

Research Article

Target Tracking Algorithm in Football Match Video Based on Deep Learning

Wei Zhao 

School of Physical Education, Shaanxi University of Technology, Hanzhong 723000, China

Correspondence should be addressed to Wei Zhao; zhaowei@snut.edu.cn

Received 26 April 2022; Accepted 23 May 2022; Published 31 May 2022

Academic Editor: Wen-Tsao Pan

Copyright © 2022 Wei Zhao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

After the introduction of online learning mechanism, the traditional target tracking algorithm in football game video based on TLD has good tracking ability, but it will lose the target when the target is seriously obscured. Therefore, a soccer game video target tracking algorithm based on deep learning is proposed. The target detection algorithm of GoogLeNet-LSTM is used for feature mapping to obtain the feature mapping array. After processing, a high reliability candidate box for training and matching is obtained, and the feature maps of the detection results are collected to obtain the depth features required for tracking. Scale space discriminant tracking algorithm and Markov Monte Carlo algorithm are used to track single target or multi-target, respectively. Experimental results show that the average frame rate of the algorithm is maintained above 35 Hz, and the tracking time is about 12.5 s. The average center position deviation index is 39, the average coverage index is 40, and the resource utilization is low. The algorithm can track the target in the football game video well.

1. Introduction

In modern life, sports video is the important video that is popular among the majority of viewers. It has a large proportion in existing TV programs and the Internet [1, 2]. With the continuous improvement of people's quality of life and the rapid advancement of technology, people's demands for sports video are also rising. In terms of sports competition viewing, passive and flat viewing methods will gradually fail to meet the requirements of TV viewers [3–5]. The broadcasters need to add various visual effects to meet the visual requirements of the audience. In terms of game analysis, the team coach needs to extract relevant data from the football game video to assist the coach in researching the tactics. In terms of commercial applications, the broadcasters also need to more fully explore the commercial value of the football game broadcast. These need to analyze the video data of the game and process the game images for different requirements in order to meet the requirements of the game video [6–8]. Among the many sports competitions, football matches have the largest number of viewers and the highest level of attention. Therefore, the detection,

extraction, location, and tracking of moving targets in the video have high practical and practical significance.

The extraction and tracking of targets in the football game video are hot spots in the current sports field image and video processing. The technology required covers many areas of image processing analysis and computer vision. In general, football match video scene consists of background and goal, where the goal is the important part of the video, which contains important information. Therefore, quickly and efficiently segmenting objects in video and tracking the target of interest are the bases for subsequent image analysis [9].

Although the research of target tracking has made great progress and breakthrough to some extent, the robust target tracking algorithm has been full of challenges due to the complexity of the environment and the influence of target deformation. The core problem of target tracking is feature representation. The early features are manually selected, and the appropriate features are selected according to the different application scenarios, but the effect is far from meeting the actual needs. Since the advent of deep learning technology, the field of computer vision has developed

rapidly, and deep learning techniques were first used for image classification problems [10–12]. In recent years, the multi-target tracking algorithm based on deep learning has also made some breakthroughs. Multi-target tracking is the very challenging research direction in the field of computer vision and has a wide range of practical applications, for example, intelligent video monitoring control, abnormal behavior analysis, and mobile robot research. Traditional multi-target tracking algorithms tend to have poor tracking performance due to poor target detection. Depth-based learning detectors can achieve better results, which in turn improves tracking accuracy. Therefore, how to achieve effective combination of target tracking and deep learning has become the focus of researchers.

Traditional target tracking algorithms have many problems. For example, the block-based scale-adaptive CSK rigid body target tracking algorithm proposed in Document 4 does not consider the confidence of the candidate box, and the tracking result has low precision. The KCF (kernelized correlation filter)-based tracking algorithm in Document 5 is only applicable to single-target tracking and has limited limitations. The TLD tracking algorithm proposed in the literature 6 causes the target to be lost when the target is severely occluded. The new target tracking algorithm that combines SIFT (scale-invariant features) and compression features is proposed in Document 7. This algorithm has poor effect on feature extraction, which results in lower center position error and coverage, and higher resource occupancy rate of the algorithm. Deep learning is a new research direction in the field of machine learning. It is introduced into machine learning to make it closer to the original goal—artificial intelligence (AI). Deep learning is the internal law and representation level of learning sample data. The information obtained in the learning process is very helpful to the interpretation of data such as text, image, and sound. Its ultimate goal is to make the machine have the ability of analysis and learning like human beings, and can recognize characters, images, sounds, and other data. Deep learning is a complex machine learning algorithm, which has achieved far more results in speech and image recognition than previous related technologies.

In response to the above problems, the target tracking algorithm in the football match video based on deep learning is proposed in this paper. GoogLeNet is used to perform convolution to obtain the feature map array through target detection algorithm-based GoogLeNet+LSTM. After processing, candidate boxes of high confidence used to perform training and matching are obtained to achieve target detection. The feature map of the detection result is pooled to obtain the depth feature required for tracking. According to this feature, the discriminant scale space tracking algorithm and the Markov Monte Carlo algorithm are used to achieve single-target or multi-target tracking.

2. Materials and Methods

2.1. Target Detection-Based GoogLeNet+LSTM. Target detection is the basis of the multi-target tracking algorithm based on data association. The GoogLeNet+LSTM

framework is used for target detection for problems such as small targets and occlusions in football video. First use GoogLeNet for convolution. In the last layer, the $1 \times 1024 \times 15 \times 20$ feature map array is obtained and transposed into 300×1024 feature map array. Each 1024-dimensional vector corresponds to the area of 139×139 in the original picture.

