Hindawi
Mobile Information Systems
Volume 2021, Article ID 3577541, 7 pages
https://doi.org/10.1155/2021/3577541

Research Article

Neural Network for Intelligent and Efficient Volleyball Passing Training

Bo Liu, Ning Yang, Xiangwei Han, and Chen Liu

Shandong Youth University of Political Science, Jinan 250103, China

Correspondence should be addressed to Ning Yang; 190040@sdyu.edu.cn

Received 17 October 2021; Accepted 6 November 2021; Published 22 November 2021

Academic Editor: Jianhui Lv

Passing is a relatively basic technique in volleyball. In volleyball passing teaching, training the correct passing technique plays a very important role. The correct pass can not only accurately grasp the direction of the ball point and the drop point but also effectively connect the defense and the offense. In order to improve the efficiency and quality of volleyball passing training, improve the precise extraction of sport targets, reduce redundant feature information, and improve the generalization performance and nonlinear fitting capabilities of the algorithm, this paper studies volleyball based on the nested convolutional neural network model and passing training wrong movement detection method. The structure of the convolutional neural network is improved by nesting mlpconv layers, and the Gaussian mixture model is used to effectively and accurately extract the foreground objects in the video. The nested multilayer mlpconv layer automatically learns the deep-level features of the foreground target, and the generated feature map is vectorized and input to the Softmax classifier connected to the fully connected layer for passing wrong behavior detection in volleyball training. Based on the detection of nearly 1,000 athletes' action datasets, the simulation experiment results show that the algorithm reduces the acquisition of redundant information and shortens the calculation time and learning time of the algorithm, and the improved convolutional neural network has generalization performance and nonlinearity. The fitting ability has been improved, and the detection of abnormal volleyball passing behaviors has achieved a higher accuracy rate.

1. Introduction

1.1. Volleyball Passing Technique. For volleyball, passing technique is a very basic technique. To be able to pass the ball accurately, it is necessary to have the corresponding passing technique. The volleyball passing technique seems simple, but it is actually a very delicate and complex technique that requires high wrist strength. In the process of volleyball passing training, after the students learn the movements for the first time, the connection between the nerves and muscles of the main body of the movement is not precise, so wrong movements are often accompanied during the movement, so how to correct them in the process of volleyball passing teaching and preventing wrong moves are the basic requirements for physical education teachers to improve the quality of volleyball teaching. According to relevant investigations and studies, in the past correction of volleyball passing training actions, wrong actions could not be accurately detected, so wrong actions could not be corrected in time [1–3]. Driven by the general trend of international sports, volleyball sports activities continue to enter people's field of vision, and people begin to pay attention to volleyball training [4, 5]. In the actual training process, the volleyball passing technique seems relatively easy, but it is relatively difficult to learn. In the process of passing technique learning, many students often have uncoordinated body movements, and their fingers and wrists are obviously insufficient. It is easier to rub hands when passing volleyball. Many students have different situations such as being afraid of passing. This leads to more difficulties in teaching volleyball passing techniques. In this case, it is required to pass training in volleyball. Instructors in the middle school can effectively point out and correct the wrong actions of the athletes [6].

1.2. The Role of Neural Networks. In recent years, pattern recognition using machine learning to build detectors has been successfully applied in the field of visual detection, such as face recognition and car recognition [7]. For the detection of volleyball players, there are two main challenges: first, the target may be distorted due to the change of posture and scale [8]. Second, the resolution of the image is low, and the target object may be represented less than 200 pixels. At present, a new method for athlete detection in sports videos is the AdaBoost algorithm [9], which firstly extracts Har features from rectangular images of athletes and then uses AdaBoost algorithm to cascade weak classifiers to build strong classifiers [10], achieving good detection performance. However, it needs too many features, the detection speed is slow, and it cannot meet the real-time requirements well. With the further development of research in related fields, convolutional neural networks have been widely used in the process of human action recognition [11] and target detection [12]. It can realize the further processing of the action samples, which leads to the poor applicability of applying this method to the detection of wrong actions in physical education training.

