Research Article

Research on the Recognition Algorithm of Basketball Technical Action Based on BP Neural System

Xiangfeng Hou¹ and Qing Ji²

¹*School of Physical Education, Shanxi University, Taiyuan, Shanxi 030006, China*

²*School of Physical Education, Yanshan University, Qinhuangdao, Hebei 066000, China*

Correspondence should be addressed to Xiangfeng Hou; hxf50006@sxu.edu.cn

Received 29 October 2021; Revised 16 November 2021; Accepted 25 November 2021; Published 31 January 2022

Academic Editor: Bai Yuan Ding

Copyright © 2022 Xiangfeng Hou and Qing Ji. 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.

Vision-based intelligent human action recognition is the most challenging direction in the field of computer vision in recent years. It detects human actions in video sequences, extracts action features and learns action features, and then recognizes human actions in videos. This paper is based on BP neural network's basketball technique action recognition and experimental verification. First, design a basketball technique action recognition method based on BP neural network, analyze basketball actions, collect relevant test data, and divide the methods of basketball action recognition. Finally, analyze the action characteristics and waveform conditions of the upper- and lower-limb movements of the basketball action and analyze the key basketball action recognition data. The designed classification method realizes the effective recognition of basketball actions; then, the basketball recognition method used in this article is experimentally verified, and the feasibility and effectiveness of the recognition method selected in this article are verified by recognizing basketball technical actions, and the experimental results are carried out. Compared with other related studies, this method proposes a division of unit actions to complete the cycle division of basketball actions. The division results do not include the overlap of other actions, avoiding repeated calculations of actions and greatly reducing the amount of calculation of the system. In addition, the method for the recognition of basketball movement includes the separate recognition of upper- and lower-limb movements, comprehensive consideration of arm and leg movements, and a more comprehensive and accurate analysis of basketball movements.

1. Introduction

The development of modern basketball has been popularized all over the world. It not only is a competition between athletes, but also has gradually become a comprehensive competition for technological development in various countries. The breakthrough of various records in basketball matches is not only a manifestation of humans' breakthroughs in physiological limits, but also a manifestation of sports technology innovation. Therefore, the disciplines that intersect with sports science have received more and more attention from scholars. How to improve the competitive level of basketball has become the research focus of biomechanics, psychophysiology, sports medicine, and computer science. Among them, from the perspective of computer science, by extracting various parameters and

action recognition in the training process of athletes and basketball players, it can improve the scientificity of daily basketball skill training. In the study of basketball movement recognition technology based on acceleration sensors, how the sensor collects the smooth and effective acceleration data of basketball gestures is the premise, and which basketball movement recognition method is used is the key to whether the basketball movement can be effectively recognized. As a commonly used deep learning model in the field of action recognition, BP neural network has been widely used in various scenarios, such as signal processing or pattern recognition, the construction of expert systems, and the production of robots. It has good learning performance and recognition accuracy. Based on the advantages of the BP network algorithm, the recognition of basketball technical actions can be better realized. Therefore, this article mainly

studies basketball actions based on BP neural network and validates the recognition method through basketball action experiments.

2. Related Work

With the rapid development of machine learning technology, people's demand for automatic identification and monitoring of various sports image objects is increasing, so as to realize the statistics of sports data and bring more scientific training methods for the improvement of athletes' later skills. Cunado et al. [1] developed a 3D convolutional neural network architecture for human behavior recognition. When viewing motion images, the computer captures the human body motion trajectory and motion trend in real time to determine the position and shape of human body parts, then through the computer analyze the technical characteristics of the action, and report the analysis results to the coach or athlete. Trung et al. [2] proposed a progressive motion detection algorithm based on deep residual network and instance search, focusing on achieving high-precision and fine positioning of athletes' movements. Preece et al. [3] used the distance information between all joints contained in the previous image and each joint point in the current image. The distance information of the joint points corresponding to the reference action is used to describe the athlete's offset characteristics, posture characteristics, and movement characteristics. Shutler et al. [4] used the deep learning algorithm model to study the changes in the load of the hind lower-limb joints and the foot posture of running during long-distance running. Scientific running training method is used for joint damage. It can be seen from this that, with the rapid development of machine learning technology, people's demand for automatic recognition and monitoring of various sports image objects is increasing, so as to realize the statistics of sports data and bring more scientific improvement to the later skills of athletes. The research content of BP neural network is quite extensive, reflecting the characteristics of multidisciplinary and interdisciplinary technical fields. The main research work focuses on the study of biological prototypes, the establishment of theoretical models, the study of network models and algorithms, and the study of BP neural network application systems. It has been widely used in various scenarios, such as signal processing or pattern recognition, the construction of expert systems, and the production of robots. The same is true for basketball. This paper develops a set of visual movement tracking recognition and target detection models in response to the demand for automation of action assessment in basketball sports and uses BP neural network to realize basketball movement recognition.

