Hindawi
IET Circuits, Devices & Systems
Volume 2023, Article ID 6620581, 13 pages
https://doi.org/10.1049/2023/6620581

Research Article

Generative Target Tracking Method with Improved Generative Adversarial Network

Yongping Yang and Hongshun Chen

School of Information Technology, Beijing Normal University, Zhuhai, Guangdong 519000, China

Correspondence should be addressed to Yongping Yang; yangyongping@bnuz.edu.cn

Received 8 June 2023; Revised 28 July 2023; Accepted 18 August 2023; Published 23 October 2023

Academic Editor: Shashikant Patil

Multitarget tracking is prone to target loss, identity exchange, and jumping problems in the context of complex background, target occlusion, target scale, and pose transformation. In this paper, we proposed a target tracking algorithm based on the conditional adversarial generative twin networks, using the improved you only look once multitarget association algorithm to classify and detect the position of the target to be detected in the current frame, constructing a feature extraction model using generative adversarial networks (GANs) to learn the main features and subtle features of the target, and then using GANs to generate the motion trajectories of multiple targets, finally fuzing the motion and appearance information of the target to obtain the optimal match. The optimal matching of the tracked targets is obtained. The experimental results under OTB2015 and IVOT2018 datasets demonstrate that the proposed multitarget tracking algorithm has high accuracy and robustness, with 65% less jumps and 0.25% more accuracy than the current algorithms with minimal identity exchange and jumps.

1. Introduction

Targets in complex real-world scenes are susceptible to interference factors such as motion blur, low resolution, illumination scale changes, and occlusion deformation, so designing a robust tracking algorithm to achieve robust real-time tracking of targets still faces a great challenge among existing target tracking algorithms, which are mainly divided into traditional classical target tracking algorithms based on the artificial features and deep network target tracking algorithms based on the depth features [1]. The overall artificial features widely used in traditional target tracking algorithms can be divided into grayscale features, color features concave, and gradient features. Grayscale features are the simplest and most intuitive feature representation with high-computational efficiency [2]. However, color features are more affected by illumination and are susceptible to illumination changes as well as background interference with similar colors [3]. Gradient features characterize the appearance by counting the local gradient distribution of the target image. The widely used gradient feature in the target tracking algorithms is the HOG ((histogram of oriented gradient) feature [4]. The core idea of the HOG feature is to make full use of the chunking unit to extract the gradient information of the image, so that the appearance and shape of the local target can be well-described by the gradient or the directional density distribution of the edges, while having good invariance to the illumination changes.

Although the above artificial features are rich in target information, they cannot extract higher level semantic information and require a strong priori information, which are highly adaptable in the specific scenes but difficult to achieve robust tracking of the targets in complex scenes [5]. In recent years, deep neural networks (DNNs) have made breakthroughs in image classification and target detection due to their excellent feature learning and representation capabilities, which indicate that deep features have powerful characterization capabilities for targets, and therefore people apply deep learning to the visual tracking [6]. The accuracy of the depth feature-based target tracking algorithm has great advantages over the artificial feature-based target tracking algorithm; however, the complexity of the adopted network structure leads to a large amount of computation, which greatly limits the tracking speed of the tracking algorithm and makes the accuracy and real-time performance of the

tracking algorithm not reach a good balance [7]. The position response of the target is obtained. Due to the advance offline training and online similar evaluation during tracking, the speed aspect is far beyond the real time while ensuring high-accuracy tracking.

The feature extraction networks used in the twin network framework are relatively shallow Alexnet [8] networks, and the motion blur and low-resolution video frames generated when the target is in violent and fast motion make the tracked target indistinguishable, making it difficult for the Alexnet network to extract the effective features of the target and making the model drift easily, leading to poor tracking results or even tracking failure [9].

In order to address the problem that the lack of an effective adjustment mechanism leads to the degradation of the model's characterization ability when the target generates motion blur and low-resolution video frames due to fast motion, this paper embeds a conditional adversarial generative network module (CGAN) in the feature extraction network to advance the model's characterization ability and enhance the robustness of the tracking algorithm in the case of motion blur and low resolution [9]. In order to fully validate the effectiveness of the CGAN deblurring network module proposed in this paper on the tracker performance improvement, a test evaluation was conducted and compared with the traditional deblurring algorithm based on the Lrsiamfo, the original Siamfo, and several other classical target algorithms for analysis. The experimental results show the effectiveness of the method in this paper.

2. Related Work

In recent years, many scholars have conducted in-depth research on target tracking technology, and many excellent target tracking algorithms have emerged. At present, target tracking algorithms are generally divided into classical traditional tracking algorithms based on the artificial features and deep learning tracking algorithms based on the depth features [10]. The development history of target tracking algorithms is mainly divided into three development stages.

Target eye tracking algorithms mainly use statistics-based iterative prediction and feature point-based optical flow methods. The classical target tracking algorithms are mainly mean drift tracking algorithm, particle filtering tracking algorithm, and optical flow tracking algorithm.

Single-target long-time tracking algorithm I-TLD24 algorithm was first proposed by Cui et al. [11]. This algorithm is a discriminative tracking algorithm that learns the features of positive and negative sample ports 2–26 through training and then uses the features to collect samples at predicted locations for classification to distinguish the target from the background [12]. The emergence of TLD has greatly promoted the development of the tracking field, Khattak et al. [8] proposed the correlation filter-based tracking algorithm KCF32 based on MOSSEL3 and CSK3 algorithms, KCF uses Fourier transform and circular matrix to correlate the template and sample in the frequency domain to obtain the outgoing response map, and the maximum response position is the target position. By Sha et al. [13], a circular matrix equivalent sliding window is designed to replace the dense sampling, which solves the problem of unbalanced positive and negative samples in the tracking algorithm, and greatly reduces the computational effort and increases the tracking speed to more than 100s. The core-needle biopsy by Ali et al. [14] uses color features to characterize the appearance of the target, refines the RGB three-channel color into 11 color features, and then uses principal component analysis to reduce the 11-dimensional color features to 2 dimensional, and selects the most significant color features as the target features for tracking. A new transfer learning approach is proposed to improve the stability and training speed of generative adversarial networks (GANs). The method involves using pretrained variational autoencoders with the controlled degrees of freedom. By sampling from a standard normal distribution and inputting it into the model, the method achieves faster convergence and increased model accuracy across the different datasets. This approach addresses the limitations of GANs and shows promising results in the domain of generative modeling [15]. Two innovative visual object tracking approaches are introduced to improve the classifier performance and integration with existing trackers. The first method, MDResNet, replaces MDNet's convolutional layers with ResNet-50 layers to enhance the feature extraction capabilities. The second method, ROIAL, integrates GAN networks with MDResNet and MDNet, harnessing GAN-based learning to further enhance tracking performance. Both approaches aim to leverage ResNet's strengths and GAN-based learning to achieve superior results in the visual object tracking [16]. Zin [17] proposes a novel object tracking method that uses generative adversarial learning and incorporates distractors and a distractor generator into a Siamese network. This approach enables robust tracking in scenes with dramatic shape changes or environmental variations by removing distractors from the input instance search image. The method utilizes a generalized intersection over union (GIoU) loss during training, leading to more accurate tracking results. Experimental evaluations on challenging benchmarks demonstrate the method's effectiveness and precision in the object tracking.

Since 2012, deep learning has been applied to the field of target tracking and many algorithms with excellent tracking results have been proposed by the experts and scholars due to the large number of results achieved in the field of classification and detection. By Yau et al. [18], DLT network was proposed, which used the idea of offline pretraining + online fine-tuning to solve the problem of insufficient training samples in tracking. By Zhou et al. [19], FCNT38 was proposed to build a feature filtering network and two complementary heat map prediction networks by analyzing different layers of feature maps, which have better tracking effect on targets that produce deformation. Zhou et al. [19] proposed the end to end target tracking algorithm Mdnet, which creatively proposed a multibranch training method, i.e., pretraining V Genet using video dataset to obtain the features of the tracked target, and robustly tracking the target by searching for the appropriate tracker in each frame, but with poor real-time performance and speed of only 1 FPS.

