Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8885670, 7 pages
https://doi.org/10.1155/2020/8885670

Research Article

Algorithm for Target Detection in Smart City Combined with Depth Learning and Feature Extraction

Feng Wang[1], **Zhiming Xu**[1], **Zemin Qiu**[1], **Weichuan Ni**[2], **Jiaqi Li**[1] and **YiLan Luo**[1]

[1]*Department of Information Science, Xinhua College of Sun Yat-Sen University, Guangzhou, China*
[2]*Department of Equipment and Laboratory Management, Xinhua College of Sun Yat-Sen University, Guangzhou, China*

Correspondence should be addressed to Feng Wang; iswf@xhsysu.edu.cn

Received 10 April 2020; Revised 3 August 2020; Accepted 20 September 2020; Published 5 October 2020

Academic Editor: Wei Wang

The target detection algorithms have the problems of low detection accuracy and susceptibility to occlusion in existing smart cities. In response to this phenomenon, this paper presents an algorithm for target detection in a smart city combined with depth learning and feature extraction. It proposes an adaptive strategy is introduced to optimize the algorithm search windows based on the traditional SSD algorithm, which according to the target operating conditions change, strengthening the algorithm to enhance the accuracy of the objective function which is combined with the weighted correlation feature fusion method, and this method is a combination of appearance depth features and depth features. Experimental results show that this algorithm has a better antiblocking ability and detection accuracy compared with the conventional SSD algorithms. In addition, it has better stability in a changing environment.

1. Introduction

The concept of a smart city originated from the idea of smart earth proposed by IBM in 2008. Among them, the target detection algorithm is one of the key technologies of smart cities. However, existing computer vision algorithms are difficult to deal with target detection problems in complex backgrounds, such as the effects of light, target size changes, and target occlusion. The introduction of deep learning has opened up a new path for target detection. In recent years, more and more researchers have begun to conduct in-depth research on deep learning algorithms in target detection [1, 2]. Deep learning avoids the drawbacks of the traditional method of manually extracting features, because of the characteristics that its deep structure can effectively learn from large amounts of data [3, 4]. Currently, based on the target detection algorithm, the depth study of literature is not much. However, from the perspective of the depth model, it can be broadly classified into a target detection algorithm based on CNN and a target detection algorithm based on SAE [5–7]. The target detection algorithm of the SAE depth model is usually combined with the traditional classical algorithms. It uses hidden layers to learn a representation of data and to preserve and better obtain more efficient information by using a nonlinear feature extraction method that does not use classification tags. This method is not conducive to information classification, but visual tracking itself needs to distinguish the target from the background. Therefore, target tracking is not the strength of the SAE algorithm. The target detection algorithm based on CNN combines artificial neural networks and convolution operations. It can recognize a wide variety of target modes, and a certain degree of distortion and deformation has good robustness. Therefore, we use the target detection algorithm based on CNN for this article. Among them, the SSD algorithm recognition structure is superior to a similar algorithm in the mAP and training speed [8, 9]. By using different sizes and different proportions of anchors at different levels, the algorithm can find the best matching anchor with ground truth for training. However, the recognition effect of the target on the small size is relatively poor and is easily affected by the occluder, which undoubtedly affects the application of the algorithm in practical applications.

Regarding the issue above, this article proposes a target detection method based on the SSD algorithm and feature extraction fusion. It is based on the traditional SSD algorithm [10, 11]. Its search window is dynamically adjusted by using an adaptive strategy changes according to its operating conditions, which can reduce unnecessary calculation accuracy problems during the entire detection target fixed occurring. At the same time, in order to improve the classification ability of the features of the algorithm, we did the following optimization; like in the feature fusion method of weighted correlation, we combined with the appearance depth feature and the motion depth feature to improve the accuracy of the objective function and perform experiments in different complexity image environments. This algorithm is more time-consuming than the traditional SSD algorithm. The accuracy of the target gradually increases with the complexity of the image. The gap between the detection accuracy and success rate and the traditional SSD algorithm gradually widens and remains at about 86%. It is said that the algorithm can be applied to a variety of environments and maintain good stability. It has good practical value in the development of a smart city.