The 300×1024 feature map array is obtained by GoogLeNet convolution, and then, each 1024-dimensional vector is processed in parallel by the LSTM sub-module. The hidden state of each output goes through two different fully connected layers: the position and width of direct output box, and the confidence of this box through the softmax layer. The LSTM sub-module has a total of five such units; that is, each input can predict 5 boxes and confidences. In the training, the frame is concentrated at the 64×64 position in the center of the sensing area, and the confidence is ranked from high to low.

After processing, five detection frames corresponding to 64×64 small blocks in the original image and confidence levels can be obtained. The processing of the sub-module needs to filter all the detection frames of the video frame and then remove the frame with low confidence by the given thresholds. Finally, detection result can be obtained [13–17].

The specific process is as follows: if the candidate box intersects with the determined frame, the candidate box is removed. A determined box removes at most one candidate box. In the above matching, the cost is expressed as (m, d) , m means whether the two intersect, the value is $\{0, 1\}$, and d is the Manhattan distance between the two boxes. The importance of m is greater than that of d ; that is, the result obtained by the two matching schemes is first compared with the size of m . If the conclusion cannot be reached, the size of d is compared. The Hungarian algorithm is used to find the least costly match [18–20]. Assume that the filter's confidence threshold is 0.5, then those boxes with the confidence below 0.5 are removed.

In order to effectively train the target detection model, the following training method is adopted: more candidate boxes are obtained in the LSTM sub-module [21, 22], but there are detection errors.

There are three types of errors [23, 24]:

- (1) Point out the place that is not the tip of the person.
- (2) The difference between the predicted and the true value frame positions.
- (3) Multiple prediction frames are generated for the same target.

The lower confidence level is assigned to the candidate box to prevent the occurrence of Case 1. The error is corrected to avoid the occurrence of Case 2; the lower confidence is given to the prediction boxes generated by the same target to eliminate the problem of Case 3.

The loss function of the model training is

$$L(G, C, f) = \alpha \sum_{i=1}^{|G|} l_{\text{pos}}(b_{\text{pos}}^i, \tilde{b}_{\text{pos}}^{f(i)}) + \sum_{j=1}^C l_c(\tilde{b}_{\text{pos}}^j, \{f^{-1}(j)\}), \quad (1)$$

where G is the true value of the box, C is the candidate box, f is the matching algorithm, b_{pos}^i means the i th box in the truth box, \hat{b}_{pos}^j means the \hat{b}_{pos}^j th box in the to-be-selected box, l_{pos} is the Manhattan distance between the two, and l_c is the cross-entropy loss, which is the softmax loss in the corresponding network [25–27]. The first term of this loss function represents the position error of the candidate box and the matching truth box, the latter represents the confidence of the candidate box, and α adjusts the balance between the two losses.

The matching algorithm is the Hungarian algorithm, and the comparison function used is

$$\Delta(b_i, \hat{b}_j) = (o_{ij}, r_j, d_{ij}), \quad (2)$$

where $o_{ij} \in \{0, 1\}$. If the center of the to-be-selected box falls in the true value box, it is 0; otherwise it is 1. r_j is the sequence number generated for the candidate box. The goal is to have high confidence frame first generated when matching [28–30]. Therefore, when matching the same target, the lower the ranking is, the lower the cost is [31, 32]; d_{ij} is the distance between the two boxes, that is, the distance error.

The target detection results in the football match video are obtained by the detection algorithm-based GoogLeNet + LSTM. Based on this, deep learning is used to extract depth features.

2.2. Extraction of Depth Features. The corresponding box position obtained in the upper section is the position of the human head in the video, which is enlarged by certain scale to cover the whole body [33]. After obtaining the position and size of the target frame, the feature map array obtained by the last layer convolution of GoogleNet is used to extract features. The depth features of each target detected can be obtained by pooling the feature map, because each feature is highly abstract and can well characterize the appearance characteristics of the target in the football match video.

The feature of the proposed algorithm is that the feature map is used for pooling to obtain the depth features required for target tracking without re-training. Therefore, under the premise that the real-time performance of the target tracking algorithm is unchanged, the accuracy of target tracking in the football game video is improved.

2.3. Single-Target Detection-Based DSST. The discriminant scale spatial tracking algorithm is simply referred to as DSST tracking algorithm. After obtaining the depth features, the DSST algorithm is used to track the single target in the video. DSST combines the two-dimensional positional filter with one-dimensional scaled filter. The candidate position is first determined by using the two-dimensional position correlation filter, and this area is used as reference area for the one-dimensional scale filter calculation. In this way, candidate blocks of different scales are obtained, and the scale with high matching degree is searched. The principle of scale selection is as follows:

$$a^n P \times a^n R, n \in \left\{ \left[\frac{S-1}{2}, \dots, \left[\frac{S-1}{2} \right] \right] \right\}, \quad (3)$$

where P and R are the width and height of the target in the previous frame; a is the setting factor, and its value is set as 1.02; and S indicates the number of scales, with the setting of 33. The scale in the formula is not linear relationship, but only the detection process from fine to coarse and from inside to outside.