Therefore, in order to make up for the abovementioned deficiencies in the process of volleyball passing training wrong action detection, this paper proposes an improved convolutional neural network crowd abnormal behavior recognition method, which improves the convolutional neural network structure by nesting mlpconv layers and uses a mixture of Gaussian models to effectively and accurately extract foreground targets in volleyball passing videos. The nested multilayer mlpconv layer automatically learns the depth-level features of foreground targets, and the generated feature maps are vectorized and input to the Softmax classification connected to the fully connected layer detectors for abnormal behaviors in volleyball passing training. The simulation experiment results show that the algorithm reduces the acquisition of redundant information and shortens the calculation time and learning time of the algorithm, and the improved convolutional neural network has improved generalization performance and nonlinear fitting ability, and it is abnormal for volleyball passing. Behavior detection achieves a high accuracy rate.

The remaining structure of this paper is organized as follows: Section 2 introduces convolutional neural network architecture. Section 3 gives the detailed design on volleyball passing training scheme. Section 4 reports the rich results. Finally, Section 5 concludes this paper.

2. Convolutional Neural Network Architecture

A convolutional neural network [13] is a hierarchical neural network based on local connections between neurons. Compared with traditional machine learning methods, it has a more complex network structure and more powerful feature learning and expression ability. The visual mode is decomposed into multiple submodes, and the submodes are processed by the feature planes connected step by step, so that the target can be well recognized even with small distortion [14]. The convolutional neural network consists of six different types of convolutional layers. The input layer receives 21×43 grayscale images, and C1 input images are convolved. The layer consists of four feature maps, each sharing an acceptance field and a bias. The S1 layer performs a secondary sampling and local average operation on the image to form four feature maps. The secondary sampling operation reduces the two dimensions of the input and enhances the invariance of image translation, scaling, and deformation. In addition, the output of the mixed feature map combines different features, which helps to extract more complex information. The C2 layer is not completely connected with the S1 layer, and the output image of the S1 layer is convolved in the 3×3 acceptance domain to generate 14 feature maps. The S2 layer has the same function as the S1 layer and consists of 14 feature maps. Each neuron in the N1 layer is connected to the feature map of the S2 layer. The N2 layer is the output layer and is fully connected to the N1 layer. The N2 layer uses a typical sigmoid neuron. After completing feature extraction and input dimensionality reduction, the role of the N1 and N2 layers is to perform output classification. The output of the neurons in the N2 layer is the identifier of the input image for the athlete or nonathlete, -1 for nonathletes and $+1$ for athletes.

3. Proposed Volleyball Passing Training Detection Scheme

3.1. Detection and Extraction of Volleyball Passing Movement Targets. In actual volleyball passing training, especially in outdoor sports, the environment and background of the sports target are constantly changing, and there are often some nontarget small sports in the background of the video image, such as throwing volleyball and shaking. In order to avoid the influence of background transformation and interfering targets on the detection effect of volleyball passes, the researchers proposed a mixture of Gaussian background modeling. If the difference between the gray value of the target area and the background information in the video image is large, the gray histogram of the video image is a double-peak-valley type, where one peak represents the moving target and the other peak represents the background of the image. For more complex images, the resulting grayscale histogram is multimodal, which can be seen as using multiple single Gaussian models to describe the change of a certain pixel over a period of time, which is a mixed Gaussian background modeling. In short, the mixed Gaussian background modeling is to accurately describe the characteristics of the pixels. By setting multiple Gaussian models for each pixel to improve its ability to describe the background, the purpose of accurately describing the image background is achieved, thereby obtaining a complete moving target. The number of Gaussian models usually selected in different documents is between 3 and 5 [15]. For more complex scenes, a larger number of models are chosen to improve the model's ability to describe the background. For simpler scenes, choosing a small number of models can avoid overdepicting the background by the model and causing the loss of moving targets [16].

At present, there are three main methods of passing target detection in volleyball training videos: optical flow method, interframe difference method, and background difference method [17]. In this paper, the Gaussian Mixture Model (GMM) [18] in the background difference method is selected. Compared with other methods of foreground target extraction, this model can not only successfully detect volleyball passing targets but also reduce the influence of small repetitive objects in the background scene on foreground target detection. For the detection of passing targets in volleyball training videos, firstly, Gaussian distribution is used to establish a background model for each pixel, and then background model parameters are automatically updated. Finally, the successful detection and extraction of passing targets in volleyball training videos are realized.

3.1.1. Mixed Gaussian Background Modeling. For any pixel, its historical pixel sequence can be traced as follows:

$$\{x_1, x_2, \ldots, x_t\} = \{F_i(x, y), \quad 1 \le i \le t\}, \tag{1}$$

where $F_i(x, y)$ is the gray value at the i moment.