2. Principle of Algorithm

Traditional SSD is based on a forward-propagating CNN network. It produces a series of fixed-size bounding boxes and has the possibility of containing object instances in each box then performs a nonmaximal suppression to get the final predictions. The model network structure is as follows.

From the structure diagram of Figure 1, we can get the SSD network can be divided into two parts: the basic network and additional functional layers; the former is used for the standard network for image classification, but all the layers involved in the classification are eliminated; the latter mainly achieves the following goals [12]:

Multiscale feature maps for detection: the convolutional feature layer is added to obtain feature layers of different scales so as to achieve multiscale target detection.

Convolutional predictors for detection: for each added feature layer, a set of convolution filters is used to obtain a fixed set of target detection predictions.

Each of these convolutions results in a set of scores or coordinate offsets from the default candidate regions. Finally, combining the obtained detection and classification results, the position of each object in the image and the object category in the image can be obtained.

3. The Algorithm of This Paper

3.1. Select the Aspect Ratio of the Default Box. The feature map will be smaller and smaller at deeper layers. This is not only to reduce the computational and memory requirements but also has the advantage that the last extracted feature map will have some degree of translation and scale invariance [13].

In the SSD structure, the default boxes do not necessarily correspond to the receptive fields of each layer. Predictions are made by introducing m feature maps; the size calculation formula of the default box in each feature map satisfies:

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m-1}(k-1), \tag{1}$$

where k is $[1, m]$, $s_{\min}$ value 0.2, and $s_{\max}$ value 0.95.

But usually, the size of each default box will not be adjusted after this calculation. This is undoubtedly unfavorable for detecting the object whose size will change and then affecting the detection effect. Figure 2 shows the structure of the target of the traditional SSD algorithm.

Among them, the objective loss function is a weighted sum of the localization lossloc and confidence loss in Figure 2:

$$L(x, c, l, g) = \frac{1}{N}\left(l_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g)\right). \tag{2}$$

In the formula:

$$\begin{aligned} L_{\text{loc}}(x, l, g) &= \sum_{i\in\text{Pos}}^{N} \sum_{m\in(cx,cy,w,h)} x_{ij}^{k}\text{smooth}\left(l_i^m - \widehat{g}_j^m\right), \\ L_{\text{conf}}(x, c) &= -\sum_{i\in\text{Pos}}^{N} x_{ij}^{p}\log\left(\widehat{c}_i^p\right) - \sum_{i\in\text{Neg}}\log\left(\widehat{c}_i^0\right), \end{aligned} \tag{3}$$

among them, $\widehat{c}_i^p = \exp\left(c_i^p\right)/\sum_p \exp(c_i^p)$, N is the number of matching default boxes, x indicates whether the matched box belongs to category p, value {0,1}, l is a predictive box, and g is the true value of that ground truth box. c is the confidence that the selected target belongs to category p. Weight item α=1.

We need to filter out the boxes; we finally give from these boxes.

The pseudocode is as follows.

for every conv box:
for every class:
if class_prob < theshold:
continue
predictive box = decode(convbox)
nms(predictive box) # Remove very close boxes

In this way, the target coordinates can be found effectively, thereby improving the detection effect of the algorithm on target detection.

3.2. Adaptive Strategy. Usually, the width and height of each default boxes are fixed during the entire target detection process. However, when the behavior of the controlled object changes due to changes in the characteristics of the object, this type of parameter fixation tends to produce undesirable results. Control the effect, so you need to make use of adaptive strategies for dynamic adjustments.