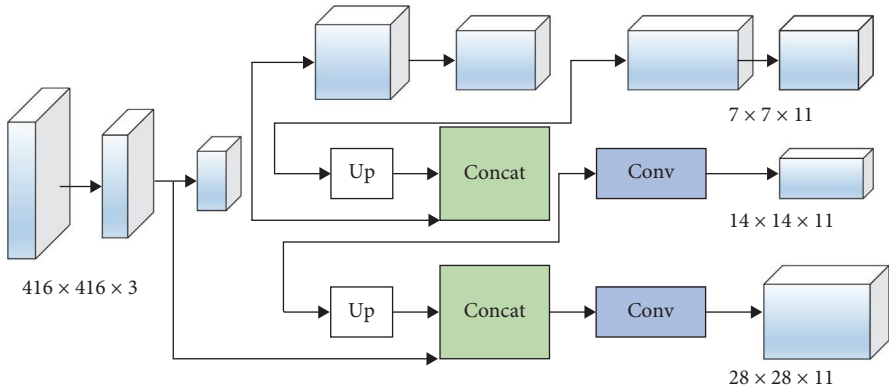


FIGURE 1: Structure of YOLO-based human face detection network. YOLO, you only look once.

3. Algorithm Framework

The overall algorithm framework proposed in this paper consists of four modules, which are detection module, feature extraction module, prediction module, and matching module [20, 21]. Our model structure is shown in Figure 1. Firstly, by detecting the current frame of the tracked video sequence, we can obtain the position information of the target. Then, the relevant target frame and edge frame are detected simultaneously, which can compensate for the coarse features of the target frame. The feature extraction module consists of two networks for feature extraction. Net1 is a pedestrian feature extraction network based on generative adversary, and Net2 is a common module [22, 23]. The state of each target's motion trajectory is also estimated using a generative adversarial-based pedestrian multitarget trajectory prediction network [24]. The above information is fed to the final matching module for trajectory update to achieve continuous tracking of each target.

3.1. You Only Look Once-Based Multitarget Correlation Detection Algorithm. In this paper, we propose a you only look once (YOLO)-based multitarget correlation target detection algorithm, which mainly solves the difficult problem of pedestrian target detection in the dense places [25, 26]. Adding target features can increase the difference in appearance features when the target appearance is similar. YOLO uses this object detection method, while GANs are used to generate similar data instances. YOLO is a popular object detection algorithm, which is specialized in real-time object detection through bounding box predictions and class probabilities on an image grid, while GANs are used to generate data instances resembling a given training dataset. However, beyond YOLO and GANs, various other methods and architectures are available for feature construction and representation. For example, convolutional neural networks performs better in capturing the hierarchical features from edges to object shapes in image-related tasks, whereas RNNs are valuable for handling sequential data like text or time series, autoencoders provide unsupervised learning for the feature extraction, and transfer learning allows leveraging pretrained models as feature extractors or for the fine-tuning on new tasks. Moreover, transformers, originally developed for natural language processing, have shown powerful feature generation and representation capabilities even in computer vision tasks. In this paper, the network of YOLO is improved, and the network structure is shown in Figure 1. First, the detection images are fed into the network, and the output layer includes three feature maps of different scales to ensure the detection capability of the model for objects of various scales. The vectors containing the features are sorted in descending order according to the confidence level, and the position information of the box with top confidence level (bounding box, bbox for short) is first traversed through other bboxes for IOU calculation. If the value is greater than the threshold, the bbox is considered as a duplicate box and is eliminated [27–29]. Then we repeat the above operation from the top2 bboxes of the remaining bboxes after rejection until the end of the iterative process, and finally obtain the streamlined detection results.

The improved output layer adds four dimensions to the original one to store the position information of the target frame associated with the target frame, which are the horizontal position, vertical position, width and height information of the target frame relative to the target frame.

$t_x^{\text{person}}, t_y^{\text{person}}, t_w^{\text{person}}, t_h^{\text{person}}$ corresponds to the first four dimensions of the output feature, and $t_x^{\text{person}_{\text{f}}\text{ace}}, t_y^{\text{person}_{\text{f}}\text{ace}}, t_w^{\text{person}_{\text{f}}\text{ace}}, t_h^{\text{person}_{\text{f}}\text{ace}}$ corresponds to the last four dimensions of the output feature. When the detected object is a target, the relevant calculation is not performed. In this paper, a more stable L1 loss is used, and the loss function is as follows:

$$\text{loss}_{x\_y}^{\text{person}} = \lambda_{x\_y}^{\text{person}} \sum \sum \begin{bmatrix} \left| x_i^{\text{person}} - \widehat{x}_i^{\text{person}} \right| + \\ \left| y_i^{\text{person}} - \widehat{y}_i^{\text{person}} \right| \end{bmatrix}, \tag{1}$$

$$\text{loss}_{x\_y}^{\text{face}} = \lambda_{x\_y}^{\text{face}} \sum \sum \begin{bmatrix} \left| x_i^{\text{face}} - \widehat{x}_i^{\text{face}} \right| + \\ \left| y_i^{\text{face}} - \widehat{y}_i^{\text{face}} \right| \end{bmatrix} \tag{2}$$

$$\text{loss}_{w-h}^{\text{person}} = \gamma_{w-h}^{\text{peron}} \sum \sum \begin{bmatrix} \left| \sqrt{w_i^{\text{person}}} - \sqrt{\widehat{w}_i^{\text{peron}}} \right| + \\ \left| \sqrt{h_i^{\text{person}}} - \sqrt{h_i^{\text{person}}} \right| \end{bmatrix}, \tag{3}$$

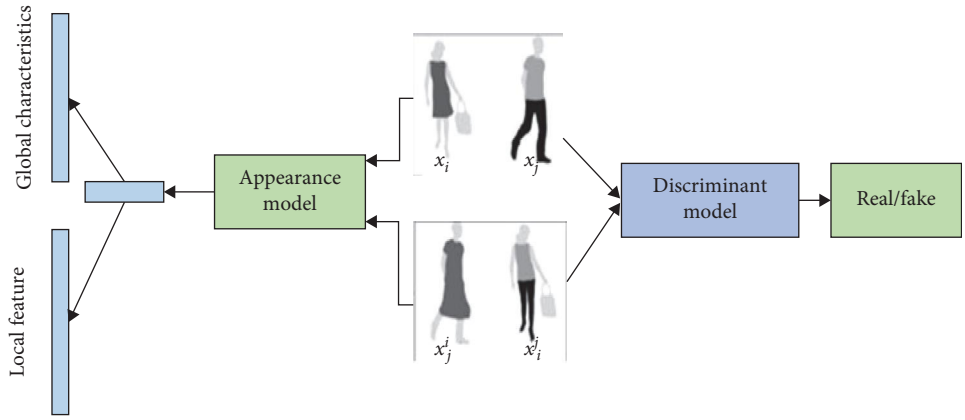


FIGURE 2: Features and discrimination model.

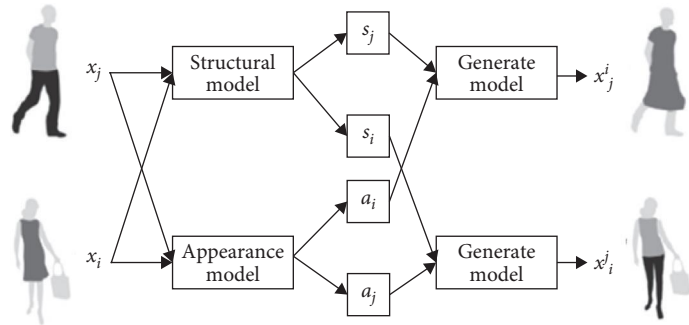


FIGURE 3: Schematic diagram of images generated by different D.