At time t, the calculation formula of the probability function of the pixel (x, y) is as follows [19]:

$$p(F_t(x, y)) = \sum_{i=1}^{k} H_{i,t} \times \eta\left(F_t(x, y), \mu_{i,t}, \sum_{i,t}\right), \tag{2}$$

where $H_{i,t}$ is the i model weight value at time t; $\mu_{i,t}$ is the mean value of the i Gaussian distribution at time t; $\sum_{i,t}$ is the covariance at time t; and $\eta(F_t(x, y), \mu_{i,t}, \sum_{i,t})$ is the probability density function at time t. The calculation formula is as follows:

$$\eta\left(F_t(x, y), \mu_{i,t}, \sum_{i,t}\right) = \frac{1}{(2\pi)^{\pi/2}\left|\sum_{i,t}\right|^{1/2}} \times \exp\left(-\frac{1}{2}\left(F_t(x, y) - \mu_{i,t}\right)^T \sum_{i,t}^{-1}\left(F_t(x, y) - \mu_{i,t}\right)\right). \tag{3}$$

3.1.2. Update of Gaussian Mixture Model Parameters. The Gaussian distribution of the frame pixel value $F_i(x, y)$ is sorted according to the priority, and formula (4) is satisfied, indicating that the frame pixel value $F_i(x, y)$ matches the Gaussian distribution i successfully, and the parameters of the Gaussian distribution of the frame pixel value $F_i(x, y)$ follow formula (4) and equation (8) is updated; equation (4) is not satisfied, the frame pixel value $F_i(x, y)$ matches the Gaussian distribution unsuccessfully, the parameters of the Gaussian distribution remain unchanged, and the weight value is updated according to equation (9).

$$\left|F_t(x, y) - \mu_{i,t-1}\right| < D \times \sigma_{i,t-1}, \tag{4}$$

$$H_{i,t} = (1 - \alpha)H_{i,t-1} + \alpha, \tag{5}$$

$$\mu = (1 - \beta)\mu_{i,t-1} + \beta F_i(x, y), \tag{6}$$

$$\delta_{i,t}^2 = (1 - \beta)\delta_{i,t-1}^2 + \beta\left(F_t(x, y) - \mu_{i,t-1}\right)^T\left(F_t(x, y) - \mu_{i,t-1}\right), \tag{7}$$

$$\beta = \alpha\eta\left(F_t(x, y) \mid \mu, \delta_{i,t}\right), \tag{8}$$

$$F_{i,t} = (1 - \alpha)F_{i,t-1} + \alpha, \tag{9}$$

where α and β, respectively, represent the learning rate and update rate of the Gaussian mixture model.

3.1.3. Extraction of Volleyball Pass Target by the Gaussian Mixture Model. After the Gaussian mixture model of each pixel is generated, the Gaussian distribution is arranged in descending order according to the value of ψ/μ, and the first B Gaussian distribution is obtained as the background model [20]. The formula is as follows:

$$B = \arg\min \sum_{i-1}^{b} \psi_{i,t} > T, \tag{10}$$

where T is the set threshold.

The first B Gaussian distribution is obtained as the background model, and the current pixel value $F_i(x, y)$ is matched with the generated background. If the current pixel value $F_i(x, y)$ is not successfully matched with the generated background, then the current pixel $F_i(x, y)$ is a good target for volleyball passing. Otherwise, the current pixel $F_i(x, y)$ point is the background point. After the above process, the Gaussian mixture model realizes the detection and extraction of the passing target in the volleyball training video.

3.2. Obtaining Volleyball Pass Characteristic Information

3.2.1. Mlpconv Layer. The mlpconv layer consists of a linear convolutional layer and a multilayer perceptron (MLP), and the input mapping in the local perception field of view corresponds to the feature vector. The mlpconv layer uses

multiple fully connected layers with nonlinear activation functions to extract the feature information of the volleyball pass target, converts the extracted feature information into a feature map, and then uses the feature map as the input of the next layer [21].

