CenterTrack. FairMOT algorithm is based on CenterTrack and adds Re-ID model to enable it to detect and track objects at the same time [12]. The JDE algorithm uses joint target detection and embedded vector learning to track the target, and its accuracy is improved. On the basis of CenterNet, CenterTrack down-samples and adds the pictures of the current frame, the pictures of the previous frame and the hotspot map, and then upsamples the information of the current frame through convolution and Batch Normalization [13]. Because the application scenario of multi-target tracking is much higher than that of single target tracking, this paper mainly reviews the multi-target tracking algorithm

3.2 Deep-SORT

Deep-sort (Deep Simple Online and Realtime Tracking) is based on the improvement and enhancement of SORT algorithm. It introduces Deep neural networks and combines traditional target tracking methods to enable the algorithm to capture richer visual features of targets. It also improves the smoothness and consistency of the target trajectory.

The first step of the DEEP-SORT algorithm is to use a pre-trained target detector to detect the target object in each frame. The target detector identifies the target position in the image and generates bounding boxes and corresponding features for each target. Then, for each detected target, DEEP-SORT extracts vectors that describe the appearance and characteristics of the target, called embedding vectors. A deep convolutional neural network (DEEP-CNN) is used to convert the target's image blocks into embedded vectors that capture the target's visual features. In each frame, the data association of the target is performed by calculating the similarity between the embedded vectors. By matching the embedding vector in the current frame with the target in the previous frame, the trajectory of the target can be established and the location of the target in the current frame can be predicted. DEEP-SORT uses a method called the "maximum weight Hungarian algorithm" to optimally assign matches between embedded vectors, which helps to deal with overlap, occlusion, and crossover between multiple targets.

The algorithm assigns targets to trajectories, maintaining a unique trajectory identification for each target. Finally, DEEP-SORT constantly updates existing trajectories according to new detection results and data association information, and screens out some unstable trajectories by setting thresholds to deal with some uncertainties of trajectory tracking, thereby improving the stability and accuracy of tracking.

Deep-sort algorithm combines Deep learning and traditional target tracking methods, overcomes the limitations of many traditional target tracking methods, has high accuracy, multi-target processing capability, online identity switching and other advantages, and is suitable for more complex practical application scenarios.

3.3 FairMOT

FairMot algorithm achieves high accuracy, robustness and fast operation in multi-target tracking tasks through end-to-end training, accurate target detection, spatio-temporal correlation modeling and efficient feature extraction. The goal of FairMOT algorithm combined with FAIRMOT algorithm is to achieve accurate and fair target tracking, emphasizing fair treatment of all targets. The typical framework of FairMOT algorithm is shown in Figure 3:

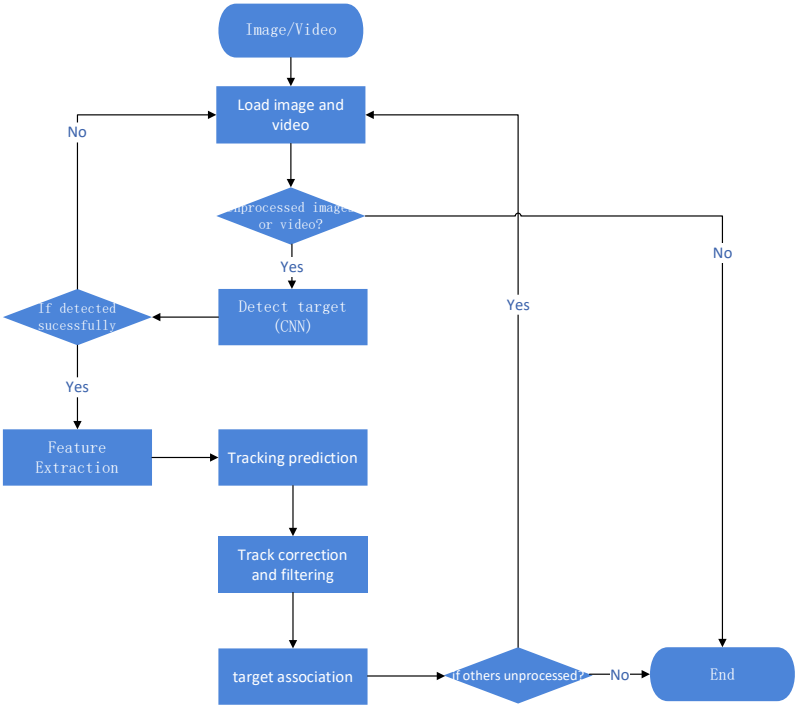

Fig. 3. FairMOT Framework（Photo/Picture credit :Original ）

The first step in the FairMOT algorithm framework is to perform object detection on an image or video frame using a pre-trained object detector, such as YOLO or RetinaNet, to obtain the location information of the target and the binding frame. Then, for each detected target, DEEP-CNN is used to extract its feature embedding vector to capture the visual features and appearance information of the target. Then, in the target association stage, FairMOT uses a fair association algorithm to ensure that all targets are treated equally in the association process, thus avoiding situations where some targets are ignored or given insufficient attention. Next, the tracking prediction is carried out, by matching the embedding vector similarity of the target in the previous frame, the multi-target association is achieved by using the Hungarian algorithm and other methods, and the motion trajectory of the target is established. Finally, in the process of multi-target tracking, the track information of the target is continuously updated, and the new detection results are used to adjust and update the track of the target to ensure the consistency and accuracy of the tracking results.

FairMOT algorithm combines target detection, feature extraction, target association and other technologies to track all targets accurately and fairly, which makes the algorithm perform well in the scene concerned with the equality of targets.

3.4 Goal tracking based on deep reinforcement learning

In the application of general scenes, the tracker will produce tracking deviation and other problems in the case of complex scenes caused by lighting or occlusion, resulting in the loss of tracking targets. In view of this, Deep Reinforce-ment Learning has more advantages in solving sequential decision tasks and tasks in which the loss gradient is difficult to calculate in discrete action space. By using DRL to build the generator, the 3D features of the sample class containing the hidden 3D information can be obtained by means of discrete sampling. The target tracking algorithm based on deep reinforcement learning integrates the obtained three-dimensional features with the previous two-dimensional features of the tracker to make up for the lack of tracking target loss in complex scenes [14].

The tracking framework built based on Deep reinforcement learning is named Deep Reap-forcement Generate Tracker (DRGT). By mining potential 3D information of the target, a 3D like feature is generated to optimize the 2D features and parameters of the

CNN tracker. To achieve robust performance of the tracking task, the DRG's 3D feature generation part and double-ended CNN are used as the core feature extraction part, and then the sampled image slices are input into the deep reinforcement learning network as states.

The first step of the target tracking algorithm based on DRGT is perception and feature extraction. Deep learning techniques are utilized to extract visual features from input images or video frames, aiding in the identification and localization of the target. The extracted features are then used to construct a representation of the target's state, which includes information such as its position, speed, direction, size, and the surrounding context. Based on the current state, the reinforcement learning agent takes actions and makes decisions to update the tracking state of the target. These actions could involve adjusting the predicted position or changing the tracking strategy. A crucial aspect of the process is the design of the reward function, which evaluates the agent's behavior and provides feedback to guide the learning process. The reward function considers factors like accuracy, stability, and robustness of the target tracking. Through training and optimization using techniques such as deep Q networks or policy gradients, the reinforcement learning agent progressively improves its performance. Once trained, the algorithm can perform target tracking in real scenes by selecting appropriate actions based on the current state and constantly updating the predicted location of the target. The effectiveness of the tracking can be evaluated by comparing it with the actual trajectory of the target. Overall, target tracking involves the integration of perception, decision-making, and learning processes to achieve accurate and reliable tracking results.