3. BP Neural Network-Related Theoretical Methods

3.1. BP Neural Network Model

3.1.1. BP Neural Network Model Theory. Using BP network theory for basketball technical action, it can realize any nonlinear mapping from input to output of basketball

technical action. First of all, for the strong ability of nonlinear mapping, the BP network can get the nonlinear mapping relationship between n -dimensional input and m -dimensional output by learning the actions of basketball training; secondly, the generalization ability is strong, and the generalization ability is through the BP network. The training extracts the nonlinear mapping relationship of basketball technical actions hidden in the sample mode and stores it in the weight matrix. When the nonsample data that has not appeared before is input into the trained BP network, the input and output can complete the ability of mapping correctly; finally, the fault tolerance is good. The process of extracting from a large number of basketball technical action samples and counting the law is the process of adjusting the weight matrix. The adjustment of the weight matrix does not depend on the error of individual samples. This is due to the error of all basketballs. The technical action samples contain correct rules, and when there are individual errors in the input samples, it will have little effect on the BP network.

Therefore, a typical BP neuron model is shown in Figure 1. It has n neuron model inputs. All the inputs of this layer are connected to the next layer by connecting weights w , and the final network output can be expressed as $o = f(wx + b)$.

If the transfer function from the hidden layer to the output layer in the BP neural network selects purlin type, any value can output the final output of the layer; and if the transfer function from the hidden layer to the output layer in the BP neural network selects the sigmoid type, the final output layer will be limited to a small range; when solving practical application problems, a detailed analysis of the problem is required to select a suitable transfer function. Generally speaking, the most common application is a single hidden layer network. The single hidden layer perceptron is usually called a three-layer perceptron; that is, the three layers include an input layer, a hidden layer, and an output layer. The schematic diagram of the three-layer BP network is shown in Figure 2 [5].

For the output layer,

$$O_k = f(\text{net}_k) = 1, 2, \dots, l, \quad (1)$$

$$\text{net}_k = \sum_{j=0}^m W_{jk} y_j, \quad k = 1, 2, \dots, l. \quad (2)$$

For the hidden layer,

$$y_j = f(\text{net}_j), \quad j = 1, 2, \dots, m, \quad (3)$$

$$\text{net}_j = \sum_{i=0}^n v_{ij} x_i, \quad j = 1, 2, \dots, m. \quad (4)$$

Generally, the transfer function $f(x)$ is a unipolar sigmoid function

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (5)$$

At this time, the transfer function $f(x)$ is continuous and derivable and satisfies

FIGURE 1: BP neuron model.

FIGURE 2: Three-layer BP neural network.

$$f(x) = f(x)[1 - f(x)]. \quad (6)$$

The main program implementation steps of BP algorithm are the following:

(1) initialization.

Initialize the weight matrix W and V with random values, set the training times counter q and the sample pattern counter p to 1, and set the error E to 0, the learning rate $\eta \in (0,1]$, and use a positive decimal as the network training requirement. The achieved accuracy is $E_{\min}$.

(2) Input the training sample pair and calculate the output of each layer.

Use the current samples X_p and d_p to assign values to the vector arrays X and d and calculate the components of Y and O .

(3) Calculate the output error of the network.

Assuming that there are P pairs of training samples, there are different errors for different sample networks $E^p = \sqrt{\sum_{k=1}^l (d_k^p - o_k^p)^2}$.

In this paper, the root mean square error $E_{\text{RME}} = \sqrt{1/P \sum_{p=1}^P (E^p)^2}$ is used as the total output error of the network.

(4) Calculate the error signal of each layer.

Use the above formula to calculate δ_k^o and δ_j^y .

(5) Modify the weights of each layer.

Use the above formula to calculate the components in W and V . Because the weight adjustment method

described above has low learning efficiency and slow convergence speed, it is easy to form a local minimum but cannot obtain the global optimum. It is learned during training. The new sample has many problems such as forgetting the trend of the old sample. Therefore, in this paper, we choose to use the increasing momentum term in the weight adjustment formula to solve this problem; that is, not only the adjustment of the error gradient descent direction at time t but also the adjustment of the error gradient descent direction before time t is considered.

Assume that the weight matrix of a certain layer is represented by W and the input vector of a certain layer is represented by X ; when the momentum term is added, the expression of the weight adjustment vector is

$$\Delta W(t) = \eta \delta X + \alpha \Delta W(t-1). \quad (7)$$

α is called the momentum coefficient, generally $\alpha \in (0,1)$. In addition, the above problems can also be solved by adaptively adjusting the learning rate or introducing the jitter factor λ .