In extracting image features and generating filters, MOSSE correlation filters are employed [34]. In this way, a series of image blocks are extracted from the target as training samples, which are, respectively, recorded as $y_1, y_2, y_3, \dots, y_n$. The corresponding filter response values are Gaussian functions, which are, respectively, recorded as $g_1, g_2, g_3, \dots, g_n$. The peak is at the center, and the end result is to find a filter that meets the minimum mean square error.

The MOSSE optimal correlation filter formula is

$$\varepsilon = \sum_{j=1}^t \|h_t * y_j - g_j\|^2 = \frac{1}{MN} \|H_t Y_j - G_j\|^2, \quad (4)$$

where G is Gaussian function, $*$ represents complex conjugate, and H_t means the minimum value of the filter. The right equal sign is derived from the Parseval theorem. The right side of the equal sign is the frequency domain equation, and the left side is the airspace equation. This calculation can be used to transform the problem from spatial domain solution to frequency domain solution. In the frequency domain, the minimum value of the filter is as follows:

$$H_t = \frac{\sum_{j=1}^t G_j}{\sum_{j=1}^t Y_j}. \quad (5)$$

After the correlation filter is obtained, the determination of the target position of the next frame is determined by the functional response of the correlation score [35]. The area with high response value is the new target position, and the response formula is

$$x = Y^{-1}(HtZ). \quad (6)$$

In this algorithm, (4) uses the extracted depth feature and Gaussian function G to obtain the correlation filter H . t indicates the response time. When a new frame is input, the feature Z extracted by the image block is used as an input to calculate with the correlation filter H using (5), and the response score x is obtained to get the candidate target [36].

DSST designs the input y of the image into the feature vector of the d dimension. The input signal y represents a certain image block of the input image. The optimal correlation filter H is established by the MOSSE idea. The formula is as follows:

$$\varepsilon = \sum_{t=1}^d \|h^l * y^l - g\|^2 + \lambda \sum_{t=1}^d \|h^l\|^2, \quad (7)$$

where l is one dimension of the feature and λ is regular coefficient, and the solution of the obtained minimum value is as follows:

$$\begin{aligned}
H^l &= \frac{\overline{G}Y^l}{\sum_{k=1}^d \overline{Y}^k Y^k + \lambda} \\
&= \frac{A_t^l}{B_t}.
\end{aligned} \tag{8}$$

Since the pixel points in the image block directly solve the $d \times d$ -dimensional linear equation, the calculation amount is too large and time consuming. Therefore, a robust approximate solution is obtained by updating the numerator and the denominator in the above equation. The formula is rewritten as follows:

$$\begin{aligned}
A_t^l &= (1 - \eta)A_{t-1}^l + \eta \overline{G}_t Y_t^l, \\
B_t &= (1 - \eta)B_{t-1} + \eta \sum_{k=1}^d \overline{Y}_t^k Y_t^l,
\end{aligned} \tag{9}$$

where η is learning rate. The position of the target in the new frame, that is, the maximum response value of the correlation filter, can be obtained by

$$x = Y^{-1} \left\{ \frac{\sum_{l=1}^d \overline{A}^l Z^l}{B + \lambda} \right\}. \tag{10}$$

The DSST algorithm uses the dual correlation filter to track the single target in football match video. The algorithm is more portable and efficient, but the problem still exists. When tracking multiple targets, the occlusion of the target will inevitably reduce the accuracy of the tracking, and the interference discrimination of similar targets is not strong. Therefore, the Markov Monte Carlo (HDDMCMC) algorithm is adopted when tracking multiple targets in the football match video.

2.4. Multi-Target Tracking Based on HDDMCMC Algorithm

2.4.1. MCMC Algorithm in the Segment. In the multi-target tracking algorithm, considering the stability and continuity of motion (the same target in the front and rear frame video data), the appearance characteristics will not change drastically.

In the intra-segment MCMC algorithm, the depth feature in Section 2.2 is used to measure the similarity of the target trajectory.

Each detection target is treated as a node and is described by the intra-segment time $[t, t + T]$. Suppose the set of nodes of the video frame in t is $N_t = \{N^t(1), N^t(2), N^t(3), \dots, N^t(i), \dots, N^t(N^t - 1), N^t(N_t)\}$, and the posterior probability $P(w|D)$ is as

$$\begin{aligned}
P(w|D) &= P(\{\tau_1, \tau_2, \tau_3, \dots, \tau_K\}|D) \\
&= \left(\prod_{k=1}^K \prod_{n=1}^{N_k-1} P(N^{n+1}(k)|N^n(k)) \right) \\
&\quad \times \left(\prod_{k=1}^K e^{-(1/l_k)} \right) \times \left(e^{-|\tau_0|} \right),
\end{aligned} \tag{11}$$

where $N^n(k)$ and $N^{n+1}(k)$ represent the n th and $n+1$ th nodes in the k th pedestrian trajectory, respectively. $P(N^{n+1}(k)|N^n(k))$ is the similarity of two nodes, and it can be calculated by the cosine of the angle of the two-node depth feature. l_k is the length of different tracks, and $|\tau_0|$ represents the number of false alarms to ensure that the false alarm rate is low.