$$\text{loss}_{w\_h}^{\text{face}} = \gamma_{w\_h}^{\text{face}} \sum \sum \begin{bmatrix} 1\sqrt{w_i^{\text{face}}} - \sqrt{\widehat{w}_i^{\text{face}}}| + \\ \left|\sqrt{h_i^{\text{face}}} - \sqrt{h_i^{\text{tace}}}\right| \end{bmatrix}, \tag{4}$$

$$\text{loss}_{x\_y}^{\text{person}_{\text{f}}\text{ace}} = \lambda_{x\_y}^{\text{person}_{\text{f}}\text{ace}} \sum \sum \begin{bmatrix} \left|x_i^{\text{person}_{\text{f}}\text{ace}} - \widehat{x}_i^{\text{person}_{\text{f}}\text{ace}}\right| + \\ \left|y_i^{\text{person}_{\text{f}}\text{ace}} - \widehat{y}_i^{\text{person}_{\text{f}}\text{ace}}\right| \end{bmatrix}, \tag{5}$$

$$\text{loss}_{w\_h}^{\text{person}_{\text{f}}\text{ace}} = \gamma_{w\_h}^{\text{person}_{\text{f}}\text{ace}} \sum \sum \begin{bmatrix} \left|w_i^{\text{person}_{\text{f}}\text{ace}} - \widehat{w}_i^{\text{person}_{\text{f}}\text{ace}}\right| + \\ \left|h_i^{\text{person}_{\text{f}}\text{ace}} - \widehat{h}_i^{\text{person}_{\text{f}}\text{ace}}\right| \end{bmatrix}, \tag{6}$$

where $\text{loss}_{w\_h}^{\text{person}}$ is the target detection loss function; x_i^{person} is the predicted relative lateral position of the pedestrian; $\widehat{x}_i^{\text{person}}$ is the corresponding true label; y_i^{person} is the predicted relative longitudinal position of the pedestrian; $\widehat{\mathcal{Y}}_i$ person is the corresponding true label $\text{loss}_{w\_h}^{\text{face}}$ is the target detection loss function x_i^{face} is the relative lateral position of the target predicted by the algorithm; y_i^{face} is the relative vertical position of the target; $\widehat{x}_i^{\text{face}}, \widehat{y}_i^{\text{face}}$ are the true label; $w_i^{\text{person}}, h_i^{\text{person}}$ are the relative width and height of the target prediction. $\widehat{w}_i^{\text{person}}, \widehat{h}_i^{\text{person}}$ corresponding labels. $w_i^{\text{face}}, h_i^{\text{face}}$ are the relative width and height of the target prediction, $\widehat{w}_i^{\text{person}}, \widehat{h}_i^{\text{person}}$ are the corresponding label. $\lambda_{x\_y}^{\text{person}}, \lambda_{x\_}^{\text{face}}, \gamma_{w\_h}^{\text{person}}, \gamma_{w\_h}^{\text{face}}$ are the parameters.

3.2. Generative Adversarial Based Feature Extraction Algorithm. In the feature extraction module, this paper adopts a generative adversarial-based algorithm to extract the pedestrian features. Compared with the general deep learning feature extraction methods, the new data is generated by the generative adversarial, so that the feature extraction network can minimize the intra-class feature variation among the same ID images and distinguish the interclass features among different I image. In this paper, we use an encoder as the backbone network for recognition learning and learn the main features as well as fine features of the target using images generated under the different conditions.

With $X = \{x_i\}_{i=1}^N$ and $Y = \{y_i\}_{i=1}^N$ denoting the real images and their corresponding labels, N denoting the number of images, and $y_i \in [1, K]$ being the number of IDs identified in the dataset, two real images x_i and x_i in the training set are selected to generate a new image, and their structural codes or appearance codes are exchanged in the generation module. As shown in Figure 2, the generation module $G(a_i, S_j) \rightarrow x_j^i$ consists of an appearance coding model $E_{\text{a}} : x_i \rightarrow a_i$ and a structure coding model $E_{\text{s}} : x_j \rightarrow s_j$, where the structure coding allows the geometric and positional features of the target to be preserved. The discriminative model is used to discriminate the later generated image from the original real image.

For image generation with different IDs (Figure 3), given two images x_i and x_j, The generated image $x_j^i = G(a_i, s_j)$

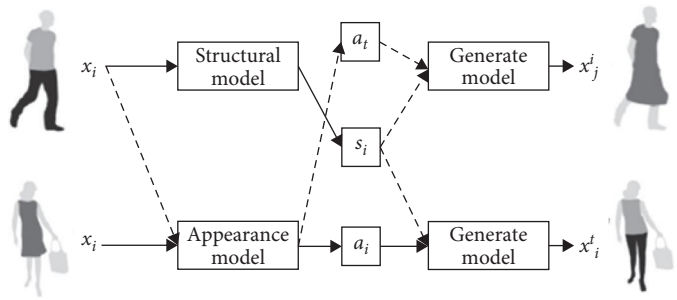


FIGURE 4: Schematic diagram of same D generated image.

needs to retain the appearance codes from x_i separately x_j and a_i for structural encoding s_j. Then, it should be possible to reconstruct the two latent encodings after encoding the generated image and to force the ID loss function on the generated image according to the encoding of the image in order to maintain the consistency of the identity ID.

$$L_{\text{recon}}^{\text{code}_1} = E\left[\left\|a_i - E_a\left(G\left(a_i, s_j\right)\right)\right\|_1\right], \tag{7}$$

$$L_{\text{recon}_2}^{\text{cod}_2}\left[\left\|s_j - E_s\left(G\left(a_i, s_j\right)\right)\right\|_1\right], \tag{8}$$

$$L_{\text{id}}^{c} = E\left[-\log\left(p\left(y_i|x_j^i\right)\right)\right], \tag{9}$$

where $p\left(y_i|x_j^i\right)$ is the predicted probability of x_j^i and belongs to the real label class x_i of y_j, which is provided as encoding in the generated x_j^i image. In addition, in this paper, an adversarial loss function is used to match the distribution of the generated images with the distribution of the real data.

$$L_{\text{adv}} = E\left[\log D(x_i) + \log\left(1 - D\left(G\left(a_i, s_j\right)\right)\right)\right]. \tag{10}$$

Reconstruction of images between any two images using the same identity ID, as shown in Figure 4, To reduce intraclass feature variation for a given image, x_i the generation module first learns how to reconstruct from itself x_i. In addition, the generator should be able to reconstruct $y_i = y_t$ by images x_i with the same identity x_t using I loss to distinguish between different identity IDs:

$$L_{\text{recon}}^{\text{img}_1} = E\left[\left\|a_i - E_a\left(G\left(a_i, s_j\right)\right)\right\|_1\right], \tag{11}$$

$$L_{\text{reeon}}^{\text{img}_2} = E\left[\left\|x_i - G_s\left(a_t, s_i\right)\right\|_1\right], \tag{12}$$

$$L_{\text{id}}^{s} = E\left[-\log\left(p\left(y_i|x_i\right)\right)\right], \tag{13}$$

where $p(y_i|x_i)$ is the predicted probability that the image appearance encoding belongs to the true label category.

The labels are dynamically assigned using a supervised model x_j^i which depends on the appearance encoding and structural encoding it obtains from x_i and x_j. For the discriminative module, in order for it to obtain the recognition capability of the main features of the image. In this paper, the discriminant module is trained by minimizing the information scatter between its predicted probability distribution $p\left(x_j^i\right)$ and the supervised predicted probability distribution $q\left(x_j^i\right)$.

$$L_{\text{prim}} = E\left[-\sum_K q\left(k|x_j^i\right)\log\left(\frac{p\left(k|x_j^i\right)}{q\left(k|x_j^i\right)}\right)\right]. \tag{14}$$

Instead of using generated data, this paper provides an alternative approach to generative branching by simulating the clothing changes of pedestrian targets in images for learning the main features. When pairs are trained in this way, the discriminator module is able to learn subtle I-related attributes unrelated to clothing. The images generated by the combination of different structural and appearance codes are considered as the same class of real images that provide the structural codes. For this implementation of the image minutiae mining discriminator module, training is performed using identity ID loss.

$$L_{\text{fine}} = E\left[-\log\left(p\left(y_i|x_j^i\right)\right)\right]. \tag{15}$$

In order to optimize the overall objective, the appearance encoder, structure encoder, decoder, and discriminator are trained together using the following weighted sum of losses:

$$\begin{aligned} L_{\text{total}}\left(E_a, E_s, G, D\right) &= \lambda_{\text{img}} I_{\text{reeon}}^{\text{img}} + L_{\text{recon}}^{\text{code}} + L_{\text{id}}^{s} \\ &+ \lambda_{\text{idd}} L_{\text{id}}^{c} + L_{\text{adv}} + \lambda_{\text{prim}} L_{\text{prim}} + \lambda_{\text{fine}} L_{\text{fine}}, \end{aligned} \tag{16}$$

where: $L_{\text{recon}}^{\text{img}} = L_{\text{recon}}^{\text{img1}} + L_{\text{recon}}^{\text{img2}}$ is the self (same as ID) discriminant loss in image reconstruction; $L_{\text{recon}}^{\text{code}} = L_{\text{recon}}^{\text{codel}} + L_{\text{recon}}^{\text{code2}}$ is the coding reconstruction loss in cross-identity (different IDs) generation; $\lambda_{\text{img}}, \lambda_{\text{id}}, \lambda_{\text{prim}}, \lambda_{\text{fine}}$ are the weight that controls the importance of the associated loss term.