In this paper, the width and height are adaptively adjusted by combining the size of the default box obtained above. The calculation formula is as follows:

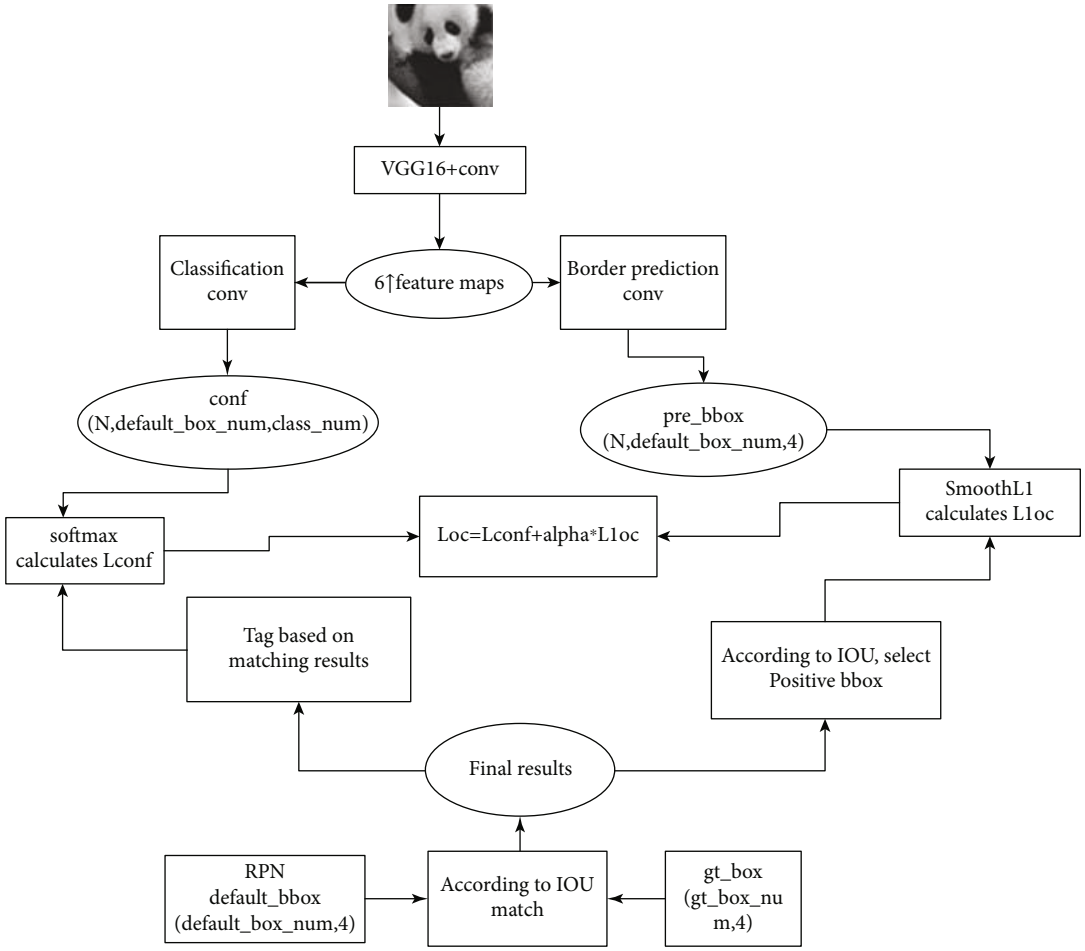

Figure 1: SSD model network structure.