The calculation process of the mlpconv layer is as follows:

$$\begin{aligned} g^1_{i,j,k_1} &= \max\left(H^{1T}_{k_1} x_{i,j} + b_{k_1}, 0\right), \\ g^2_{i,j,k_2} &= \max\left(H^{2T}_{k_2} g^1_{i,j} + b_{k_2}, 0\right), \end{aligned} \tag{11}$$

$$g^n_{i,j,k_n} = \max\left(H^{nT}_{k_n} g^{n-1}_{i,j} + b_{k_n}, 0\right), \tag{12}$$

where (i, j) is the position of the pixel in the feature map, $x_{i,j}$ is the input block centered at the pixel point (F, j), k_1, k_2, and k_n are the channel numbers in the feature map, and n is the number of MLP layers.

3.2.2. Batch Normalization Technology. In the neural network learning process, with the changes of the parameters of each layer, especially the algorithm's learning rate and weight initialization, it will take a long time to find a suitable value, which reduces the training speed of the neural network. When using a saturated nonlinear activation function to train a neural network model, the input data will mistakenly enter the saturation region of the activation function, which reduces the convergence speed of the neural network.

Ioffe et al. [22] used BN (Batch Normalization) technology to standardize the input of each layer to solve the above problems. BN technology makes the input data have zero mean and unit variance:

$$\widehat{x}_{i,j,n} = \frac{\widehat{x}_{i,j,n} - E[x_n]}{\sqrt{\mathrm{Var}[x_n]}}. \tag{13}$$

After normalization, the parameters need to be scaled and translated accordingly:

$$g_{i,j,n} = \gamma_n \widehat{x}_{i,j,n} + \beta_n, \tag{14}$$

where $\widehat{x}_{i,j,n}$ is the value of the input data at position (i, j), n is the channel number in the feature map, and γ_n and β_n are the newly introduced zoom and translation parameters in network training.

3.3. Building a Nested Model of the Convolutional Neural Network. The core idea of the convolutional neural network nested model is as follows: the nested network model can automatically learn deep-level features excellently. The deep-level features acquired by this model are mainly local features. When the nested network model obtains the feature information of the moving target, especially in the separation of the target in the background, the local features will play an important role. In addition, the nested network model also has a certain degree of robustness when dealing with drastic changes in the background target.

When the network nested model is trained, first, the weights of the convolutional neural network model is initialized containing a single mlpconv layer, and then the convolutional neural network is trained. The entire training process is over, and the weights of the single mlpconv layer are updated; then, it is connected to the second mlpconv layer. The input of the second mlpconv layer is the output of the first mlpconv layer. The weights of the second mlpconv layer are initialized, and then the convolutional neural network is trained. The whole training process is over, and an update of the weight of the second mlpconv layer is received. When a new mlpconv layer is added, weight initialization, convolutional neural network training, and weight update are performed according to the above process.

In addition, the use of BN technology after the convolution calculation also enables the nonlinear unit to produce a relatively stable distribution and achieve the effect of desaturation. The BN operation is added to the nested mlpconv layer, and the calculation method of the feature map in the model is as follows:

$$\begin{aligned} g^1_{i,j,k_1} &= \max\left(\mathrm{BN}\left(H^{1T}_{k_1} x_{i,j} + b_{k_1}\right), 0\right), \\ g^2_{i,j,k_1} &= \max\left(\mathrm{BN}\left(H^{2T}_{k_2} g^1_{i,j} + b_{k_2}\right), 0\right), \\ g^n_{i,j,k_n} &= \max\left(\mathrm{BN}\left(H^{n^T}_{k_n} g^{n-1}_{i,j} + b_{k_n}\right), 0\right), \end{aligned} \tag{15}$$

where $\mathrm{BN}(g)$ represents the BN layer, (i, j) is the position of the pixel in the feature map, $x_{i,j}$ is the input block centered on the pixel point (i, j), k_1, k_2, and k_n are the channel numbers in the feature map, and n is the MLP layer number.

4. Experiment and Result Analysis

On 3.0 GHz CPU, 64 bit Windows 7 operating system, MATLAB 2016a, and Open CV are used as development tools for simulation experiments. In order to prove the effectiveness of the intelligent and efficient volleyball pass training detection modeling method based on convolutional neural network, an experiment is needed. The experimental objects were collected from the data collection of volleyball passing training of 1,000 students in a physical education college. Two indoor and outdoor scenes and different shooting angles were selected to record different volleyball passing behaviors. The input data is to crop each frame into 80×60 grayscale images. The convolution kernels used in the three convolutional layers of the convolutional neural network model are 9×7, 7×7, and 6×4 scales. The convolution kernels used in the two downsampling layers are all 3×3 scales. The input $80 \times 60 \times 9$ volleyball pass video block is finally transformed into a 128-dimensional feature vector. On this basis, all experimental data are divided into two groups, one group is used for deep convolutional neural network training and the other group is used for experimental testing.