2.4.2. Inter-Segment MCMC Algorithm. The data used are mainly the target trajectory generated by the MCMC algorithm in the segment. The main actions taken by the algorithm include fusion, splitting, and switching operations. After passing the intra-segment MCMC algorithm, many more reliable target trajectories are generated. At this time, if there is a case where the same target trajectory is broken, it is caused by unstable detection data. Therefore, the purpose of the inter-segment MCMC is to further combine the target trajectory data of the two time periods. In the current state, the posterior probability is updated as follows:

$$\begin{aligned}
P(w|D) &= P(\{\tau_1, \tau_2, \tau_3, \dots, \tau_K\}|D) \\
&= \left(\prod_{k=1}^K \prod_{n=1}^{N_k-1} P(N^{n+1}(k)|N^n(k)) \right) \times \left(\prod_{k=1}^K e^{-(1/l_k)} \right),
\end{aligned} \tag{12}$$

where false alarm factors are no longer considered because they are mainly used to divide the target trajectory [37]. For fusion operations, the allowed time interval is set as $t_{\text{gap}} = 10$ and the frame difference at the junction between the track segments of the two targets cannot exceed 6. The standard deviation of the probability is set to $3\sigma = \text{size}(d_i^j)$. In this way, the unit that can be transferred between different states is a relatively complete target trajectory segment that has been generated previously [38–40].

The inter-segment MCMC algorithm moves the target that has gone out of the video scene out of the current data set. The current data set is assumed to be τ' . After the MCMC gets the trajectory in the next segment, it is matched by the inter-segment MCMC algorithm. That is to continue to build on the previous target data, combined with the current target data, to further data integration to optimize. The entire algorithm is continuously performed in such a sliding manner.

3. Results

In order to verify the superiority of each aspect of the proposed algorithm, it is compared with some traditional algorithms such as CSK, KCF, Struck, CT, and TLD. The experimental object is the video of football match. The threshold set by the accuracy is 20 pixels, and the threshold of the success rate is set as $t_0 = 0.5$. The results are shown in Tables 1 and 2.

By comparison, the accuracy of the algorithm is up to 0.88, and the success rate is up to 0.81. Although the effect of this algorithm is not optimal for some video sequences, the algorithm is robust to the overall performance. When the

TABLE 1: Accuracy of different algorithms and comparison results.

Video sequence	Tracking algorithm					
	CSK algorithm	KCF algorithm	Struck algorithm	CT algorithm	TLD algorithm	Algorithm in this paper
1	0.49	1.77	0.46	0.35	0.69	0.78
2	0.48	0.80	0.39	0.37	0.82	0.84
3	0.56	0.76	0.42	0.35	0.43	0.80
4	0.45	0.81	0.48	0.40	0.63	0.82
5	0.56	0.82	0.47	0.42	0.84	0.83
6	0.46	0.17	0.45	0.43	0.32	0.88
7	0.26	0.23	0.41	0.38	0.52	0.77
8	0.35	0.88	0.43	0.39	0.83	0.84

TABLE 2: Comparison of success rates of different algorithms.

Video sequence	Tracking algorithm					
	CSK algorithm	KCF algorithm	Struck algorithm	CT algorithm	TLD algorithm	Algorithm in this paper
1	0.40	1.52	0.52	0.32	0.42	0.71
2	0.45	0.46	0.46	0.28	0.78	0.68
3	0.28	0.45	0.35	0.30	0.37	0.75
4	0.33	0.53	0.53	0.31	0.46	0.67
5	0.29	0.47	0.41	0.28	0.39	0.81
6	0.40	0.48	0.38	0.25	0.44	0.48
7	0.34	0.46	0.46	0.18	0.36	0.52
8	0.45	0.53	0.32	0.33	0.41	0.63

TABLE 3: Average center position deviation of different algorithms (pixel).

Video sequence	CSK algorithm	KCF algorithm	Struck algorithm	CT algorithm	TLD algorithm	Algorithm in this paper
1	121.6	18	—	103.8	62.1	12.0
2	9.9	276.7	—	7.0	40.3	10.9
3	86.7	165.4	—	82.4	3.0	3.5
4	27.8	75.6	—	30.5	5.5	3.6
5	18.9	18.8	10.4	19.4	14.2	13.0
6	134.6	104.3	14.3	125.3	20.4	11.6
7	75.1	181.6	—	90.3	12.8	16.0
8	212.3	154.8	—	212.7	7.0	9.7

TABLE 4: Average coverage of different algorithms (%).

Video sequence	CSK algorithm/ %	KCF algorithm/ %	Struck algorithm/ %	CT algorithm/ %	TLD algorithm/ %	Algorithm in this paper/ %
1	20.8	60.3	—	20.8	35.4	50.9
2	62.2	1.0	—	69.1	37.2	49.6
3	22.5	7.5	—	20.5	76.3	77.3
4	41.2	41.2	—	34.2	72.3	85.3
5	32.4	35.5	72.0	32.1	52.7	60.8
6	4.1	10.7	41.6	2.5	48.0	39.6
7	24.1	10.2	—	30.5	74.6	71.5
8	13.5	29.4	—	12.1	73.7	60.4

target is partially occluded, the algorithm can still accurately track the target.

The center position error refers to the center deviation of the tracking frame from the real target frame, and the coverage ratio is the proportion of the intersection of the tracking frame and the real target frame in the merged portion. In order to evaluate the tracking performance of different algorithms on the entire video series, the experiment will use the average center position deviation and

average coverage as indicators to test, and the results are shown in Tables 3 and 4.