3.3. Multitarget Path Prediction Algorithm Based on GAN. In the practical scenario of multitarget tracking, the actual situation of movement needs to be considered when predicting the trajectory of pedestrians with the multiple targets, and the activities of surrounding people also affect the walking

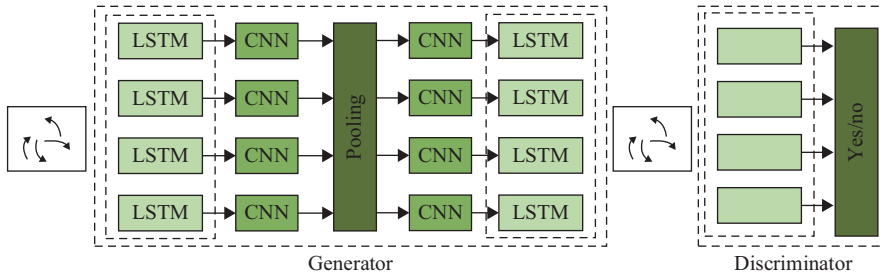


FIGURE 5: Multitarget path prediction based on generation countermeasure.

path of the target. In this paper, we adopt a multitarget path prediction algorithm based on a generative adversarial model to cope with complex human interactions, and predict future trajectories based on a generative adversarial encoder–decoder structure and a pooling module to simulate the pedestrian interactions. The target's relative position to several surrounding interfering targets is used as the input of the module, which is processed by the multilayer perceptron (MLP) and MaxPooling to obtain a vector pooling the position information of the target and the surrounding pedestrians to simulate the interaction between the target and the surrounding people.

The path prediction model in this paper is shown in Figure 5, and the whole consists of three main parts: generator, pooling module, and discriminator. The generator is based on the LSTM framework of encoding and decoding, and the pooling module is used to connect the hidden states of encoding and decoding. Finally, it is fed to the discriminator to determine whether the trajectory is true or not.

In the generator part, the position of each target is input to the LSTM cell that acts as an encoder to obtain a fixed-length vector e. The following loop is introduced:

$$\mathbf{e}_i^t = \varphi(x_i^t, y_i^t; W_{\text{ee}}), \tag{17}$$

$$h_{ei}^t = \text{LSTM}(h_{ei}^{t-1}, \mathbf{e}_i^t; W_{\text{encoder}}), \tag{18}$$

where: t is the sequence; i is the target; $\varphi()$ is the embedding function with ReLU nonlinearity; W_{ee} is the embedding weight; W_{encoder} is the weight of LSTM.

In this paper, we use a pooling module to simulate the interactions between pedestrians coming and going, and after the observable moment, the hidden states of all people in the scene are pooled and each person gets a combined tensor. The generation of the output trajectory is regulated by initializing the hidden states of the decoder.

$$c_i^t = \gamma(P_i, h_{ei}^t; W_{\text{c}}), \tag{19}$$

$$h_{di}^t = [c_i^t, Z], \tag{20}$$

where: $\gamma()$ is the MLP containing the ReLU nonlinearity; W_e is the embedding weight, and the subsequent prediction is as follows:

$$\mathbf{e}_i^t = \phi(x_i^{t-1}, y_i^{t-1}; W_{\text{ed}}), \tag{21}$$

$$P_i = \text{PM}\left(h_{d1}^{t-1}, \ldots, h_{\text{dn}}^t\right), \tag{22}$$

$$h_{\text{di}}^t = \text{LSTM}\left(\gamma\left(P_i, h_{\text{di}}^t\right), \mathbf{e}_i^t; W_{\text{decoder}}\right), \tag{23}$$

$$\left(\widehat{x}_i^t, \widehat{y}_i^t\right) = \gamma\left(h_{\text{di}}^t\right), \tag{24}$$

where: $\varphi()$ is the embedding function with ReLU nonlinearity; W_{ed} is the embedding weight.

The discriminator consists of a decoder with $T_{\text{real}} = [X_{i,}, Y_i]$, $T_{\text{fake}} = [X_{i,}, \widehat{Y}_i]$ inputs and classifies them as true or false. A MLP is applied on the final hidden state of the decoder to obtain the final classification score. A random sample of z in $N(0, 1)$ and the "best" prediction in the sense of L2 is used as the prediction of this paper, and k candidate output predictions are generated.

$$L_{\text{variely}} = \min_k \left\| Y_i - \widehat{Y}_i^k \right\|_2. \tag{25}$$

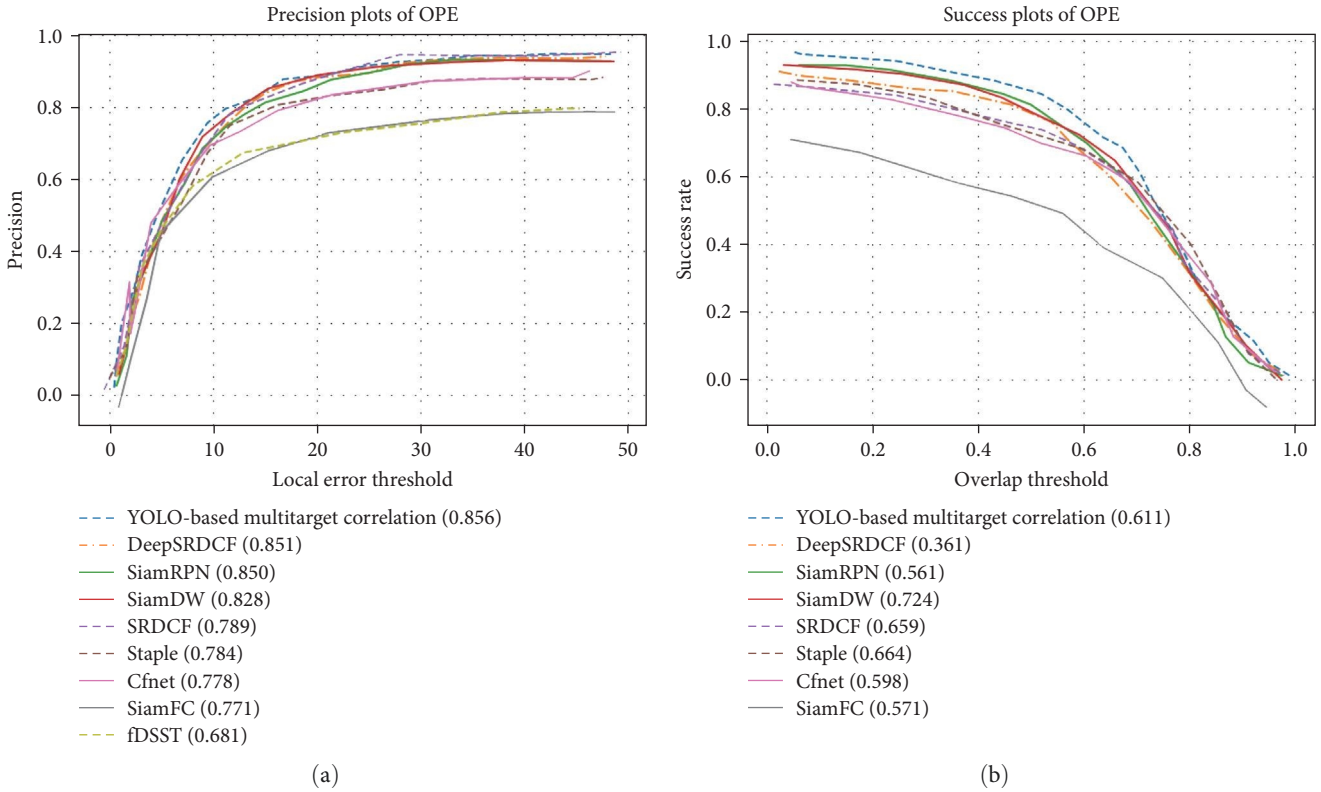
4. Experiment and Analysis

In order to verify the effectiveness of this algorithm, we use OTB2015 by Gao et al. [22] and IVOT2018 by Ali et al. [23] datasets as validation sets and compare them with several classical tracking algorithms, based on the various experimental analyses, we can see that this algorithm has excellent performance.

4.1. Training Set. In the training phase, for the training of the conditional adversarial generative network model, the Gopro dataset is used, which contains 2013 pairs of blurred and clear images, and the training dataset for the full convolutional twin network tracker is selected from two open standard datasets, GOT-10k and ILSVRC2015-VID. Over 1.5 million manually labeled bounding boxes. The ILSVRO2015VID contains more than 30 targets with over 4,000 videos and over 1 million frames labeled.