(6) Check whether all samples have completed a rotation training.

If $p < p$, then p and q increase by 1, and return to step (2); otherwise skip to step (7).

(7) Detect whether the total error of the network meets the accuracy requirements.

When $E_{\text{RME}} < E_{\min}$, the training ends; otherwise, set E to 0 and p to 1 and return to Step (2). The implementation process of BP algorithm is shown in Figure 3.

3.2. Action Recognition. Human body posture recognition is mainly to study the description of human body posture and predict human behavior. The recognition process refers to the process of recognizing human body actions based on changes in the positions of joints in the human body in a designated image or video screen. At present, many scholars have made the human action recognition of inertial sensors the research focus. Through the recognition of human actions by inertial sensors, the research results are applied to wearable devices. Human action recognition is mainly divided into data collection, preprocessing, data division, feature extraction, and model training. The data collection stage refers to the use of inertial sensors to collect the human body's movement signals and physiological signals and observe the human body's movement and physiological changes under the state of motion; the data preprocessing stage refers to the algorithm to denoise and normalize the messy original data through the algorithm. To make it meet the requirements of algorithm identification, the data division stage refers to the selection of individual actions for analysis in the data after the preprocessing is completed; the data feature extraction stage refers to extracting the relevant attribute features to become the data required for the sample location; the final model training stage refers to the

FIGURE 3: BP process algorithm.

algorithm model according to different classification principles to realize data recognition; the final model training stage refers to the realization of data recognition based on algorithm models with different classification principles, as shown in Figure 4.

3.2.1. Detection of Human Moving Targets. In the process of classification and recognition of human walking features, accurately detecting moving targets is a very important step. Target detection refers to first detecting the changed area in the video or image sequence and then separating the detected changed target from the background image. Human motion recognition is used in various scenarios, such as robot navigation, security monitoring, medical image analysis, video image coding, and transmission. Detecting and separating moving targets from images provides great convenience for image processing and feature extraction.

3.2.2. Model and Statistics-Based Feature Extraction. The feature extraction method based on the object model usually uses the geometric relationship of the object or the feature point of the object to estimate. The basic idea is to use a certain geometric model or structure to represent the

structure and shape of the object and extract certain object features in the model and correspondence is established between the images. Then, the parameter information of these models is analyzed, the spatial posture of the object is estimated by geometric or other methods, and finally the image is matched with a template [6–11]. The random field model method is currently the mainstream method in feature extraction, and Markov random field and Gibbs random field are often selected.

At present, some computer vision and identity recognition researchers use model-based methods in the process of target feature extraction. In the experiment of extracting human walking characteristics, Han converted the human body contour data into a motion model, converted a two-dimensional plane into a three-dimensional model, and processed the data through the least square method. Lee et al. used an ellipse to represent the various parts of the human body and then used the ellipse parameters such as the ratio of long and short axes, the center of mass coordinates, and the body structure parameters as the characteristics of gait recognition [12–15]. The swinging process of the lower limbs of the human body is similar to the swinging of the pointer of a clock. Based on this idea, Cunado et al. constructed a pendulum model to extract the angle between the legs and used this as a feature for gait recognition.

4. Basketball Action Recognition Model Based on BP Neural Network

4.1. Feature Extraction of Basketball Technical Action. The human body movements involved in basketball are more complicated. Figure 5 is an analysis of the composition of the basketball posture. The state of the root limbs divides the basketball movement posture into a static state and a movement state [16]. The static state refers to the state where the posture of the limbs remains unchanged, and the state of motion refers to the state of the limbs when performing basketball actions. Shooting, receiving, passing, dribbling and jumping, walking, and running of the lower limbs are defined as unit actions. In the state of motion, the action can be divided into instantaneous action and continuous action according to whether it has periodicity. For this reason, this paper proposes a division method based on unit action extraction.

4.1.1. Movement Status Division. Dispersion is an index that can be used to reflect the degree of difference between observed variable values, and dispersion is an index that indicates the difference between sensor signal sample values. Taking angular velocity as an example, ω_n^x represents the n th angular velocity value in the x -axis direction, ω_{n-1}^x represents the $n-1$ th angular velocity data in the x -axis direction, and d_n^x represents the sensor at the n th and previous moments. The angular velocity difference in the x -axis direction at a certain moment where there exists the dispersion index ω_{n-1}^x can be obtained by the following formula:

$$d_n^x = |\omega_n^x - \omega_{n-1}^x|. \quad (8)$$

Angular velocity and acceleration data are included in the motion data package. Comprehensive consideration of

FIGURE 4: Flow chart of the human body gesture recognition method.