It can be seen from Tables 3 and 4 that among the eight tracking video sequences, the average center position deviation index of the algorithm has three groups of best and two groups of two; the average coverage has two groups of best and four groups of second. The experiment used the scoring method to evaluate the two indicators separately. The rule is to sort the two indicators according to their

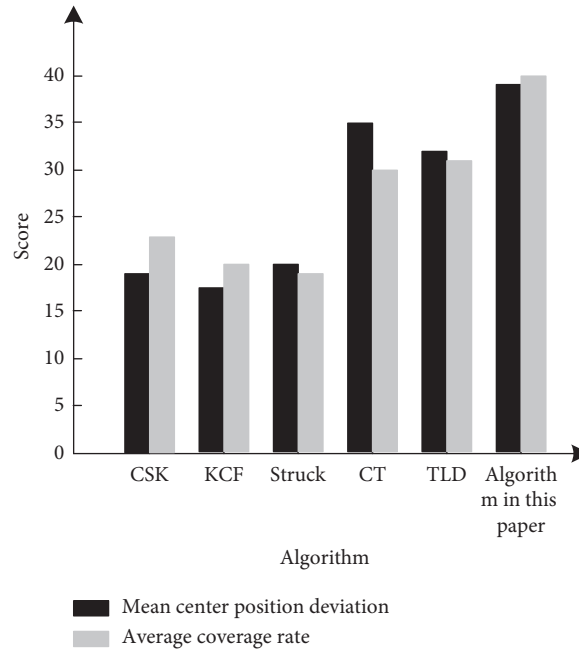


FIGURE 1: Performance comparison between average center position deviation and average coverage rate of different algorithms.

TABLE 5: Average frame rate comparison results of different algorithms (Hz).

Video sequence	CSK algorithm/ Hz	KCF algorithm/ Hz	Struck algorithm/ Hz	CT algorithm/ Hz	TLD algorithm/ Hz	Algorithm in this paper/ Hz
1	25.3	27.6	—	27.3	24.6	38.2
2	20.8	25.3	—	26.9	28.4	35.9
3	24.6	21.7	—	29.2	24.8	37.6
4	22.7	22.4	—	30.6	29.3	40.4
5	26.8	20.8	19.9	31.4	27.2	39.3
6	24.5	23.2	22.7	28.4	30.6	38.6
7	23.9	24.5	—	29	33.7	40.2
8	21.2	23.6	—	30.1	25.5	37.5

performance from high to low, and then score them in 6, 5, 4, 3, 2, and 1. Each video sequence is scored in turn, and finally, they are summed and used as their final result, as shown in Figure 1.

Analysis of Figure 1 shows that the algorithm scores 39 points on the average center position deviation and 40 points on the average coverage rate, which are better than other algorithms. This shows that the algorithm is better in the listed tracking algorithms.

The video frame rate is a measure used to measure the number of displayed frames and reflects the smoothness of the tracking results. The average frame rate of the eight video sequences is compared using different algorithms, and the results are shown in Table 5.

Analysis Table 5 can be obtained that the average frame rate of the algorithm in this paper is higher than other algorithms, both above 35 Hz. This shows that the tracking results using the algorithm of this paper are more fluent.

In order to verify the efficiency of the algorithm, the iterations and time consumption of different algorithms are compared, respectively, and the results are described in Figure 2.

It can be seen from Figure 2 that the iteration number of the algorithm is similar to the number of CT algorithms, with an average of 1–2 times, which is significantly lower than the CSK algorithm. In the comparison of time consumption, the time consumed by the algorithm is similar to the CSK algorithm, with an average of 12.5 s, which is significantly lower than the CT algorithm. The comparison results show that the tracking efficiency of the algorithm is higher.

To verify the stability of the proposed algorithm, three algorithms are used to track the target in the same experimental environment. The average outage probability of different algorithms is compared, and the results are shown in Table 6.