4.2. Analysis of Experimental Results

4.2.1. Quantitative Analysis of OTB2015. OTB2015 has 100 manually annotated video sequences containing 11 attributes, which represent common difficulties in the current target tracking field. The algorithm in this paper is compared with CFNet [24], SiamDW, SiamRPN, SRDCF [25], fDSST [27], Staple [28], and representative trackers of SiamFC.



(a) (b)

Figure 6: Comparison of accuracy and success rates of different algorithms on OTB2015 dataset.

As shown in Figure 6, the quantitative comparison results between the algorithm in this paper and the comparison algorithm on the OTB2015 data are shown. The accuracy of this algorithm reaches 85.6% and the success rate reaches 63.7%, both of which are better than other comparison algorithms. Compared with the benchmark algorithm SiamFC, the algorithm in this paper clearly achieves a good performance, with an improvement of 8.5% points in accuracy and 5.5% points in success rate.

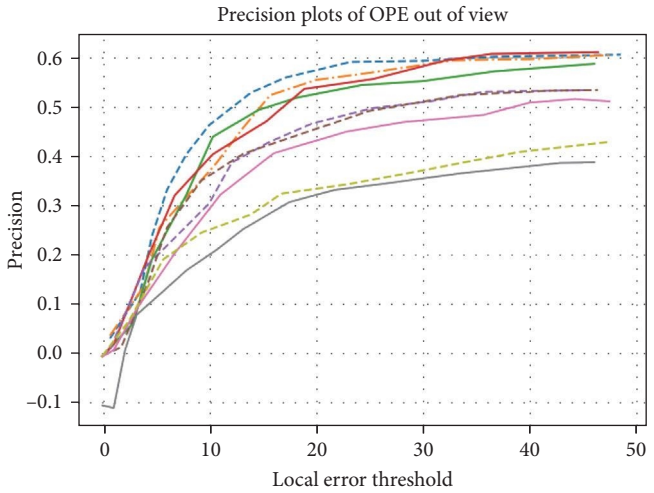
The results for various difficult attributes in the OTB2015 dataset are shown in Figure 7, especially for low resolution, fast motion, and motion blur of the object, and achieved 0.933, 0.832, and 0.849 in the accuracy rate, respectively. Further proving the effectiveness of the conditional adversarial generative network model and the effectiveness of conditional adversarial generative network model and multilayer feature fusion in the target tracking is further demonstrated.

4.2.2. OTB2015 Qualitative Analysis. In order to compare the differences between this algorithm and other excellent algorithms, the test results of OTB2015 were selected for the qualitative analysis. The test results are shown in Figure 8 from top to bottom for Skating1, Coke, Motor Rolling Skiing Carscale, Football video sequences, six video sequences containing six challenging scenes such as illumination change, occlusion, motion blur, low resolution, scale change, and similar background interference. Red is the algorithm of this paper, green, blue, black, and pink are Siamdw, SiamFC, Cnet, and SiamRPN algorithms, respectively.

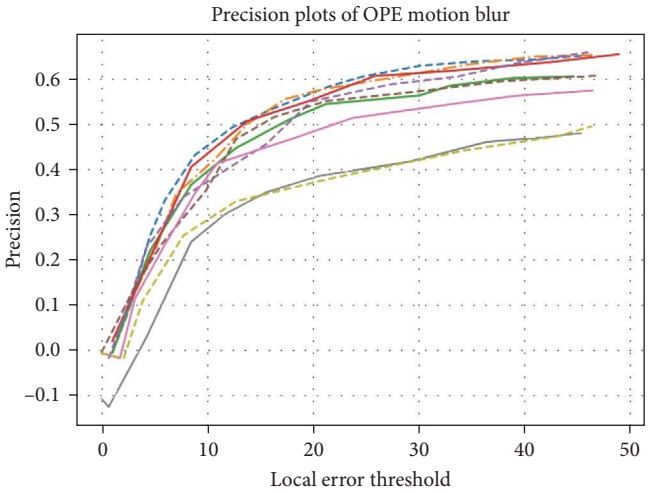
(1) Illumination changes: in the Skating video sequence, the target moved rapidly, which also included occlusion, illumination changes etc., which greatly affected the tracking process. At around frame 173, the target is blocked and the algorithms show some tracking drift. Around frame 31, the SiamFC algorithm failed to track the target because the target features were not obvious due to the change of illumination, but the algorithm in this paper could obtain more target features due to the addition of multifeature fusion model, so that it could make effective judgment on the current target position.

(2) Occlusion: in the tracking process, the target is occluded, and the target is gradually occluded by the green leaves in the Coke video sequence, and Siamfo has already made a certain offset, and the target continues to move, but in the whole tracking process, compared with other comparative algorithms, this algorithm has good effect on the overall tracking of the target.

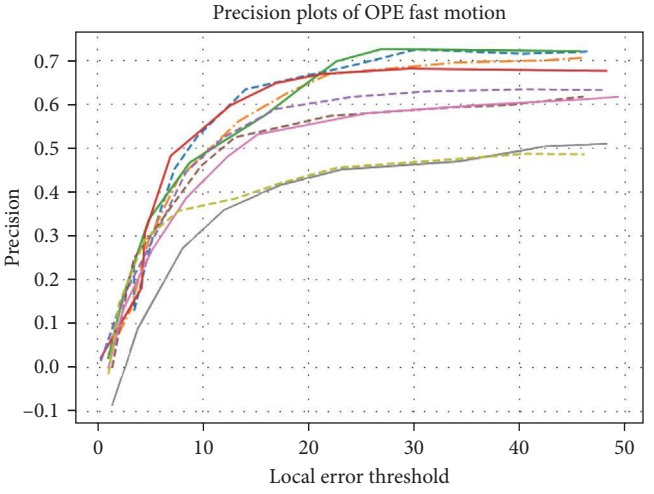
(3) Motion blur: due to the rapid motion of the target, it can bring problems such as image blur. In the Motor Rolling video sequence, the motorcycle is moving fast, which causes motion blur, and along with the target rotation and other challenges, the tracking is difficult. In about 32 frames, Siamfo and Cnet have lost the target, causing the subsequent tracking failure, but this algorithm and SiamRPN can achieve continuous tracking.



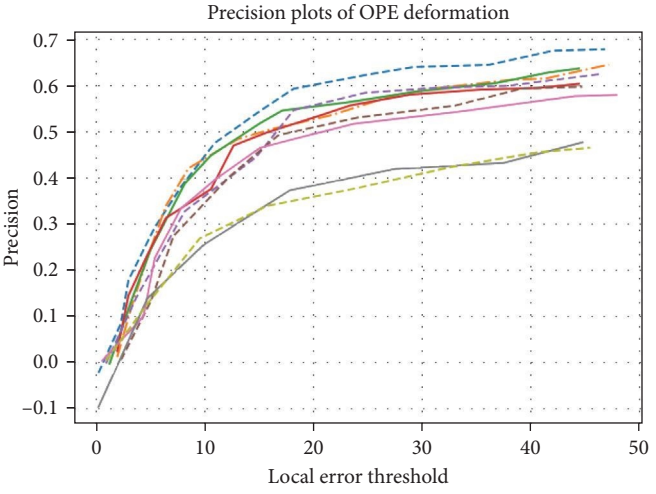
(a)



(b)



(c)



(d)

Figure 7: Continued.