the data characteristics of each sensor is an important step to accurately divide the movement. d_n^x represents the dispersion of the acceleration sensor data at the n th time, d_n^g represents the dispersion of the angular velocity sensor data at the n th time, and $d_n^{a_x}$, $d_n^{a_y}$, $d_n^{a_z}$, $d_n^{g_x}$, $d_n^{g_y}$, and $d_n^{g_z}$, respectively, represent the dispersion of the acceleration and angular velocity of each axis; then, d_n^a and d_n^g can be obtained by the following formulas, respectively:

$$d_n^a = d_n^{a_x} + d_n^{a_y} + d_n^{a_z}. \quad (9)$$

$$d_n^g = d_n^{g_x} + d_n^{g_y} + d_n^{g_z}. \quad (10)$$

In the case of sports, the changes of the athletes' actions will be reflected in the sensor data in real time. Since the dispersion reflects the degree of difference between the sensor data, the division of the athlete's physical state can refer to the characteristics of the dispersion. In static conditions, the angular velocity and acceleration dispersion are kept below the threshold values λ_g and λ_a in sequence. The state of the athlete's limbs at the n th moment is represented by γ_n . When γ_n is 0, it indicates the static state, and when γ_n is 1, it indicates the motion state. The definition is shown in

$$\gamma_n = \begin{cases} 0, & d_n^a < \lambda_a \text{ and } d_n^g < \lambda_g \\ 1, & d_n^a \geq \lambda_a \text{ and } d_n^g \geq \lambda_g \end{cases} \quad (11)$$

The degree of data dispersion of different sensors is calculated, and different motion states can be distinguished and identified for different thresholds. At the same time, the instantaneous action and continuous action can be distinguished by whether the angular velocity of each sensor changes periodically. It can be known that the extraction of basketball motion state data can be achieved through motion state division.

4.1.2. Unit State Division. Studies have shown that the division of motion can be achieved with the help of the motion data of the legs and arms. In describing the angle change in the process of rigid body movement, the angular velocity has a very good performance. Therefore, the angular velocity can be used to divide the data. In reducing the influence of external noise, the Kalman filter algorithm can be used to fuse acceleration, magnetic field strength, and angular velocity data to effectively solve the problem.

4.1.3. Feature Extraction of Basketball Posture. After data division, the unit action data is obtained, which is composed of acceleration and angular velocity, a_n^x , a_n^y , and a_n^z represent the acceleration of the three axes of the n th sampling point, and g_n^x , g_n^y , and g_n^z indicate the angular velocity of the three axes at the n th sampling point. Use v and g_n to denote the acceleration vector sum and the angular velocity vector sum, respectively, which can be obtained by the following formulas:

$$a_n = \sqrt{(a_n^x)^2 + (a_n^y)^2 + (a_n^z)^2}. \quad (12)$$

$$g_n = \sqrt{(g_n^x)^2 + (g_n^y)^2 + (g_n^z)^2}. \quad (13)$$

An eight-dimensional vector can be constructed through the four parameters of three-axis acceleration, three-axis angular velocity, combined acceleration, and combined angular velocity, where the sampling point of each action is represented by N , and an $N \times 8$ -dimensional matrix can form a sample, which can be based on the fact that each sample calculates the characteristics of each dimension. The time domain features in this experiment are composed of mean

FIGURE 5: The composition of motion posture in basketball.

and variance. μ_a and δ^2 are used to represent the mean and variance of a certain component of acceleration in a unit action. It can be calculated by the following formulas, where a is the acceleration of a certain component:

$$\mu_a = E(a) = \frac{1}{N} \sum_{i=1}^N a_i. \quad (14)$$

$$\delta^2 = \frac{1}{N} (a_i - \mu_a)^2. \quad (15)$$

The peak value of the discrete Fourier transform and its corresponding frequency is the performance of frequency domain characteristics. The process of converting the signal from the time domain to the frequency domain can be realized by the discrete Fourier transform method. Among them, $S_{\text{DFE}}(n)$ represents the Fourier transform result of the n th sampling point, and j represents the imaginary part calculated by the following formula:

$$S_{\text{DFE}}(n) = \sum_{i=1}^{N-1} a_i e^{-j2\pi/Nin}. \quad (16)$$

From the Fourier transform result, $S_{\text{DFE}}(K)$ represents the peak value, the sampling point K corresponds to the Fourier transform peak value, and the Fourier transform frequency f is obtained by the following calculation formula, where the sampling frequency is represented by f_s :

$$f = K \times \frac{f_s}{N}. \quad (17)$$

Through feature calculation, the time domain and frequency domain features of each dimension data in the

sample can be obtained, thereby constructing a 32-dimensional feature vector, as shown in Tables 1 and 2.