4.1. Experimental Results. During the experiment, the dataset used in this article is a nonpublic dataset. Machine vision technology is used to capture volleyball passing training actions, and the captured results are denoised and enhanced to improve the accuracy of the experimental results. In the simulation experiment, quantitative evaluation

adopts the AUC evaluation index, equal error rate (EER) and running time (Time), and other indexes. This paper selects algorithms that have achieved better recognition rates in the above databases for comparisons, such as TCP model [23], AMDN (double fusion) model [24], Motion Energy model [23], SpatioTemporal Convolutional Neural Network (ST-CNN) model [25], and Commotion model [26]. It can be seen from Table 1 that, using frame-level measurement tests, the algorithm in this paper has an advantage in EER and AUC evaluation indicators, and the algorithm has been improved in terms of time-consuming.

Six methods are used to detect volleyball passing training wrong actions on experimental samples, the error rate of different methods of volleyball passing wrong action detection is compared, and the comparison results are used to measure the comprehensive effectiveness of six different methods for detecting volleyball passing training wrong actions. The comparison results are shown in Table 2. Analyzing Table 2 shows that with the continuous increase of the number of experiments, the detection error rate of the method in this paper for volleyball passing training errors has been maintained at a low level. The average error rate of error detection for volleyball passing training in this paper is about 0.027%, which is lower than other methods. When using the method in this paper to detect the wrong action of volleyball passing training, the error can be controlled within a reasonable area.

In order to verify the effectiveness and robustness of this method in the detection of wrong actions in volleyball passing training, four test indicators, ACC (accuracy rate), TPR (sensitivity), FPR (specificity), and PPV (positive prediction rate), are adopted. For quantitative comparison, the specific calculation formula is described as follows:

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}, \tag{16}$$

$$\begin{aligned} \mathrm{TPR} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \\ \mathrm{FPR} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \\ \mathrm{PPV} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \end{aligned} \tag{17}$$

Among them, TP means the number of positive samples, which is actually the number of positive samples; FP means the number of positive samples but actually the number of negative samples; TN means the number of negative samples, which is actually the number of negative samples; FN indicates the number of samples that are judged as negative, but in fact, it is the number of positive samples. The higher the ACC and TPR, the lower the FPR and PPV and the better the detection performance.

Table 3 is the test experiment result of wrong action detection in physical education training. From Table 3, among the six methods of volleyball passing training error detection, the four parameters of ACC, FPR, PPV, and TPR in this method are better than the other five methods, and the detection accuracy of this method has reached more than 95%, Therefore, the experimental results verify the superior performance of the deep convolutional neural network.

4.2. Time Complexity Analysis and Comparison. The time complexity of an algorithm is a function that quantitatively describes the running time of the algorithm.

4.2.1. Time Complexity of a Single Convolutional Layer. Time complexity refers to the computational workload required to execute the algorithm. The time complexity of a single convolutional layer is calculated as follows:

$$\mathrm{Time} \sim O\left(P^2 \cdot Q^2 \cdot C_{\mathrm{in}} \cdot C_{\mathrm{out}}\right), \tag{18}$$

where P represents the side length of each convolution kernel output feature map; Q represents the side length of each convolution kernel; C_{in} represents the number of channels of each convolution kernel, that is, the number of input channels (number of output channels of the previous layer); and C_{out} represents the number of convolution kernels in the convolution layer, that is, the number of output channels.

It can be seen from equation (16) that the time complexity of the convolutional layer is determined by the output feature map area P^2, the convolution kernel area Q^2, the input C_{in}, and the number of output channels C_{out}; the size of the output feature map is in turn determined by the input matrix size X, and the size of the convolution kernel is determined by Q. The and the expression of the side length P of the output feature map is as follows:

$$P = \frac{(X - Q + 2 \times \mathrm{Padding})}{\mathrm{Stride}}. \tag{19}$$

4.2.2. The Overall Time Complexity of the Convolutional Neural Network. The complexity of a single-layer convolutional neural network is calculated by equation (16). The overall time complexity of a convolutional neural network (including multilayer structure) is the sum of the time complexity of each layer. The calculation formula is as follows:

$$\mathrm{Time} \sim O\left(\sum_{l=1}^{D} P_l^2 \cdot Q_l^2 \cdot C_{l-1} \cdot C_l\right), \tag{20}$$

where D represents the number of convolutional layers of the neural network, that is, the network depth; l represents the lth convolutional layer of the neural network; C_1 represents the number of output channels of the lth convolutional layer of the neural network C_{out}, that is, the number of convolution kernels in this layer; the number of input channels of the lth convolutional layer; and X_5 is the number of output channels of the $(l-1)$th convolutional layer.