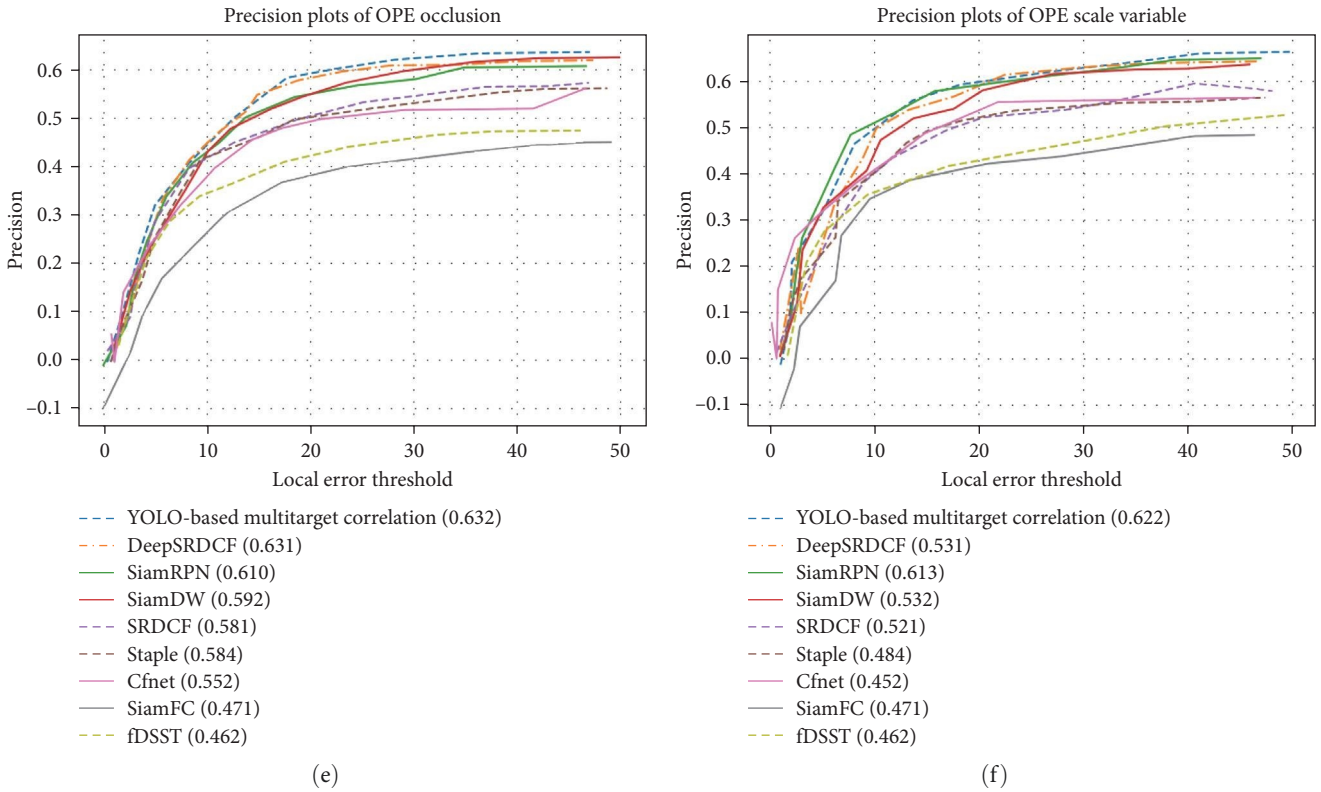


Figure 7: Comparison of accuracy rates of different algorithms under six types of challenges.

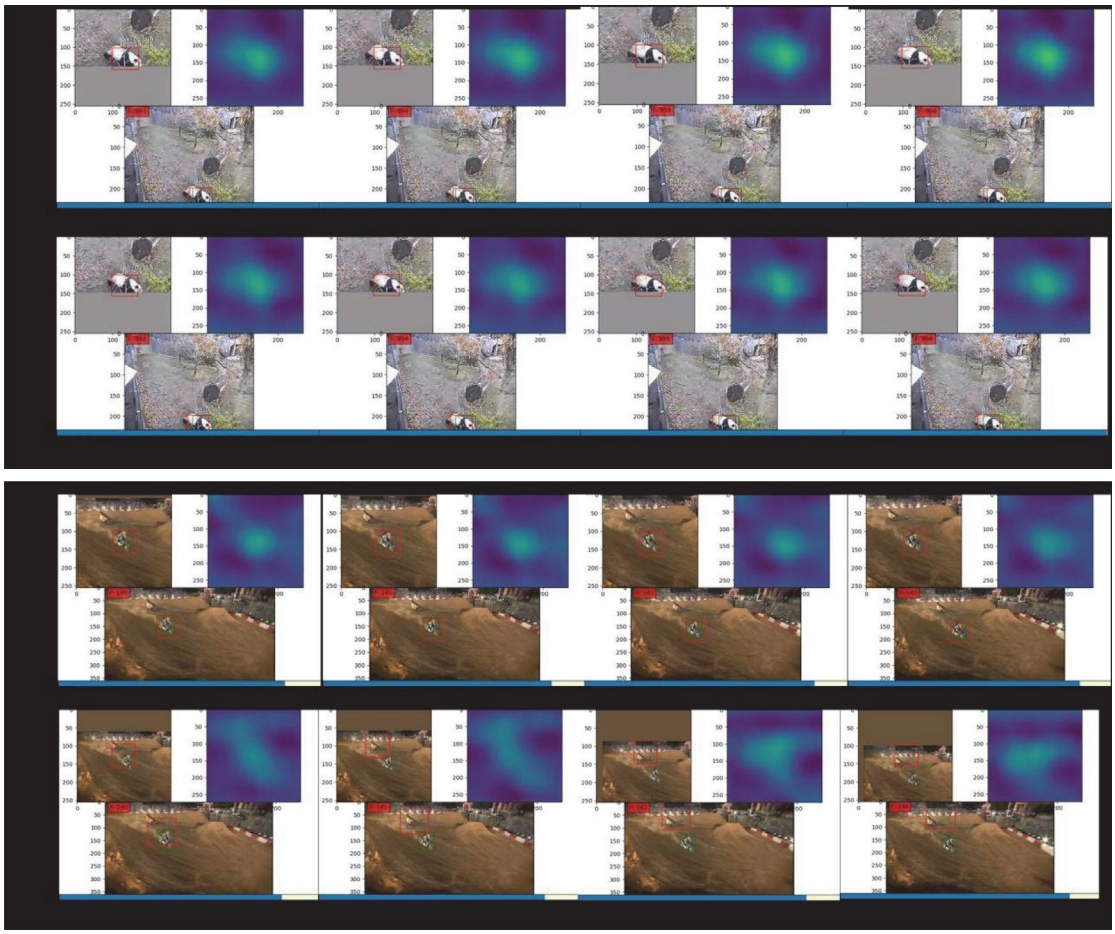


Figure 8: Tracking effects of various algorithms under different OTB2015 video sequences.

(4) Low resolution: when the resolution of image frames is low, the extracted features are not obvious. In Skiing, only the present algorithm and SiamRPN can achieve continuous tracking, and both Siamfo and Cnet lose the target at about 60 frames. Compared with SiamRPN, this algorithm has better tracking accuracy in low-resolution scenes, which is largely due to the deblurring effect of the video frames based on the adversarial network model.

(5) Scale change: in the tracking process, the target scale often changes, in the Carscale video sequence, as the car comes from far and near, the target becomes larger, compared with other comparison algorithms, this algorithm has better scale estimation results.

(6) Similar background interference: the interference of similar targets has always been one of the difficult problems in target tracking, especially in Football, where the tracking target moves faster on the one hand and the illumination changes drastically on the other hand, and there is the situation that the target is obscured. The target is obscured at around frame 289, and the tracking of the benchmark algorithm Siam is lost, while the algorithm in this paper achieves a continuous and stable tracking of the target with the multilayer feature increment.

4.2.3. Quantitative Analysis of VOT2018. The visual object tracking (VOT) is a challenge for single target tracking. VOT2018 has a total of 60 finely labeled short-time tracking video sets, and the evaluation metrics are more refined. VOT2018 has more complex target changes in the tracking sequence compared to the OTB2015, and the tracking difficulty is higher.

As shown in Figure 9, this algorithm is compared with the other seven algorithms in the VOT2018 dataset at baseline. Table 1 shows that the average expected overlap rate EAO and accuracy A of this algorithm are only lower than SiamRPN, but the robustness R is better than SiamRPN, where the higher the accuracy is, the lower the robustness is, and the higher the expected average overlap rate is. Compared with the benchmark algorithm Siamfo, the EAO of this algorithm is improved by 16.4% points. At the same time, the running speed is 39 frames per second, which further proves that the algorithm is robust and meets the real-time requirements and can achieve good tracking results.

4.2.4. Qualitative Analysis of VOT2018. Five video sequences are selected on the VOT208 dataset for quantitative analysis to prove that this algorithm outperforms SiamRPN and other algorithms for small target tracking and fuzzy target tracking. The test results are shown in Figure 10, red is the algorithm of this paper, purple, blue and green are SiamFC, SiamRPN, and KCF [26] algorithms respectively, and cyan is the annotation result of VOT2018 dataset.