In the experiment, sensor nodes are used to detect the movement information of different limbs to fix them on the forearms and calves of the experimenter. Due to the different placement of nodes, the division of upper- and lower-limb action sets of the data set is also different. Therefore, in order to further realize the movement division of the upper and lower limbs, it is necessary to establish different sample classifiers for the data set. Combining the results of the upper and lower limbs, the basketball posture of the current candidates can be obtained [17, 18].

4.2. BP Network Design

4.2.1. Preparation of Training Sample Set. First, the input and output variables of the modeling system should be screened. Usually, the goal to be achieved is selected as the output variable [5, 19]. The variables that have a large impact on the output, easy to extract, and have little correlation should be selected as the input variables of the network. Use samples with or without a certain input variable to train the network and compare the effects to determine whether the variable is suitable as an input variable. After selecting the input and output variables, they must be represented. From the perspective of the nature of input and output, it can be divided into two types: language variables and numerical variables. Variables expressed in natural language are linguistic variables, and various attributes of things represented by natural language are their “linguistic values.” The sample data that has undergone scale transformation and distribution transformation can also be used for

TABLE 1: Time domain features of data.

Feature	Feature ID	Feature description
Time domain characteristics	1 ~ 3	Acceleration sensors x , y , and z axis data mean value
	4	Mean value of total acceleration
	5 ~ 7	The mean value of the gyroscope's x , y , z axis data
	8	Mean value of closing angular velocity
	9 ~ 11	Acceleration sensors x , y , and z axis data variance
	12	Variance of total acceleration
	13 ~ 15	Variance of the total velocity: the variance of the gyroscope's x , y , and z axis data
	16	Variance of angular velocity

TABLE 2: Frequency domain features of data.

Feature	Feature ID	Feature description
Frequency domain analysis	17, 18	The DFT peak value of the acceleration sensor x data and the frequency of the corresponding peak value
	19, 20	The DFT peak value of the acceleration sensor y data and the frequency of the corresponding peak value
	21, 22	The DFT peak value of the acceleration sensor z data and the frequency of the corresponding peak value
	23, 24	The DFT peak value of the combined acceleration and the frequency of the corresponding peak value
	25, 26	The DFT peak value of the gyroscope x data and the frequency of the corresponding peak value
	27, 28	The DFT peak value of the gyroscope y data and the frequency of the corresponding peak value
	29, 30	The DFT peak value of the gyroscope z data and the frequency of the corresponding peak value
	31, 32	The DFT peak value of the gyroscope x data and the frequency of the corresponding peak value

network training. After transformation processing, the input and output data of the network are limited to the interval of $[-1,1]$ or $[1,0]$. This process is called scale transformation, which can make the positions of various variables closer. The commonly used transformation formula is

$$\overline{x_i} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}. \quad (18)$$

$x_{\min}$ is the minimum value of the variable; $x_{\max}$ is the maximum value of the variable; x_i is input or output data

Normalization is a linear transformation, and when the distribution of the sample is not ideal, the distribution transformation is needed to change the sample. This distribution law generally uses logarithmic transformation, as well as square root and cube root transformation. The distribution transformation is a nonlinear transformation. The training sample set also needs to be prepared. The information capacity of the network may affect the classification ability of the BP neural network. The information capacity of the network is characterized by the total number of network weights and thresholds n_w . Generally, the matching relationship between the number of training samples P and the given training error ε is

$$P \approx \frac{n_w}{\varepsilon}. \quad (19)$$

There is a reasonable matching relationship between the information capacity and the number of training samples in the network. When solving the problem of less actual training sample data, it is difficult to meet the above requirements. For a certain number of samples, a suitable network parameter is required. Too few cannot express all the hidden laws, and too many samples will not be fully

trained. Based on experience, the number of training samples can be 5–10 times n_w . After determining the number of sample sets, the samples need to be selected and organized. The samples should be representative and balanced in categories; when organizing the samples, the input samples should be randomly selected from the training sample set, or samples of different categories should be cross-input.

4.2.2. Design of BP Network Structure. In the BP network structure design, it is necessary to design the number of nodes in the input layer and the output layer. At the same time, the design of the number of hidden layers and the number of hidden nodes of the network is also particularly important. The number of nodes in the input layer and output layer can be determined according to actual application requirements. The number of nodes in the input layer is usually obtained through the dimensions of the training sample vector; the number of nodes in the output layer is usually used as the dimension of the output space of the approximation function in the approximation network and is usually used as the number of categories in the classification network. Theoretical analysis shows that a network with an S-shaped hidden layer plus a linear input layer can approximate any rational function. Increasing the number of layers can improve accuracy and reduce errors, thereby making the network more complex. Therefore, in actual demand applications, usually consider designing a network with hidden layers.