In terms of time complexity, this article selects the TCP model, AMDN (double fusion) model, motion energy model, spatio-temporal convolutional neural network model, and commotion model to compare with the algorithm in this paper. From equation (12), the time complexity of each algorithm can be calculated. Because the specific parameter data of each algorithm are not clear, this article only calculates which order the time complexity of the algorithm belongs to. The common time complexity

Table 1: AUC and EER used for the frame and pixel-level comparison on the datasets.

Algorithm	EER (%)	AUC	Time (s)
TCP	19.5	0.802	0.34
AMDN	18.6	0.909	0.26
Motion energy	22.1	0.923	0.13
ST-CNN	25.5	0.895	0.52
Commotion	22.8	0.872	0.28
This paper	16.2	0.936	0.12

Table 2: Detection results of wrong actions in physical education training.

Number of experiments/time	Error rate (%)					
	TCP	AMDN	Motion energy	ST-CNN	Commotion	This paper
200	0.092	0.098	0.086	0.132	0.076	0.024
400	0.095	0.108	0.123	0.191	0.085	0.017
600	0.088	0.082	0.092	0.255	0.126	0.036
800	0.097	0.089	0.102	0.234	0.089	0.012
1000	0.086	0.072	0.089	0.145	0.097	0.047

Table 3: Detection results of wrong actions in physical education training.

Test index	ACC	TPR	PPV	FPR
TCP	0.892	0.802	0.142	0.086
AMDN	0.835	0.909	0.264	0.032
Motion energy	0.868	0.923	0.138	0.097
ST-CNN	0.814	0.895	0.129	0.112
Commotion	0.886	0.872	0.282	0.062
This paper	0.976	0.966	0.087	0.013

Table 4: Time complexity of the algorithms.

Algorithm	Time complexity
TCP	$O(n^2)$
AMDN	$O(n^3)$
Motion energy	$O(n^2)$
ST-CNN	$O(n^2)$
Commotion	$O(n^2)$
This paper	$O(n \log n)$

relationship is $O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n^3)$.

As shown in Table 4, the time complexity of this algorithm and other algorithms is mostly on the $O(n \log n)$ order. Analyzing the neural network structure model of other algorithms, the output feature map area P^2, the convolution kernel area Q^2, the input C_{in}, and the output channel number C_{out} are all more complicated than the algorithm in this paper. It can be concluded that the algorithm in this paper is better than other algorithms in terms of time complexity.

5. Concluding Remarks

Since the traditional methods cannot accurately obtain the characteristics of the wrong movements in volleyball passing training, resulting in the decrease of detection accuracy, this paper proposes a method for detecting the wrong movements in volleyball passing training based on a convolutional neural network. The convolutional neural network structure is improved by the nested mlpconv layer. The mixed Gaussian model is used to extract the passing target from the volleyball training video sequence effectively and accurately. The mixed Gaussian model shows robustness in the complex scene background, which can not only successfully detect the volleyball passing training target but also reduce the influence of the small repetitive objects in the background scene on the passing target detection. The nested multilayer mlpconv layer automatically learns the deep-level pass features of the volleyball target that has been extracted, and the improved convolutional neural network reduces the acquisition of redundant information. Experiments show that the method can effectively detect the wrong actions of the athletes during the volleyball passing training process, the detection accuracy is high, the detection error can be effectively controlled, and the wrong actions can be accurately judged in time. And the improved convolutional neural network has excellent generalization performance and nonlinear fitting ability.

Data Availability

All data used to support the findings of the study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Social Science Planning and Research Project of Shandong Province (Grant No. 21CTYJ18).

References

[1] X. Y. Zhang, X. Y. Zhou, M. X. Lin, and J. Sun, "ShuffleNet: an extremely efficient convolutional neural network for mobile devices," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6848–6856, Salt Lake City, UT, USA, June 2018.