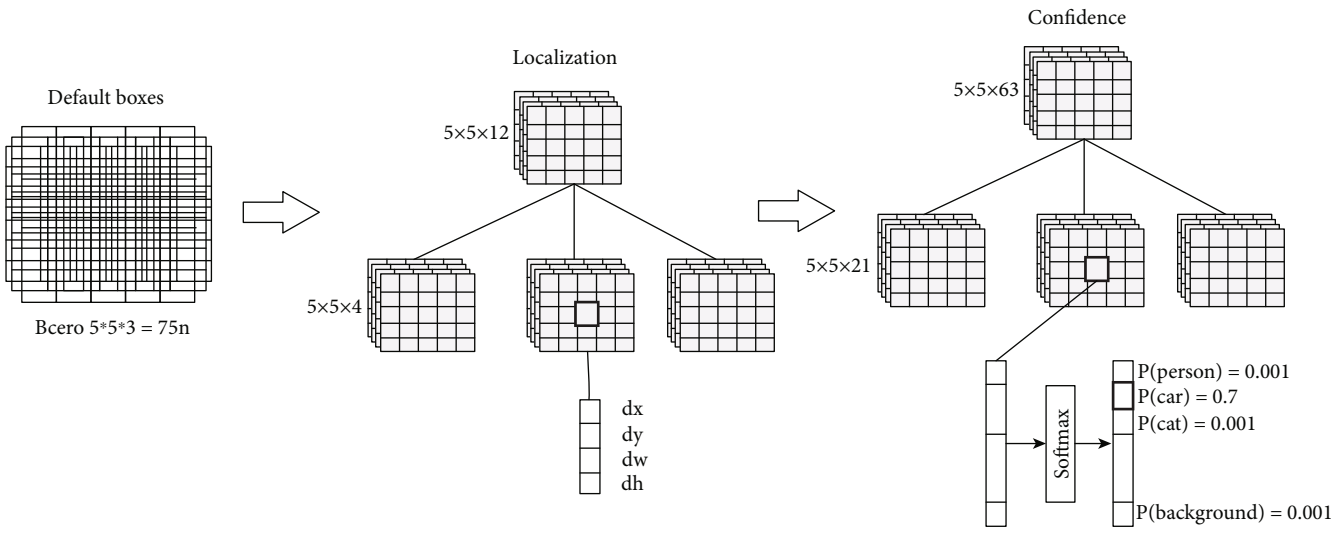

Figure 2: Block diagram of the traditional SSD algorithm target positioning.

$$L_w = \sqrt{\frac{\sum_{i=1}^{n} \boldsymbol{s}_{k_i}{}^2}{n}},$$
$$L_h = \sqrt{\frac{\sum_{j=1}^{n} s_{k_j}{}^2}{n}}, \quad (4)$$
$$L = \max\left(L_w, L_j\right),$$

where n is equal to the number of blocks in the image; s_{ki}, s_{kj} is the component of the horizontal direction i and the vertical direction j of the objective function, respectively; L_w, L_h is newly born into the width and height of the default boxes; and L is the whole frame of motion intensity.

By determining the motion complexity of the block based on the calculated motion vector and the motion vector of the current block, the degree of motion of the current block can be effectively judged by the degree of difference between the horizontal component and the vertical component. The formula is as follows:

$$S_{\mathrm{w}} = \mathrm{Max}\left[\mathrm{horizontal}(s_k) - \mathrm{horizontal}\left(s_{kii\mathrm{middle}}\right)\right],$$
$$S_h = \boldsymbol{Max}\left[\mathrm{vertical}(s_k) - \mathrm{vertical}\left(s_{kii\mathrm{middle}}\right)\right], \quad (5)$$
$$S = \mathrm{Max}\ (S_w, S_h).$$

Among them, $s_{kii\mathrm{middle}}$ is the median of the three macroblocks in the left, top, and upper right directions. $s_{ki}$is the objective function of the i-th block obtained above. S_w, S_h separately expressed the horizontal and vertical movement complexity. S represents the complexity of the motion of the current block.

The search window size is as follows:

$$W = \begin{cases} L & S < L \\ S + L & \text{other} \end{cases}. \quad (6)$$

3.3. Feature Extraction of Weighted Correlation. In order to improve the classification ability of features, a feature fusion method based on weighted correlation is used to combine the appearance depth feature and the movement depth feature to form a multidimensional feature vector. For the convenience of presentation, the appearance depth feature and the movement depth feature are denoted by y_1 and y_2, respectively. The merged feature y is:

$$y = (\omega_1 y_1, \omega_2 y_2) \quad (7)$$

Here, ω_i is the weighting factor, and $(\omega_1)^2 + (\omega_2)^2 = 1$. The weighting coefficients are determined according to intraclass consistency and class separability. Intraclass consistency: It is generally expected that the samples in the same class are as close as possible in the feature space. However, there is usually a large variance in the sample characteristics in the same class. Therefore, it is not necessary to require all samples in the same class to be close to each other. A tradeoff is to ensure that the samples in the same neighborhood within the same class are as close as possible.