The number of samples in training, the complexity of the hidden law, and the size of the noise can determine the number of hidden nodes. Since it is difficult to grasp the complexity of the sample rule and the size of the noise in the actual process, it is difficult to design the number of hidden

FIGURE 6: BP neural network recognition process. (a) BP neural network training process. (b) BP neural network recognition process.

nodes. In determining the optimal number of hidden nodes, the same sample set can be used to train networks with different numbers of hidden nodes, because the network error is the smallest at this time. The number of hidden nodes can be roughly estimated by the following formulas:

$$m = \sqrt{n+1} + \alpha, \quad (20)$$

$$m = \log_2 n, \quad (21)$$

$$m = \sqrt{n}, \quad (22)$$

$$m = \sqrt{0.43nl + 0.12l^2 + 0.51 + 2.54n + 0.77l + 0.35}, \quad (23)$$

M is the number of hidden nodes; n is number of output layer nodes; l is number of output layer nodes; α takes a constant between 1– and 10.

4.2.3. Design of Parameters. Under normal circumstances, a small random value is selected to assign the initial weight, which not only ensures that the input value is small and can work in the area where the slope of the excitation function changes greatly, but also can avoid the unreasonable and unlimited increase of the absolute value of some weights after multiple continuous learning and training. The initial weight is generally a random number of $(-1, 1)$. The learning rate is generally smaller in order to ensure the stability of the system. A large learning rate may make the system unstable, but a small learning rate makes the training time longer. In order to ensure that the error value of the network does not jump out of the bottom of the error surface and eventually tends to the minimum error, the learning rate is within the range of 0.01–1. The recognition process of BP network model basketball technical action recognition based on BP neural network mainly includes two aspects, namely, the training process of BP neural network and the basketball technical action recognition process of BP neural network. The basketball technical action recognition process of BP neural network is shown in Figure 6.

First, collect a lot of sample data amounts to train the designed BP neural network. After reading the training sample data, the sample data is preprocessed to extract the

acceleration characteristic value of the gesture action, and the BP neural network is trained under the guidance of the expected output (teacher signal) to achieve the preset target accuracy. The trained BP neural network can then be used to recognize the samples to be recognized for predefined basketball technical actions. Of course, the sample to be recognized also needs to be preprocessed to extract the acceleration characteristic value of the gesture action, then use the trained BP neural network to recognize the technical action, and finally output the recognition result.

5. Experimental Design and Result Analysis

5.1. Experimental Design. The entire data collection process is mainly to collect basketball's dribbling, running, standing dribbling, catching, shooting, passing, jumping, and other actions, as shown in Figure 7; each action is effective and repeated for 50 seconds per rate. A total of 5650 samples were collected, of which 3000 were upper-limb basketball movements and 2,650 were lower-limb basketball movements. The sampled movements were completed in strict accordance with regulations. The number of specific movements was recorded by the scorer, as shown in Table 3.

5.2. Result Analysis and Discussion. From the results reflected in Tables 4 and 5, it can be clearly seen that the BP neural network has a good recognition effect in the recognition of upper-limb movements and lower-limb movements. Among them, the recognition accuracy rates of the entire upper and lower extremity basketball technical movements are as high as 93.2% and 99.2%, respectively, and the average recall rates of the upper and lower extremity basketball technical movements are as high as 93.2% and 99.2%, respectively. The recognition accuracy rates are as high as 93.2% and 99.2%, and the average recall rates of upper and lower limbs are 93.2% and 99.2%, respectively. Regardless of the algorithm, the accuracy of upper- and lower-limb basketball actions is, respectively, 97% and 84.9%. As for the recognition rate of upper-limb basketball action which is lower than that of lower-limb basketball action, the main reason is that upper-limb

FIGURE 7: Nine kinds of basketball moves.

TABLE 3: The number of different movements of the upper and lower limbs.

Upper and lower limbs	Behavior	Quantity
Arm movements	Dribble (standing)	500
	Dribble (walking)	500
	Dribble (running)	500
	Shot	500
	Pass	500
	Catch the ball	500
Leg movements	Run	500
	Jump	500
	Shot	500
	Dribble (running)	600
	Dribble (walking)	550

TABLE 4: Classification results of upper-limb movements by different classification algorithms.