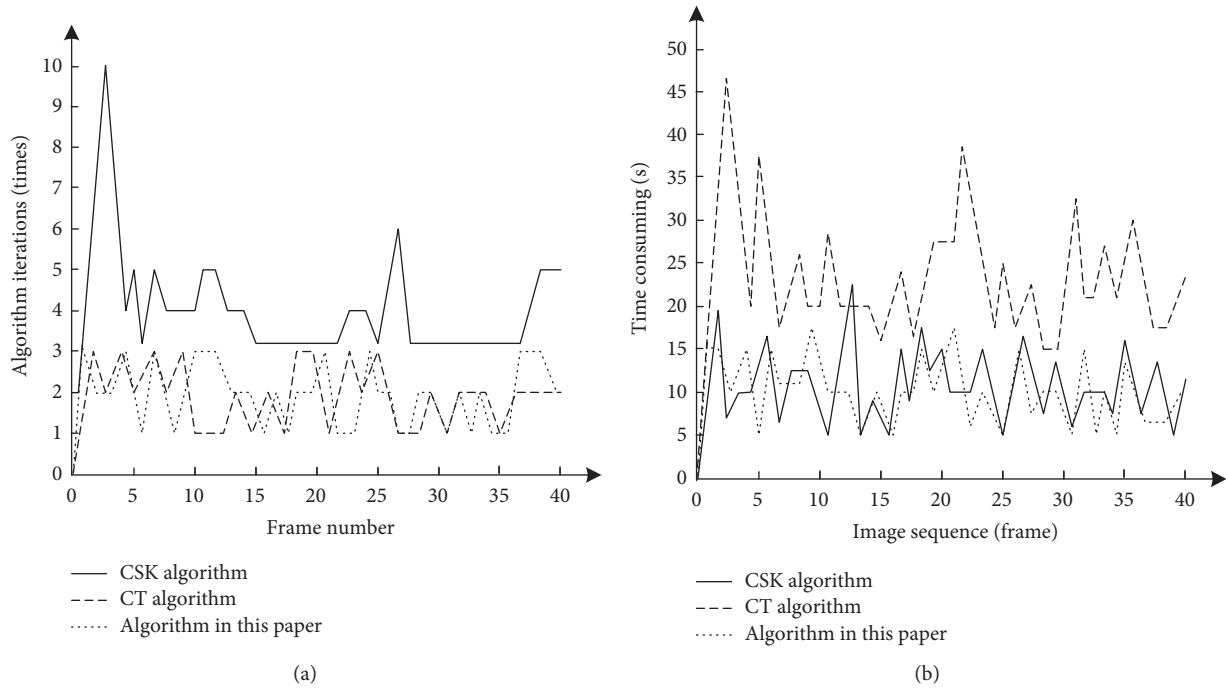


FIGURE 2: Comparison of efficiency of different tracking algorithms. (a) Comparison of iterations. (b) Time consumption contrast.

TABLE 6: Comparison of average outage probability of different location systems.

Average signal-to-noise ratio/dB	Algorithm in this paper	CSK algorithm	CT algorithm
0–5	0.012	0.014	0.011
5–10	0.014	0.014	0.015
10–15	0.017	0.016	0.018
15–20	0.264	0.383	0.461
20–25	0.297	0.455	0.518
25–30	0.306	0.498	0.624
30–35	0.325	0.534	0.659
35–40	0.330	0.607	0.768
40–45	0.384	0.711	0.839
45–50	0.422	0.793	0.972
Average value	0.2371	0.4025	0.4885

In order to more clearly show the stability of the algorithms, the data in Table 6 are described by the implementation of the line graph, as shown in Figure 3.

Analysis of Table 6 and Figure 3 shows that the average outage probability of the algorithm is 0.2371, which is lower than the other two algorithms. The experimental results show that the proposed algorithm has better stability when tracking the target in the football match video.

In order to test the resource occupancy rate of the proposed algorithm, it compares and analyzes the target detection, feature extraction, single-target tracking, and multi-target tracking. The results are shown in Table 7.

It can be seen from Table 7 that the CPU and memory usage of the proposed algorithm are 26%–34% and 7%–17%, respectively. The usages of the CSK algorithm are 64%–70%

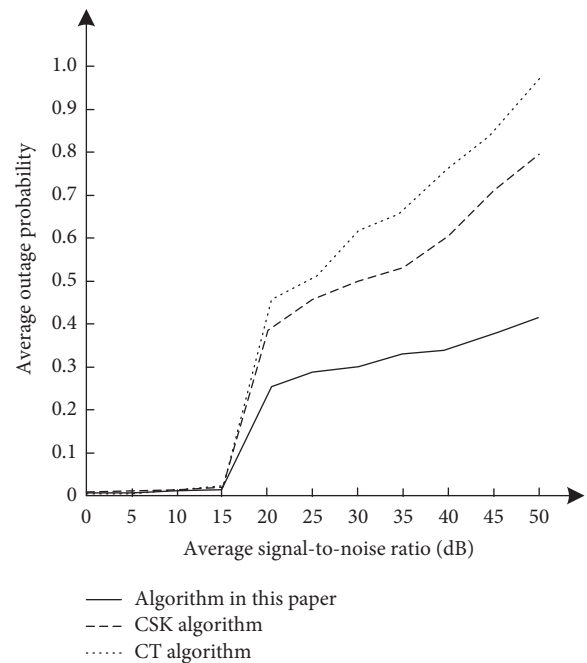


FIGURE 3: Comparison of average outage probability of different location systems.

and 33%–41%, respectively. Those of the CT algorithm are 64%–72% and 29%–39%, respectively. The experimental results show that compared with the other two algorithms, the algorithm of this paper tracks the resource occupancy rate of the target in the football match video which is low.

TABLE 7: Resource occupancy ratio of different platforms/%.

		Object detection	Feature extraction	Single-target tracking	Multi-target tracking
Algorithm in this paper	CPU occupancy rate/%	29	26	31	34
	Memory usage%	9	7	13	17
CSK algorithm	CPU occupancy rate/%	69	66	64	70
	Memory usage%	39	33	34	41
CT algorithm	CPU occupancy rate/%	65	64	68	72
	Memory usage%	37	29	33	39

4. Discussion

The accuracy and success rate of the algorithm are as high as 0.88 and 0.81, respectively; the average center position deviation index and the average coverage index are 39 and 40 points, respectively; the average frame rate is maintained above 35 Hz, and the tracking time is about 12.5 s. These data show that the proposed algorithm outperforms other algorithms in performing target tracking. The reason is the use of deep learning techniques in this paper. Deep learning is artificial neural network that simulates the human brain's analysis of things. By simulating the human brain to acquire data and parse it, this structure can better learn the essential characteristics of objects. The main ideas of deep learning target tracking are as follows: first, construct a deep learning model to train standard data sets and obtain more accurate target feature information. Then, use this model for target matching and positioning to achieve efficient target tracking. Depth features can more accurately reflect the appearance characteristics of moving objects than traditional features such as scale-invariant features (SIFT). Therefore, the algorithm of this paper greatly improves the accuracy of target tracking. At the same time, it combines the discriminant scale space algorithm and the Markov Monte Carlo algorithm to track the targets in the football video to ensure efficient tracking of single targets while achieving accurate tracking of multiple targets.