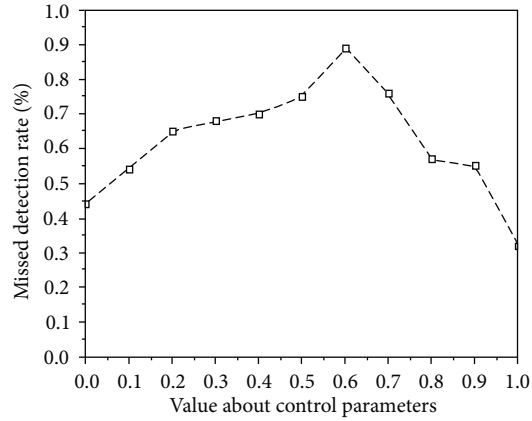

Figure 3: Control parameter λ_s value curve.

Assume $y_i = (\omega_1 y_1^i,\ \omega_2 y_2^i)$ and $y_j = (\omega_1 y_1^j,\ \omega_2 y_2^j)$.

Let us denote the i-th and j-th samples; then, the intraclass consistency is defined as:

$$S_C = \sum_{i=1}^{N} \sum_{j \in N_R(F_i)} \frac{< y_i, y_j >}{\|y_i\| \|y_j\|} = \sum_{i=1}^{N} \sum_{j \in N_R(F_i)} \frac{\sum_{k=1}^{2} w_k^2 y_i^k y_j^k}{\sqrt{\sum_{k=1}^{2} w_k^2 \left(y_i^k\right)^2} \sqrt{\sum_{k=1}^{2} w_k^2 \left(y_j^k\right)^2}}. \quad (8)$$

In the formula, $N_R(F_i)$ means the sample F_i and with F_i which belongs to the index set of the k nearest neighbor samples of the same class.

According to the target characteristics with good intraclass consistency, this paper determines the weighting coefficients by solving the following optimization problems:

$$\max\ \{S_C + \lambda_s \|\omega\|\}, \quad (9)$$

among them, $\omega_k > 0$, $\|\omega\| = 1$, and λ_s is a control parameter.

Combining the above equations, the gradient descent method is used to solve the equation, which can be solved:

$$\omega_k(\mathrm{t}+1) = \omega_k(\mathrm{t}) + \eta \left.\frac{\partial \mathrm{L}}{\partial \omega_{\mathrm{k}}}\right|_{\omega_{\mathrm{k}} = \omega_{\mathrm{k}}(t)}, \quad (10)$$

where t is the number of iterations, η is the iteration step, and $L(S_C, \omega) = S_C + \lambda_s \|\omega\|$ is the objective function.

Among them

$$\frac{\partial L(S_C, \omega)}{\partial \omega_k} = \sum_{i=1}^{N} \sum_{j \in N_r(x_i)} \frac{\partial h_{ij}(\omega)}{\partial \omega_k} + \lambda_s \omega_k, \quad (11)$$

$h_{ij} = \sum_{k=1}^{2} \omega_k^2 y_i^k y_j^k / \sqrt{\sum_{k=1}^{2} \omega_k^2 \left(y_i^k\right)^2} \sqrt{\sum_{k=1}^{2} \omega_k^2 \left(y_j^k\right)^2}$.

Thus, you can get the final objective function:

$$L = L(x, c, l, g) + L(S_C, \omega) \quad (12)$$

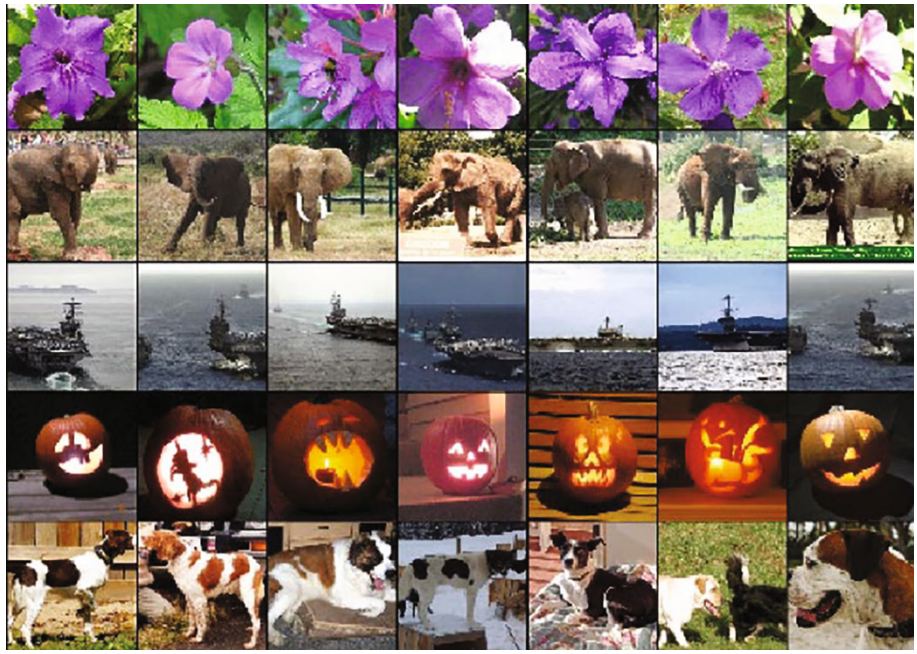

Figure 4: ImageNet datasets.

Table 1: Time spent in training and testing.

Method	Time spent on testing/s		Training period time spent/s	
	Evaluation window time taken	Target retrieval time	Evaluation window time taken	Target retrieval time
Traditional SSD algorithm	0.35	2.11	0.25	3.63
Literature [14] algorithm	0.30	1.92	0.18	2.51
Article algorithm	0.31	2.01	0.20	2.98

After many experiments, the summary data is obtained in Figure 3; we can see the experiments have found that blindly increasing the control parameters does not have any new improvement in the detection accuracy. It is appropriate when the number of key region control parameters is $\lambda_s = 0.6$. This ensures that a unique solution that can converge to a global optimum through gradient descent can be guaranteed.

4. Simulation Experiment

4.1. Data Sets and Test Standards. In order to test the speed and accuracy of the algorithm, the training data from this paper comes from the ImageNet dataset, which contains more than 14 million pictures, covering more than 20,000 categories [15], like Figure 4. The ImageNet dataset is a field that is currently applied in the field of deep learning images. Most research work on image classification, positioning, and detection is based on this dataset. It is a huge image library for image/visual training. It has been widely used in research papers in the field of computer vision and has almost become the "standard" data set for the performance testing of algorithms in the field of deep learning images.

4.2. Experiments and Results. The experiment was simulated using a laptop computer, tested using Python 3.6, TensorFlow v0.12.0, Pickle, OpenCV-Python, and Matplotlib (optional), and the data was analyzed using MATLAB 2014a. In order to detect the target detection effect of the algorithm, this paper compares the traditional SSD algorithm with the literature algorithm [14].

In order to test the retrieval efficiency of the algorithm, the data analysis is performed on the time window of the evaluation of the image and the time required for the detection of the target. Observing Table 1, we can see that in the testing phase, the evaluation window of this algorithm is more time-consuming than the traditional SSD. In algorithm 0.04s, the target detection time is better than the traditional SSD algorithm 0.1s, and in the training phase, the evaluation time of the algorithm in this paper is better than the traditional SSD algorithm 0.05s, and the target detection time is better than the traditional SSD algorithm 0.65s.

From Table 1, it can be seen that the algorithm is time-consuming in the test phase and the training phase is better than the traditional SSD algorithm, but the text is slightly weaker than the literature algorithm [14]. In addition to focusing on the time-consuming, the target detection algorithm needs to analyze and evaluate the accuracy and success rate of the detection target. Therefore, this paper detects the precision and success rate of the target and in different complex scenarios. The test was conducted, in which the selected image material was reconstructed from low to high (food, vegetable, bird, person). The data is shown in Table 2.

Table 2: Algorithm accuracy data tables in different complex scenarios.