Behavior	C4.5 Accuracy	C4.5 Recall rate	SVM Accuracy	SVM Recall rate	NB Accuracy	NB Recall rate	BP-ANN Accuracy	BP-ANN Recall rate
Catch the ball	0.937	0.949	0.964	0.977	0.899	0.928	0.964	0.979
Pass	0.909	0.931	0.978	0.989	0.935	0.954	0.982	0.979
Shot	0.924	0.910	0.986	0.992	0.956	0.918	0.980	0.984
Dribble (walking)	0.888	0.860	0.929	0.920	0.793	0.944	0.938	0.919
Dribble (standing)	0.788	0.770	0.819	0.570	0.780	0.704	0.910	0.881
Dribble (running)	0.781	0.805	0.693	0.863	0.805	0.760	0.881	0.898
Average	0.862	0.862	0.875	0.868	0.851	0.851	0.935	0.935

TABLE 5: Classification results of lower-limb movements by different classification algorithms.

Behavior	C4.5 Accuracy	C4.5 Recall rate	SVM Accuracy	SVM Recall rate	NB Accuracy	NB Recall rate	BP-ANN Accuracy	BP-ANN Recall rate
Jump	0.998	0.995	0.985	0.999	0.981	0.979	0.986	0.999
Run	0.984	0.982	0.999	0.980	0.960	0.979	0.998	0.986
Walk	0.974	0.974	0.990	0.995	0.982	0.960	0.994	0.995
Average	0.983	0.983	0.991	0.991	0.972	0.973	0.994	0.994

TABLE 6: Classification results of lower-limb movements by different classification algorithms.

Behavior	C4.5 Accuracy	C4.5 Recall rate	SVM Accuracy	SVM Recall rate	NB Accuracy	NB Recall rate	BP-ANN Accuracy	BP-ANN Recall rate
Catch the ball	0.945	0.947	0.969	0.975	0.865	0.930	0.986	0.985
Pass	0.920	0.940	0.978	0.991	0.945	0.960	0.991	0.991
Dribble	0.985	0.986	0.998	0.991	0.998	0.986	0.997	0.994
Shot	0.946	0.925	0.987	0.995	0.958	0.910	0.976	0.986
Average	0.964	0.964	0.990	0.990	0.965	0.965	0.991	0.991

movement is relatively more complicated and some actions have similarities, as shown in Table 6.

From the results reflected in Tables 4 and 5, it can be clearly seen that the BP neural network has a good recognition effect in the recognition of upper-limb movements and lower-limb movements. The recognition accuracy rates are as high as 93.2% and 99.2%, and the average recall rates of upper and

lower limbs are 93.2% and 99.2%, respectively. Regardless of the algorithm, the accuracy of upper- and lower-limb basketball actions is, respectively, 97% and 84.9%. As for the recognition rate of upper-limb basketball action which is lower than that of lower-limb basketball action, the main reason is that upper-limb movement is relatively more complicated and some actions have similarities, as shown in Table 6.

FIGURE 8: BP-ANN's recognition results of basketball gestures.

Experiments show that the best way to recognize upper- and lower-limb movements in basketball is the BP artificial neural network method. After the BP artificial neural network establishes the classifier of the basketball movement of the upper and lower limbs, it finally and effectively recognizes the basketball movement, as shown in Figure 8. It can be seen from the figure that the basketball movement recognition model established by the BP neural network is recognized. The accuracy rate can exceed 95%, and the recognition rate of the entire basketball action is as high as 98.85%.

6. Conclusion

On the basis of studying and summarizing the advantages and disadvantages of existing basketball action recognition methods, this paper designs a basketball action recognition method based on BP neural network. BP network has distributed storage and parallel processing of information, strong nonlinear mapping ability, and robustness. The advantages of good performance, fault tolerance, and strong generalization ability have successfully reduced the complexity of the traditional action recognition algorithm implementation process, improved the self-organization and self-adaptability of the algorithm implementation, and improved the recognition rate. On the basis of the above-mentioned research work, the feasibility of the recognition algorithm selected in this paper is verified through experiments.

Through analyzing the characteristics of basketball actions and aiming at the characteristics of basketball actions, it is proposed to divide the basketball action data into two stages. After the entire basketball action is fully recognized, each action unit is extracted for separate analysis, and finally test the characteristic data of 32 basketball technical movements, classify the samples according to the data obtained from the upper and lower limbs, construct a basketball movement classifier based on the technical movement data of the upper and lower limbs, identify the movements according to the classification algorithms commonly used by experts and scholars, and finally construct a most suitable action. Through a series of test actions, it is shown that the BP artificial neural network has the best recognition effect on the upper and lower limbs of basketball

actions, the overall accuracy rate is controlled above 99%, and the recognition accuracy of the movement posture in basketball is 98.85%. It shows that the effect of the basketball gesture recognition method used in this article is quite satisfactory. This research will be of great significance to the intelligent training and technical correction of basketball players in the future.