5. Conclusions

With the rapid development of computer technology, deep learning has become a big weapon for video target tracking. Deep learning technology has the advantages of high precision, wide application range, and strong stability. Aiming at the traditional target tracking algorithm in the football match video, there is the defect that the target will be lost when the target is severely occluded. A target tracking algorithm based on deep learning football game video is proposed. Combined with deep learning and target tracking technology to track the target, the experimental results show that the tracking success rate, required time, and average frame rate of the proposed algorithm are 0.81, 12.5 s, and 35 Hz, respectively. The average center position deviation index and the average coverage index are 39 and 40 points, respectively, and the resource occupancy rate is low. This shows that the algorithm can well track the targets in the football match video. In the future research, the information technology model is used to further improve the accuracy of

football game video target tracking and reduce the defect of losing the target when the target is seriously blocked [41–45].

Data Availability

The data sets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that this article is free of conflicts of interest.

References

- [1] X. Fu, P. Pace, G. Aloï, L. Yang, and G. Fortino, "Topology optimization against cascading failures on wireless sensor networks using a memetic algorithm," *Computer Networks*, vol. 177, Article ID 107327, 2020.
- [2] Z. Lv and N. Kumar, "Software defined solutions for sensors in 6G/LoE," *Computer Communications*, vol. 153, pp. 42–47, 2020.
- [3] Z. Lv and L. Qiao, "Deep belief network and linear perceptron based cognitive computing for collaborative robots," *Applied Soft Computing*, vol. 92, Article ID 106300, 2020.
- [4] Z. Lv and W. Xiu, "Interaction of edge-cloud computing based on SDN and NFV for next generation IoT," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5706–5712, 2020.
- [5] C. Mi, L. Cao, Z. Zhang, Y. Feng, L. Yao, and Y. Wu, "A port container code recognition algorithm under natural conditions," *Journal of Coastal Research*, vol. 103, no. 1, p. 822, 2020.
- [6] Q. Wu, W. Diao, F. Dou et al., "Shape-based object extraction in high-resolution remote-sensing images using deep Boltzmann machine," *International Journal of Remote Sensing*, vol. 37, no. 24, pp. 6012–6022, 2016.
- [7] X. Zenggang, T. Zhiwen, C. Xiaowen, Z. Xue-Min, Z. Kaibin, and Y. Conghuan, "Research on image retrieval algorithm based on combination of color and shape features," *Journal of Signal Processing Systems*, vol. 93, 2019.
- [8] C. Zuo, Q. Chen, L. Tian, L. Waller, and A. Asundi, "Transport of intensity phase retrieval and computational imaging for partially coherent fields: the phase space perspective," *Optics and Lasers in Engineering*, vol. 71, pp. 20–32, 2015.
- [9] S. I. Jang, K. Choi, K. A. Toh, B. J. T. Andrew, and K. Jaihie, "Object tracking based on an online learning network with total error rate minimization," *Pattern Recognition*, vol. 48, no. 1, pp. 126–139, 2015.
- [10] Z. He, Y. Cui, H. Wang, Y. Xinge, and C. C. L. Philip, "One global optimization method in network flow model for multiple object tracking," *Knowledge-Based Systems*, vol. 86, no. C, pp. 21–32, 2015.