Image	Accuracy/%	Traditional SSD algorithm	Literature algorithm [14]	Article algorithm
Food	Accuracy rate	0.848	0.875	0.874
	Success rate	0.854	0.859	0.859
Vegetable	Accuracy rate	0.795	0.804	0.854
	Success rate	0.805	0.814	0.866
Bird	Accuracy rate	0.757	0.750	0.879
	Success rate	0.754	0.771	0.862
Person	Accuracy rate	0.711	0.708	0.852
	Success rate	0.712	0.705	0.871

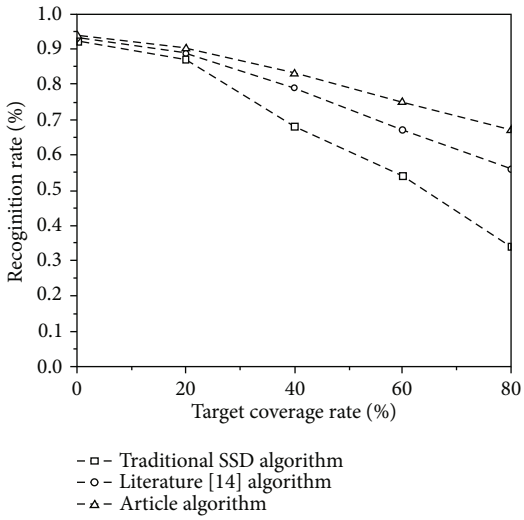

Figure 5: Food cover experiment.

As can be seen from Table 2, it is clear that the target detection accuracy of this algorithm in different environments is better than the traditional SSD algorithm, and as the complexity of the image is gradually increased, the accuracy of detection under the image and the success rate gap gradually increase; the detection accuracy rate is maintained at about 86%, while in the literature algorithm [14] gradually decreases as the complexity increases, and the target detection effect in a variety of complex scenarios embodies the algorithm and has a certain degree of universality.

In order to test whether the algorithm has accuracy in the presence of a shelter, this paper needs to use two sets of experiments to verify. One group adopts a food image with smaller image complexity, and one group employs a person image with the highest complexity; data acquisition is performed using the above algorithms, respectively, and the data image is shown.

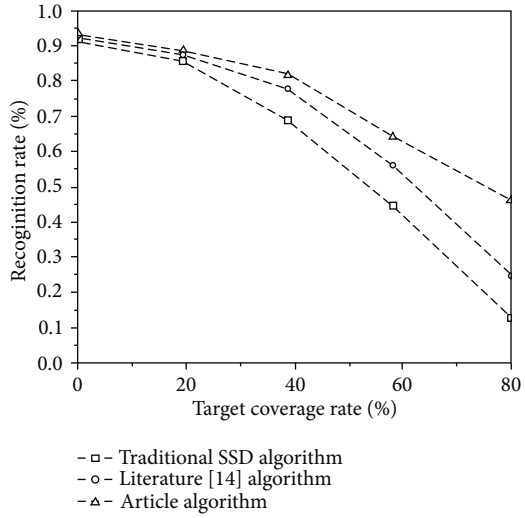

Figure 6: Person cover experiment.

Figure 7: Target extraction image.

By observing Figures 5 and 6, we can see that with the increase of target coverage rate, the setting rate of each algorithm gradually decreases, but it can be seen through observation that the data collected by using this algorithm are kept above the other two algorithms. When the target coverage rate reaches 60% and higher, the setting rates of both the algorithm [14] and the traditional SSD algorithm begin to decline drastically. However, although there is a problem of drop in the article algorithm, the decrease rate of the algorithm is slower than that of the other two methods, which effectively proves the feasibility of the feature extraction.

In order to test the detection effect of the algorithm in the actual environment, we have selected the daily traffic scene for the target detection, real-time extraction of the people, and vehicles appearing in the traffic; the following effect map is obtained.

By observing Figures 7 and 8, we can find that it can be seen from the observation that the algorithm can accurately retrieve the target in the complicated traffic area and track and identify it. Although the process will be blocked by vehicles or pedestrians, there is still no problem that is currently lost, which effectively validates the feasibility of the algorithm.

Figure 8: Target extraction image 2.