Data Availability

The dataset can be accessed upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

- [1] D. Cunado, M. S. Nixon, and J. N. Carter, "Using gait as a biometric, via phase-weighted magnitude spectra//I," in *Proceedings of the First International Conference on Audio and Video-Based Biometric Person Authentication*, pp. 93–102, Springer-Verlag, Crans-Montana, Switzerland, 12 March 1997.
- [2] N. T. Trung, Y. Makiyara, H. Nagahara, Y. Mukaigawa, and Y. Yagi, "Inertial-sensor-based walking action recognition using robust step detection and inter-class relationships[C]," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pp. 3811–3814, Tsukuba International Congress Center, Tsukuba, Japan, 11 November 2012.
- [3] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, and D. Howard, "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 3, pp. 871–879, 2009.
- [4] J. D. Shutler, M. S. Nixon, and C. J. Harris, "Statistical gait recognition via velocity moments," in *Proceedings of the IEEE Colloquium on Visual Biometrics*, IEEE, London, UK, March 2000.
- [5] L. Zhao, Y. Jiao, J. Chen, and R. Zhao, "Image style transfer based on generative adversarial network," in *Proceedings of the 2021 International Conference on Computer Network, Electronic and Automation (ICCNEA)*, pp. 191–195, IEEE, Xi'an, China, September 2020.
- [6] L. A. Schwarz, A. Bigdelou, and N. Navab, "Learning gestures for customizable human-computer interaction in the operating room," in *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2011*, pp. 129–136, Springer, Berlin, Germany, 2011.
- [7] C. K. Lim, Z. Luo, I.-M. Chen, and S. H. Yeo, "Wearable wireless sensing system for capturing human arm motion," *Sensors and Actuators A: Physical*, vol. 166, no. 1, pp. 125–132, 2011.
- [8] A. V. Dowling, J. Favre, and T. P. Andriacchi, "Inertial sensor-based feedback can reduce key risk metrics for anterior cruciate ligament injury during jump landings," *The American Journal of Sports Medicine*, vol. 40, no. 5, pp. 1075–1083, 2012.
- [9] F. Albinali, S. Intille, W. Haskell, and M. Rosenberger, "Using wearable activity type detection to improve physical activity energy expenditure estimation[C]," in *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pp. 311–320, ACM, Copenhagen, Denmark, 26 September 2010.

- [10] B. Bhanu and J. Han, "Individual recognition by kinematic-based gait analysis," *Proceedings - International Conference on Pattern Recognition*, vol. 3, pp. 343–346, 2012.
- [11] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer, "The humanID gait challenge problem: data sets, performance, and analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 162–177, 2005.
- [12] J. B. Hayfron-Acquah, M. S. Nixon, and J. N. Carter, "Automatic gait recognition by symmetry analysis," *Pattern Recognition Letters*, vol. 24, no. 13, pp. 2175–2183, 2003.
- [13] A. Taniguchi, K. Watanabe, and Y. Kurihara, "Measurement and analyze of jump shoot motion in basketball using a 3-d acceleration and gyroscopic sensor[C]," in *Proceedings of the SICE annual conference (SICE)*, pp. 361–365, Akita, Japan, 20 August 2012.
- [14] G. M. Paul, B. P. David, and L. M. Clare, "The physical and physiological demands of basketball training and competition [J]," *International Journal of Sports Physiology and Performance*, vol. 5, no. 1, pp. 75–86, 2010.
- [15] M. Zhang and A. A. Sawchuk, "A feature selection-based framework for human activity recognition using wearable multimodal sensors[C]," in *Proceedings of the 6th International Conference on Body Area Networks*, pp. 92–98, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Beijing, China, 7 November 2011.
- [16] L. N. N. Nguyen, D. Rodríguez-Martín, A. Català, C. Pérez-López, A. Samà, and A. Cavallaro, "Basketball activity recognition using wearable inertial measurement units[C]," in *Proceedings of the XVI International Conference on Human Computer Interaction*, pp. 60–67, Vilanova i la Geltrú, Vilanova i la Geltru Spain, 7 September 2015.
- [17] A. D. Ignatov and V. V. Strijov, "Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer," *Multimedia Tools and Applications*, vol. 75, no. 12, pp. 7257–7270, 2016.
- [18] H.-C. Lin, S.-Y. Chiang, K. Lee, and Y.-C. Kan, "An activity recognition model using inertial sensor nodes in a wireless sensor network for frozen shoulder rehabilitation exercises," *Sensors*, vol. 15, no. 1, pp. 2181–2204, 2015.
- [19] B. Longstaff, S. Reddy, and D. Estrin, "Improving activity classification for health applications on mobile devices using active and semi-supervised learning[C]," in *Proceedings of the Pervasive Computing Technologies for Healthcare (Pervasive-Health), 2010 4th International Conference on-NO PERMIS-IONS*, pp. 1–7, IEEE, Munich, Germany, April 2010.