This object tracking algorithm based on deep reinforcement learning trains the model through reinforcement learning, which improves the practicality and tracking efficiency of the model. Compared with similar algorithms, this algorithm has higher robustness.

4 Real-time target tracking

Real-time performance in target tracking is an important requirement of video target detection and a major problem in the development of target tracking algorithms. Real-time means that the system can produce accurate tracking results in a given time range. Target tracking involves continuously locating and tracking one or more moving objects in a video or image sequence. However, due to computational resources, algorithm

complexity and data processing, the real-time problem may affect the performance of the tracking system.

In order to solve this problem, the researchers put forward the goal tracking based on deep reinforcement learning. In order to improve the efficiency of environmental sampling and the data processing capability of model training, the researchers proposed a variety of parallel frameworks and optimization mechanisms. For example, Gorila framework provides a general parallel reinforcement learning architecture. Nair et al proposed to synchronize parameters and gradients of learners through a parameter server to realize parallel exploration of multiple agents [15]. In order to further increase parallelism and efficiency, Horgan et al proposed a Priority Experience Replay mechanism and developed a distributed priority Experience replay model (Ape-X) based on it [16]. This approach prioritizes the selection of more valuable past experiences to train the model more effectively. On the other hand, Mnih et al. introduced the Asynchronous Dominant Actor Critic (A3C) framework based on the actor-Critic (AC) structure [17]. This framework allows multiple agents to interact in parallel in different environments and use asynchronous methods to update network parameters. However, A3C has some limitations, such as not being able to efficiently utilize heterogeneous computing resources, such as Gpus. To solve this problem, OpenAI proposed the Synchronous Dominance Actor Critic (A2C) framework [18]. This algorithm does not require the use of Replay Memory, and has a greater ability to deal with continuous action Spaces. However, the complexity of the algorithm has also increased, increasing the difficulty of algorithm debugging and maintenance, and the demand for computing resources is higher, such as CPU, GPU, etc., which limits the applicability of the algorithm.

The real-time problem can also be solved by combining fast detector assistance, such as using the YOLO (You Only Look Once) algorithm, to pre-locate the target and then use a lighter tracking algorithm for real-time tracking. On this basis, the scholars also put forward suggestions to improve the minimum width and depth of YOLOv5s. The model mainly consists of input layer, Backbone network, Neck network and detection end. For pre-processing, they performed selective data enhancement on the dataset, as well as adaptive anchor frame computation and image filling processing. In the Backbone network, they choose the infrastructure in MobileNetV3 to replace the original structure, to achieve the compression of the model, making it more suitable for deployment. Neck Network is composed of Feature Pyramid Network (FPN) structure and Pixel Aggregation Network (PAN) structure [19]. These structures can effectively

extract target features and location information from different layers and communicate it between layers. At the detection end, after obtaining the feature maps of different levels, they perform the target prediction of different sizes according to the original image. This design enables the model to capture the target information of different scales better and has better real-time performance. However, fast detectors and lightweight trackers usually deal with the target at different levels, so their working methods and performance are quite different, so when the target characteristics change greatly or the target appears abnormal, lightweight trackers may be difficult to adapt, resulting in tracking failure.

5 Conclusion

This paper summarizes the development and classification of target tracking algorithms, analyzes the improvement effect of different target tracking algorithms, gives the method framework, and analyzes the function. This paper also elaborates and analyzes the mainstream solutions to the real-time problem in the field. While reducing the impact of real-time, there are also problems such as higher resource consumption and lower robustness. Therefore, as for the research on target tracking, I think there are still some problems that need to be improved, including:

1) At present, scholars pay too much attention to the optimization of target tracking algorithms, which effectively reduces the influence of occlusion on target tracking and increases the accuracy and portability of algorithms. However, at the same time, the complexity of algorithms is also improved, and the real-time performance is also reduced, which will have an impact on the overall stability and application effect of target tracking. Therefore, in the future work, the focus can be placed on the real-time optimization of target tracking.
2) Most current algorithms may have accumulated errors in long-term tracking, resulting in target position deviation. Therefore, the subsequent work can try to introduce timing information, such as LSTM, into the algorithm to capture the movement pattern of the target.