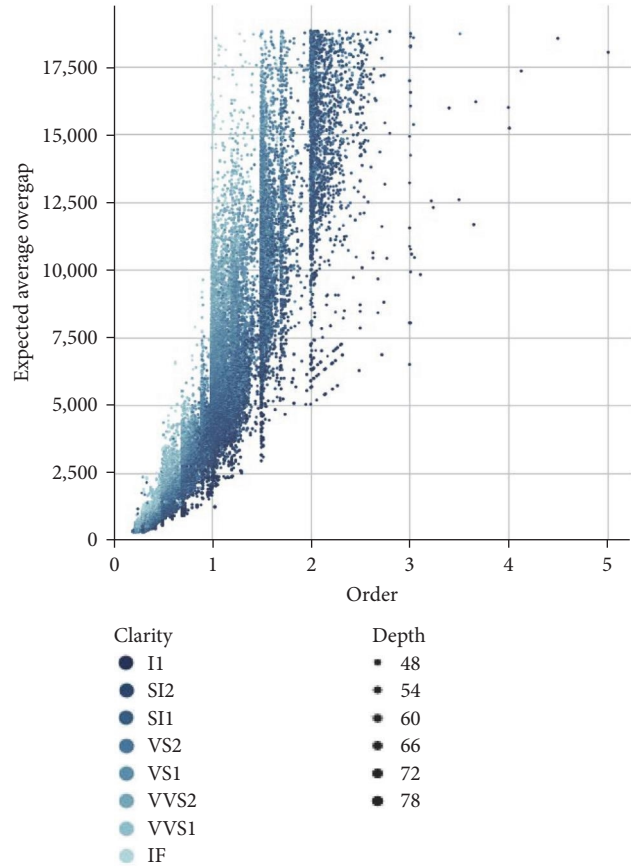


Figure 9: Comparison of EAOs of different algorithms on VOT2018 dataset.

In the birds1 sequence, on the one hand, the tracking object of the sequence is a small target, on the other hand, the image is blurred and the target features are not obvious, the algorithm in this paper can still effectively track the target, and compared with other algorithms, the algorithm in this paper has more overlap with the self-labeled results of VOT. In the basketball video sequence, the image is blurred and there is interference from similar targets, in about 265 frames, there are already tracking anomalies in the algorithm. In the bicycle moto-X sequence, other algorithms have been unable to track the overall characteristics of the target effectively due to target rotation and appearance changes, e.g., frames 37 and 69 of bicycle moto-X. In the soccer sequence, due to the blurred image, it is easy to interfere with the tracker, and in frame 115, the target is obscured, and Siamfo has a tracking loss. In the fernando sequence, the tracking is difficult due to the illumination change and the occurrence of occlusion, but the tracking effect is excellent compared with the benchmark algorithm [28, 29].

4.2.5. Ablation Experiments. Ablation experiments are conducted for the algorithm in this paper to analyze the effect of parameters. The dataset is OTB2015, and the experimental

Table 1: Comparison of test results of different algorithms on VOT2018 dataset.

Tracker	Accuracy	Robustness	EAO	Speed/(frame/s)
SiamRPN	0.586	0.276	0.383	59
YOLO-based multitarget correlation target detection algorithm	0.575	0.261	0.352	39
SASiam	0.566	0258	0.337	32
ECO	0.484	0.276	0.280	4
SiamFC	0.503	0.585	0.188	32
Staple	0.530	0.688	0.169	47
KCF	0.447	0.773	0.135	60
SRDCF	0.490	0.974	0.119	3

YOLO, you only look once.

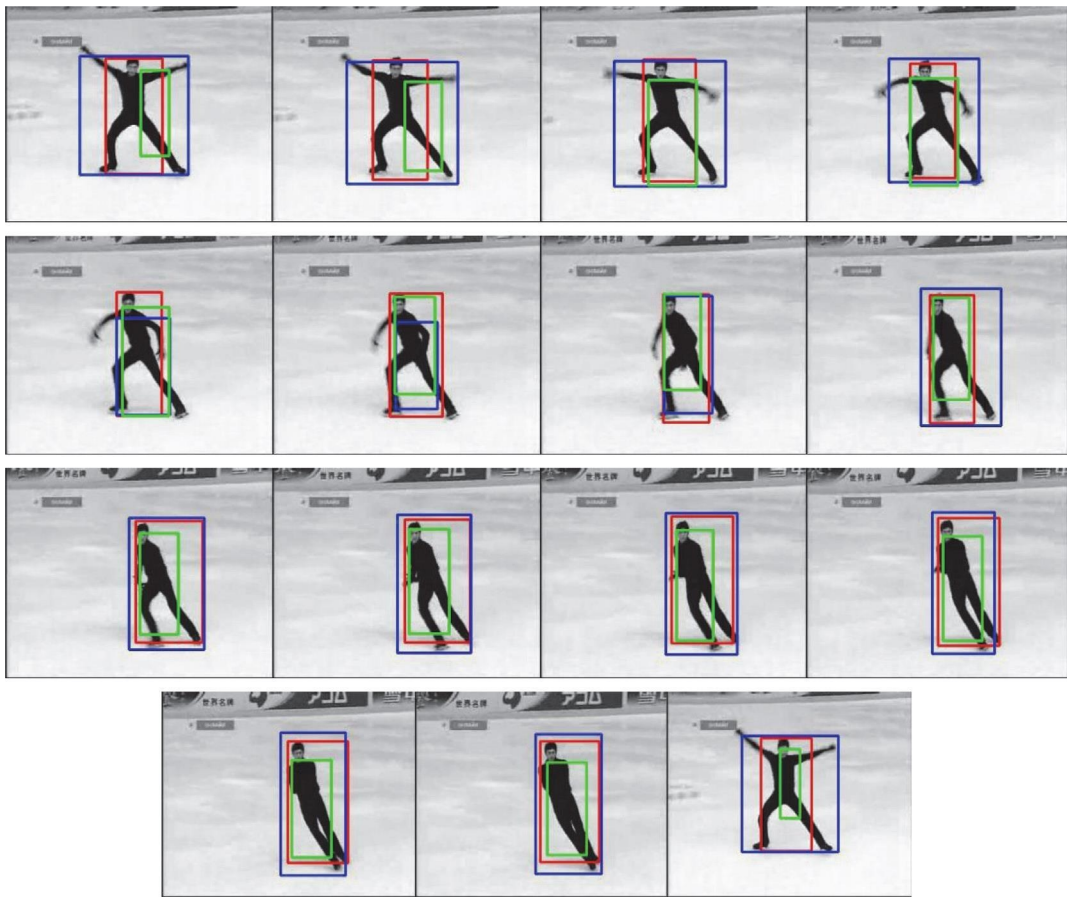


Figure 10: Tracking effect of selected VOT2018 video sequence.

results are shown in Figure 11. Where, YOLO-based multitarget correlation target detection algorithm, YOLO-based multitarget correlation target detection algorithm-VGG means the benchmark algorithm replaces only the backbone network as VGG-19 and fuzes the hierarchical features, YOLO-based multitarget correlation target detection algorithm-DeblurGAN means the DeblurGAN model for fuzzy removal is added to the benchmark algorithm, and YOLO-based multitarget correlation target detection algorithm—CGAN means the typical adversarial generative network CGAN model is added. From Figure 11, we can see that the improvement strategies such as conditional adversarial network and fusion of multilayer features are effective in improving the performance of the original algorithm, and the DeblurGAN model improves the performance of the algorithm more significantly than CGAN [30].