- [11] J. Li, K. Xu, S. Chaudhuri, E. Yumer, H. Zhang, and L. Guibas, "GRASS: generative recursive autoencoders for shape structures," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–14, 2017.
- [12] J. Zhang, C. Zhu, L. Zheng, and K. Xu, "ROSEFusion: random optimization for online dense reconstruction under fast camera motion," *ACM Transactions on Graphics*, vol. 40, no. 4, pp. 1–17, 2021.
- [13] L. Cai and Y. H. Liu, "Application of Lossless Kalman Filtering Algorithm in Target Tracking," *Automation & Instrumentation*, vol. 7, pp. 112–115, 2015.
- [14] T. Sui, D. Marelli, X. Sun, and M. Fu, "Multi-sensor state estimation over lossy channels using coded measurements," *Automatica*, vol. 111, Article ID 108561, 2020.
- [15] W. Zheng, X. Liu, and L. Yin, "Research on image classification method based on improved multi-scale relational network," *PeerJ Computer Science*, vol. 7, p. e613, 2021.
- [16] W. Zheng, X. Tian, B. Yang et al., "A few shot classification methods based on multiscale relational networks," *Applied Sciences*, vol. 12, no. 8, 2022.
- [17] W. Zheng and L. Yin, "Characterization inference based on joint-optimization of multi-layer semantics and deep fusion matching network," *PeerJ Computer Science*, 2022.
- [18] D. Li, S. S. Ge, and T. H. Lee, "Fixed-time-synchronized consensus control of multiagent systems," *IEEE transactions on control of network systems*, vol. 8, no. 1, pp. 89–98, 2021.
- [19] F. Meng, W. Cheng, and J. Wang, "Semi-supervised software defect prediction model based on tri-training," *Ksii Transactions on Internet and Information Systems*, vol. 15, no. 11, pp. 4028–4042, 2021.
- [20] C. Zuo, J. Qian, S. Feng et al., "Deep learning in optical metrology: a review," *Light: Science & Applications*, vol. 11, no. 1, p. 39, 2022.
- [21] L. Ying, T. Zhang, and C. Xu, "Multi-object tracking via MHT with multiple information fusion in surveillance video," *Multimedia Systems*, vol. 21, no. 3, pp. 313–326, 2015.
- [22] Y. Zhou, G. Xu, K. Tang, L. Tian, and Y. Sun, "Video coding optimization in AVS2," *Information Processing & Management*, vol. 59, no. 2, Article ID 102808, 2022.
- [23] J. Choi and M. Maurer, "Local volumetric hybrid-map-based simultaneous localization and mapping with moving object tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2440–2455, 2016.
- [24] B. Li, Y. Feng, Z. Xiong, W. Yang, and G. Liu, "Research on AI security enhanced encryption algorithm of autonomous IoT systems," *Information Sciences*, vol. 575, pp. 379–398, 2021.
- [25] B. Luo, C. Liang, W. Ruan, and H. Ruimin, "Stable and salient patch-based appearance model for robust object tracking," *Electronics Letters*, vol. 52, no. 18, pp. 1522–1524, 2016.
- [26] G. Sun, Y. Cong, J. Dong, Y. Liu, Z. Ding, and H. Yu, "What and how: generalized lifelong spectral clustering via dual memory," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, 2021.
- [27] W. Zhou, Q. Guo, J. Lei, L. Yu, and J. Hwang, "IRFR-net: interactive recursive feature-reshaping network for detecting salient objects in RGB-D images," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2021.
- [28] B. Cao, W. Zhang, X. Wang, J. Zhao, Y. Gu, and Y. Zhang, "A memetic algorithm based on two_Arch2 for multi-depot heterogeneous-vehicle capacitated arc routing problem," *Swarm and Evolutionary Computation*, vol. 63, Article ID 100864, 2021.
- [29] B. Cao, M. Li, X. Liu, J. Zhao, W. Cao, and Z. Lv, "Many-objective deployment optimization for a drone-assisted camera network," *IEEE transactions on network science and engineering*, vol. 8, no. 4, pp. 2756–2764, 2021.
- [30] B. Wu and X. Yi, "A wireless sensor network-weighted-centroid tracking algorithm based on the dynamic cluster," *Journal of China Academy of Electronics and Information Technology*, vol. 10, no. 4, pp. 355–360, 2015.
- [31] B. Li, J. Yang, Y. Yang, C. Li, and Y. Zhang, "Sign language/gesture recognition based on cumulative distribution density features using UWB radar," *IEEE Transactions on Instrumentation and Measurement*, vol. 1, 2021.
- [32] A. Vatavu, R. Danescu, and S. Nedeveschi, "Stereovision-based multiple object tracking in traffic scenarios using free-form obstacle delimiters and particle filters," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 498–511, 2015.
- [33] L. Yang, J. Cao, W. Zhu, and S. Tang, "Accurate and efficient object tracking based on passive RFID," *IEEE Transactions on Mobile Computing*, vol. 14, no. 11, pp. 2188–2200, 2015.
- [34] C. Chen, S. Li, H. Qin, and H. Aimin, "Real-time and robust object tracking in video via low-rank coherency analysis in feature space," *Pattern Recognition*, vol. 48, no. 9, pp. 2885–2905, 2015.
- [35] B. Zhao, J. Li, B. W. Wu, and Y. Xu, "Research on adaptive Markov transition probability matrix IMM-CKF target tracking algorithm," *Computer Simulation*, vol. 32, no. 11, pp. 106–109, 2015.
- [36] D. Wang, H. Lu, and C. Bo, "Fast and robust object tracking via probability continuous outlier model," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5166–5176, 2015.
- [37] P. Chen, Y. Dang, R. Liang, W. Zhu, and X. He, "Real-time object tracking on a drone with multi-inertial sensing data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 131–139, 2017.
- [38] P. Chiranjeevi and S. Sengupta, "Rough-set-theoretic fuzzy cues-based object tracking under improved particle filter framework," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 3, pp. 695–707, 2016.
- [39] W. Gao and W. F. Wang, "The fifth geometric-arithmetic index of bridge graph and carbon nanocones," *Journal of Difference Equations and Applications*, vol. 23, no. 1-2SI, pp. 100–109, 2017.
- [40] A. Yang, Y. Han, Y. Pan, H. Xing, and J. Li, "Optimum surface roughness prediction for titanium alloy by adopting response surface methodology," *Results in Physics*, vol. 7, pp. 1046–1050, 2017.
- [41] M. Buchheit, C. Manouvrier, J. Cassirame, and J. B. Morin, "Monitoring locomotor load in soccer: is metabolic power, powerful?" *International Journal of Sports Medicine*, vol. 36, no. 14, pp. 1149–1155, 2015.
- [42] J. J. Cheng, D. R. Feng, and X. C. Pan, "Study on frequency tracking series resonant converters based on DSP," *Journal of Power Supply*, vol. 13, no. 3, pp. 50–54, 2015.
- [43] D. Du, H. Qi, W. Li, Q. Huang, and S. Lyu, "Online deformable object tracking based on structure-aware hypergraph," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3572–3584, 2016.
- [44] C. Shi, L. Zhao, P. Ma, X. Su, J. Wang, and C. Zhang, "Multi-object tracking based on particle probability hypothesis density tracker in microscopic video," *Journal of Mathematical Biology*, vol. 72, no. 5, pp. 1–30, 2015.
- [45] K. Zhang, Q. Liu, Y. Wu, and M. Yang, "Robust visual tracking via convolutional networks without training," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1779–1792, 2016.