With the continuous development of The Times, goal tracking can be applied in more and more fields, but the demand for goal tracking in various fields is also gradually increasing. Although the field of target tracking still faces some challenges, with the continuous progress of technology, I believe that the real-time problem, one of the basic problems of target tracking, will be further and deeper discussed.

References

1. Jiao Shuai, Wu Yingnian, Zhang Jing, et al.: Social Distance Detection and Tracking Algorithm Based on Improved YOLOv3 and Kalman Filter[J]. Science Technology and Engineering 22(22), 9712-9720 (2022).
2. Xie Xiuying.: Research on Pedestrian Re-identification Method Based on Video Object Tracking (Doctoral dissertation). Guangzhou University (2023).
3. Fiaz., Mustansar., et al.: "Handcrafted and deep trackers: Recent visual object tracking approaches and trends." ACM Computing Surveys (CSUR) 52(2), 1-44 (2019).
4. Wu, Yi, Jongwoo Lim, and Ming-Hsuan Yang.: "Object tracking benchmark." IEEE Transactions on Pattern Analysis and Machine Intelligence 37(9), 1834-1848 (2015).
5. Comaniciu D., Meer P. Mean shift: A robust approach toward feature space analysis[J].IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 603–619 (2002).
6. Kalman R E.: A new approach to linear filtering and prediction problems[J] (1960).
7. Einicke G A., White L B.: Robust extended Kalman filtering[J].IEEE Transactions on Signal Processing 47(9), 2596–2599 (1999).
8. Julier S J., Uhlmann J K.: Unscented filtering and nonlinear estimation[J].Proceedings of the IEEE 92(3), 401–422 (2004).
9. Isard M., Blake A.: Condensation—conditional density propagation for visual tracking[J].In ternational Journal of Computer Vision 29(1), 5–28 (1998).
10. Bewley A., Ge Z., Ott L., et al.: Simple online and realtime tracking[C].In IEEE International Conference on Image Processing 3464–3468 (2016).
11. Wojke N., Bewley A., Paulus D.: Simple online and realtime tracking with a deep association metric[C].In IEEE International Conference on Image Processing 3645–3649 (2017).
12. Zhou X., Koltun V., Krähenbühl P.: Tracking objects as points[C].In Proceedings of the European Conference on Computer Vision 474–490 (2020).
13. Zhang Y., Wang C., Wang X., et al. FairMot: On the fairness of detection and re-identification in multiple object tracking[J].International Journal of Computer Vision 129(11), 3069–3087 (2021).
14. Wang, Z.: Research on Multi-Feature Based Object Tracking Method using Deep Reinforcement Learning (Doctoral dissertation). Liaoning Normal University (2022).
15. Li M., Andersen D G., Smola A J., et al.: Communication efficient distributed machine learning with the parameter server[J]. Advances in Neural Information Processing Systems 27, 19-27 (2014).

16. Horgan D., Quan J., Budden D., et al.: Distributed prioritized experience replay[J]. arXiv preprint arXiv:1803.00933 (2018).
17. Mnih V., Badia A P., Mirza M., et al. : Asynchronous methods for deep reinforcement learning[C]//International conference on machine learning 1928-1937 (2016).
18. Wei Q., Wang L., Liu Y., et al.: Optimal elevator group control via deep asynchronous actor–critic learning[J]. IEEE transactions on neural networks and learning systems 31(12), 5245-5256 (2020).
19. Liu, Z.: Improved Algorithm for Road Object Tracking based on YOLOv5 and DeepSort. Automotive Practical Technology 47(22), 40-44 (2022).