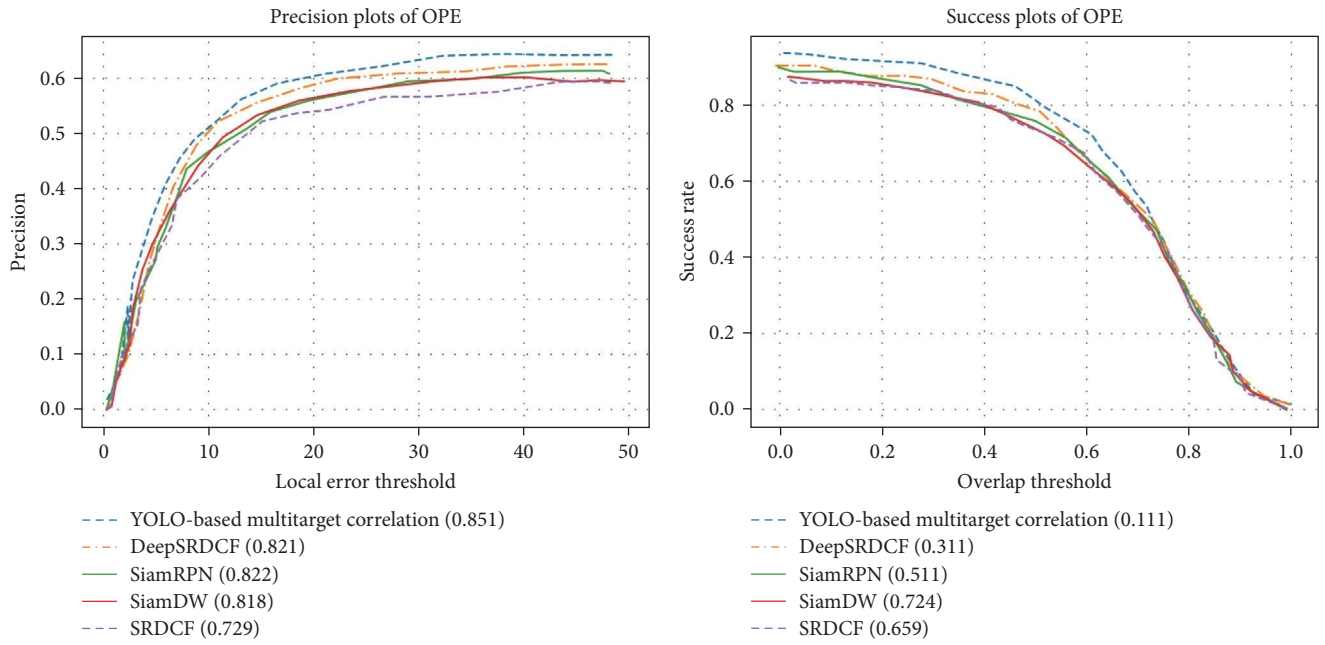


Figure 11: Impact of key algorithm links on tracking performance.

5. Conclusion

In this paper, a multitarget tracking algorithm based on GANs is proposed to address the problems of target loss, identity exchange, and jumping easily in the case of complex background, target occlusion, target scale and pose change in multitarget tracking. By using the YOLO-based multitarget association algorithm to detect the target to be detected in the current frame, a feature extraction model based on GANs is proposed, and target features are introduced to make the feature representation of the target more robust. The experimental results show that the algorithm can track the target smoothly and accurately in the presence of interference such as complex background, target occlusion, and scale change, and the occurrence of target identity jump is significantly reduced with high accuracy.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declared that they have no conflicts of interest regarding this work.

Authors' Contributions

All authors reviewed the results, approved the final version of the manuscript, and agreed to publish it.

Acknowledgments

The authors would like to show sincere thanks to those techniques who have contributed to this research.

References

[1] P. Wang, H. Fu, X. Li, J. Guo, Z. Lv, and R. Di, "Multi-feature fusion tracking algorithm based on generative compression network," *Future Generation Computer Systems*, vol. 124, pp. 206–214, 2021.

[2] A. Chaabane, E. De Cristofaro, M. A. Kaafar, and E. Uzun, "Privacy in content-oriented networking: threats and countermeasures," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 3, pp. 25–33, 2013.

[3] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

[4] Z. Zhang, X. Pan, S. Jiang, and P. Zhao, "High-quality face image generation based on generative adversarial networks," *Journal of Visual Communication and Image Representation*, vol. 71, Article ID 102719, 2020.

[5] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: a comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.

[6] D. Wu, C. Zhang, L. Ji, R. Ran, H. Wu, and Y. Xu, "Forest fire recognition based on feature extraction from multi-view images," *Traitement du Signal*, vol. 38, no. 3, pp. 775–783, 2021.

[7] H. Yan, Q. Hua, Y. Wang, W. Wei, and M. Imran, "Cloud robotics in smart manufacturing environments: challenges and countermeasures," *Computers & Electrical Engineering*, vol. 63, pp. 56–65, 2017.

[8] H. A. Khattak, M. A. Shah, S. Khan, I. Ali, and M. Imran, "Perception layer security in Internet of things," *Future Generation Computer Systems*, vol. 100, pp. 144–164, 2019.

[9] D. Rupprecht, A. Dabrowski, T. Holz, E. Weippl, and C. Pöpper, "On security research towards future mobile network generations," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2518–2542, 2018.

[10] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[11] Z. Cui, M. Zhang, Z. Cao, and C. Cao, "Image data augmentation for SAR sensor via generative adversarial nets," *IEEE Access*, vol. 7, pp. 42255–42268, 2019.

[12] X.-R. Tong, "Modeling and realization of real time electronic countermeasure simulation system based on SystemVue," *Defence Technology*, vol. 16, no. 2, pp. 470–486, 2020.

[13] Y. Sha, Z. Chen, X. Liu et al., "Adaptive industrial control system attack sample expansion algorithm based on generative adversarial network," *Applied Sciences*, vol. 12, no. 17, Article ID 8889, 2022.

[14] S. T. Ali, P. McCorry, P. Hyun-Jeen Lee, and F. Hao, "ZombieCoin 2.0: managing next-generation botnets using Bitcoin," *International Journal of Information Security*, vol. 17, pp. 411–422, 2018.

[15] Z. Wang, L. Liu, C. Wang et al., "Data enhancement of underwater high-speed vehicle echo signals based on improved generative adversarial networks," *Electronics*, vol. 11, no. 15, Article ID 2310, 2022.

[16] F. Bi, Z. Man, Y. Xia et al., "Improvement and application of generative adversarial networks algorithm based on transfer learning," *Mathematical Problems in Engineering*, vol. 2020, Article ID 9453586, 11 pages, 2020.

[17] G. Zin, *Generative adversarial networks for online visual object tracking systems*, Theses and Dissertations (Comprehensive). 2196, 2019.

[18] B. Yao, J. Li, S. Xue et al., "GARAT: generative adversarial learning for robust and accurate tracking," *Neural Networks*, vol. 148, pp. 206–218, 2022.

[19] J. Zhou, D. Zhang, W. Ren, and W. Zhang, "Auto color correction of underwater images utilizing depth information," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[20] J. Wang, "Image restoration on residual aggregation network in poor weather condition," in *2020 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, pp. 137–142, IEEE, Southend, UK, 2020.

[21] Z. Huang, J. Zhan, H. Zhao, K. Lin, P. Zheng, and J. Lv, "Real-time visual tracking base on SiamRPN with generalized intersection over union," in *Advances in Brain Inspired Cognitive Systems*, pp. 96–105, Springer, Cham, 2020.

[22] X. Gao, S. Wang, Y. Cui, and Z. Wu, "Aero-optical image and video restoration based on mean filter and adversarial network," in *2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA)*, pp. 528–532, IEEE, Changchun, China, 2022.

[23] S. T. Ali, P. McCorry, P. Hyun-Jeen Lee, and F. Hao, "ZombieCoin: powering next-generation botnets with bitcoin," in *Financial Cryptography and Data Security*, M. Brenner, N. Christin, B. Johnson, and K. Rohloff, Eds., pp. 34–48, Springer, Berlin, Heidelberg, 2015.

[24] P. Sinha, V. K. Jha, A. K. Rai, and B. Bhushan, "Security vulnerabilities, attacks and countermeasures in wireless sensor networks at various layers of OSI reference model: a survey," in *2017 International Conference on Signal Processing and Communication (ICSPC)*, pp. 288–293, IEEE, Coimbatore, India, 2017.

[25] S. Zhang and Y. Song, "Object Detection in unmanned vehicle with end-to end edge-enhanced GAN and object detector network," in *2021 33rd Chinese Control and Decision Conference (CCDC)*, pp. 7477–7482, IEEE, Kunming, China, 2021.

[26] H. Zhang and H. Hua, "FusionGAN-detection: vehicle detection based on 3D-LIDAR and color camera data," in *Seventh Asia Pacific Conference on Optics Manufacture and 2021 International Forum of Young Scientists on Advanced Optical Manufacturing (APCOM and YSAOM*, pp. 347–352, SPIE, 2022.

[27] S. Liu, Y. Yao, and B. Liu, "Saliency target detection in complex video environment based on generation countermeasure network," in *Second IYSF Academic Symposium on Artificial Intelligence and Computer Engineering*, pp. 524–529, SPIE, 2021.

[28] J. Shi, X. Zhuo, C. Zhang, Y. X. Bian, and H. Shen, "Research on key technologies of underwater target detection," in *Seventh Symposium on Novel Photoelectronic Detection Technology and Applications*, pp. 1128–1137, SPIE, 2021.

[29] S. Pisharody, J. Natarajan, A. Chowdhary, A. Alshalan, and D. Huang, "Brew: A security policy analysis framework for distributed SDN-based cloud environments," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 6, pp. 1011–1025, 2019.

[30] J. Zhou, J. Sun, W. Zhang, and Z. Lin, "Multi-view underwater image enhancement method via embedded fusion mechanism," *Engineering Applications of Artificial Intelligence*, vol. 121, Article ID 105946, 2023.