[2] A. G. Howard, M. L. Zhu, B. Chen et al., "MobileNets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861v1.

[3] G. Larsson, M. Maire, and G. Shakhnarovich, "FractalNet: Ultra-Deep neural networks without residuals," 2017, https://arxiv.org/abs/1605.07648v4.

[4] D. Y. Han, J. H. Kim, and J. M. Kim, "Deep pyramidal residual networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 6307–6315, Honolulu, HI, USA, July 2017.

[5] J. Hu, L. Shen, G. Sun, and E. Wu, "Squeeze-and-Excitation networks," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, Salt Lake City, UT, USA, June 2018.

[6] D. Chen, P. Wang, L. Yue, Y. Zhang, and T. Jia, "Anomaly detection in surveillance video based on bidirectional prediction," *Image and Vision Computing*, vol. 98, Article ID 103915, 2020.

[7] J. Carreira and A. Zisserman, "Quo Vadis, action recognition? A new model and the kinetics dataset," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4724–4733, Honolulu, HI, USA, July 2018.

[8] F.-N. Yuan, L. Zhang, S. Jin-Ting, X. Xue, and G. Li, "Theories and applications of auto-encoder neural network- s: a literature survey," *Chinese Journal of Computers*, vol. 42, no. 1, pp. 203–230, 2019.

[9] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6479–6488, IEEE, Salt Lake City, UT, USA, June 2018.

[10] L. Li, "Analysis and data mining of intellectual property using GRNN and SVM," *Personal and Ubiquitous Computing*, vol. 24, no. 1, pp. 139–150, 2020.

[11] M. Ravanbakhsh, M. Nabi, H. Mousavi, E. Sangineto, and N. Sebe, "Plug-and-Play CNN for crowd motion analysis: an application in abnormal event detection," in *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Tahoe, NV, USA, March 2018.

[12] D. X. Xu, W. Y. Yan, E. Ricci, and N. Sebe, *Detecting Anomalous E- Vents in Videos by Learning Deep Representations of Appearance and motion*, Elsevier Science Inc, Amsterdam, Netherlands, 2017.

[13] T. Chen, C. Hou, Z. Wang, and H. Chen, "Anomaly detection in crowded scenes using motion energy model," *Multimedia Tools and Applications*, vol. 77, no. 3, pp. 1–16, 2017.

[14] L. Ma, S. Cheng, and Y. Shi, "Enhancing learning efficiency of brain storm optimization via orthogonal learning design," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 11, pp. 6723–6742, 2021.

[15] H. Mousavi, M. Nabi, H. Kiani, P. Alessandro, and M. Vittorio, "Crowd motion monito- ring using tracklet-based commotion measure," in *Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP)*, Quebec City, QC, Canada, September 2015.

[16] J. Y. Ma, F. R. Jie, and Y. J. Hu, "Moving target detection method based on improved Gaussian mixture model," in *Proceedings of the Ninth International Conference on Digital Image Processing (ICDIP 2017)*, Hong Kong, China, July 2017.

[17] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning—Volume 37*, pp. 448–456, Lille, france, July 2015.

[18] R. T. Ionescu, F. S. Khan, M. Georgescu, and L. Shao, "Object-centric auto-encoders and dummy anomalies for abnormal event detection in video," in *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7834–7843, Long Beach, CA, USA, December 2019.

[19] J. Xiao, M. Shen, J. Lei, J. Zhou, R. Klette, and H. Sui, "Single image dehazing based on learning of haze layers," *Neurocomputing*, vol. 389, pp. 108–122, 2020.

[20] Y. Zhou, X. Sun, Z. Zha, and W. Zeng, "MiCT: mixed 3D/2D convolutional tube for human action recognition," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pat- tern Recognition*, pp. 449–458, Salt Lake City, UT, USA, June 2018.

[21] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid net- works for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, Honolulu, HI, USA, July 2017.

[22] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: convolutional block Attention module," in *Proceedings of the Computer Vision—ECCV 2018*, pp. 3–19, Munich, Germany, September 2018.

[23] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 784–799, Munich, Germany, September 2018.

[24] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6154–6162, Salt Lake, UT, USA, June 2018.

[25] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one- stage object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 9627–9636, Seoul, South Korea, October 2019.

[26] Y. Liu, Y. Wang, S. Wang et al., "CBNet: a novel composite backbone network architecture for object detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 11653–11660, 2020.