5. Conclusion

We propose a target detection method based on the SSD algorithm and feature extraction fusion. The algorithm is based on the traditional SSD algorithm. The algorithm adopts an adaptive strategy to dynamically adjust the search window according to the change of the running status of the image and combines the appearance depth feature and the movement depth feature in combination with the feature fusion method of weighted correlation, and finally improves the precision extraction of the objective function. Through experiments, the algorithm can maintain a high and stable target detection effect under different complexity of the image environment and is more suitable for the environment changeable target detection environment. However, it still cannot be effectively reduced in the time-consuming aspect of the algorithm. This will serve as a research focus in the future and will be further studied.

Data Availability

The data used to support the results of this study needs to be obtained with the consent of the corresponding author.

Conflicts of Interest

The authors declare that they have no competing interests.

Authors' Contributions

The authors have equally contributed to the manuscript. All authors read and approved the final manuscript.

Acknowledgments

This article is supported by Sun Yat-Sen University Xinhua College 2017 School-Level Scientific Research Startup Fund General Project: Research and Design of Target Trajectory Tracking System Based on Depth and Visual Information Fusion (Project Code: 2017YB001).

References

[1] L. Fan, L. Pengyuan, and L. Bing, "Deep learning model design of video target tracking based on tensor flow platform," *Laser&Optoelectronics Progress*, vol. 9, no. 15, pp. 277–285, 2017.

[2] G. Hao, X. Xiang-Yang, and A. Zhi-Yong, "Advances on application of deep learning for video object tracking," *Acta Automatica Sinica*, vol. 42, no. 6, pp. 834–847, 2016.

[3] Y. Zheng, C. Quanqi, and Z. Yujin, "Deep learning and its new progress in object and behavior recognition," *Journal of Image and Graphics*, vol. 19, no. 2, pp. 175–184, 2014.

[4] S. Wu, S. Wang, R. Laganiere, C. Liu, H. S. Wong, and Y. Xu, "Exploiting target data to learn deep convolutional networks for scene-adapted human detection," *Ieee Transactions On Image Processing*, vol. 27, no. 3, pp. 1418–1432, 2018.

[5] C. Shiyu, L. Yuehu, and L. Xinzhao, "Vehicle detection method based on fast R-CNN," *Journal of Image and Graphics*, vol. 22, no. 5, pp. 671–677, 2017.

[6] Z. Guangjun, W. Xuchu, N. Yanmin, T. Liwen, and Z. Shaoxiang, "Deep SAE feature learning based segmentation for digital human brain image," *Journal of Computer-Aided Design & Computer Graphics*, vol. 28, no. 8, pp. 1297–1305, 2016.

[7] X. Y. Qian, L. Han, Y. Wang, and M. Ding, "Deep learning assisted robust visual tracking with adaptive particle filtering," *Signal Processing-Image Communication*, vol. 60, no. 1, pp. 183–192, 2018.

[8] Z. Tang, H. Wu, W. Wang, J. Wei, and T. Huang, "self-adaptive SSD caching system for multiobjective optimization in virtualization environment," *Journal of Software.*, vol. 28, no. 8, pp. 1982–1998, 2017.

[9] W. Jiewen, Z. Yinwei, L. Weilin, and G. Canzhang, "Batch renormalization of real-time object detection algorithm YOLO," *Application Research of Computers*, vol. 35, no. 11, pp. 1–9, 2018.

[10] Y. Han and H. Hahn, "Visual tracking of a moving target using active contour based SSD algorithm," *ROBOTICS AND AUTONOMOUS SYSTEMS.*, vol. 53, no. 3-4, pp. 265–281, 2005.

[11] H. U. Yin and Y. A. N. G. Jing-yu, "Tracking algorithm based on fusion of SSD and MCD robust to partial occlusion," *Journal of System Simulation*, vol. 22, no. 4, pp. 908–911, 2010.

[12] H. Song, X. Zhang, B. Zheng, and T. Yan, "Vehicle detection based on deep learning in complex scene," *Application Research of Computers*, vol. 35, no. 4, pp. 1–5, 2018.

[13] W. Liu, D. Anguelov, D. Erhan et al., *SSD: single shot multi box detector*, Springer International Publishing, 2016.

[14] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proceedings of the 32nd International Confernce on Mackine Learning*, pp. 597–606, 2015.

[15] "Imagenet dataset [DB/OL]," 2020, http://www.image-net.org/download-imageurls.