International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18 (2026) pp. 1-16

https://doi.org/10.70917/ijcisim-2026-2362
Article

Practical Approaches to Integrating Smart Technology into Physical Education in Higher Education

Lizhang Cheng [1,*]

[1] College of Physical Education and Sports, Jining Normal University, Ulanqab, Inner Mongolia, 012000, China
* Correspondence author: clz2026@126.com

Abstract: The deep integration of artificial intelligence technology into the physical education curriculum in higher education institutions has significantly accelerated the transition of physical education toward intelligent systems. This paper employs the OpenPose human pose estimation algorithm to extract skeletal joint points from sports videos, and preprocesses the extracted skeletal data by addressing missing and outlier points using mean imputation and exponential smoothing. Based on this, an ST-GCN network model—fusing graph-convolutional neural networks with spatiotemporal attention mechanisms—was constructed to classify sports movements, and the identified movements were evaluated using the DTW algorithm. The results show that the average recognition accuracy of the proposed method for sports movements on the training and test sets was 90.67% and 88.42%, respectively. The model features a significantly reduced number of parameters and offers real-time processing capabilities, outperforming traditional methods. Furthermore, the feedback provided through motion evaluation effectively helps users improve motion quality and meet standard requirements, thereby comprehensively enhancing the quality and efficiency of physical education instruction. This study aims to provide theoretical references and practical approaches for leveraging artificial intelligence to facilitate the systematic and scientific development of higher education teaching systems.

Keywords: OpenPose human pose estimation; Graph-convolutional neural network; Spatio-temporal attention mechanism; DTW algorithm; Higher education physical education

1. Introduction

The essence of physical education in higher education lies in enhancing students' physical fitness, developing athletic skills, and fostering well-rounded personalities. As modern technology permeates the field of education, smart technologies are quietly reshaping the traditional landscape of physical education, bringing unprecedented opportunities for transformation and development [1]. This integration does not replace teachers; rather, it empowers educators through smart technologies, enabling them to tap into students' potential in a more scientific and precise manner and optimize the teaching process.

First, with the help of smart sensor technology, physiological data such as running speed, stride length, and heart rate can be accurately collected during physical education classes. Based on this data, teachers can develop personalized training plans for students, achieving integration between physical education and data monitoring and personalized guidance [2-3]. In this regard, Reference [4] designed a personalized fitness training recommendation system based on motion sensors and data mining technology. Experiments demonstrated that the system can effectively collect and process physical education data to provide students with personalized exercise plans, exhibiting high practicality and accuracy. Reference [5] proposed AI-based wearable sensors; their application in sports monitoring facilitates personalized training guidance based on students' physical data. Reference [6] established a model based on deep learning algorithms, recommendation algorithms, and sensor data. This model

collects human movement data via sensors and then analyzes and processes the data using recommendation algorithms to generate personalized physical training plans. Reference [7] examined the impact of wearing AI-integrated wearable training sensors on athletes' performance, physiological efficiency, and injury risk reduction. Comparative experiments demonstrated that wearable sensors can significantly enhance exercise physiological efficiency and reduce the risk of training-related injuries through personalized training. Reference [8] reviews research progress on the application of smart sensors in the field of sports science, particularly the latest achievements, challenges, and future directions in the deployment of current smart sensor technologies, and analyzes their impact on sports science and athlete development. Reference [9] points out the shortcomings of traditional methods for evaluating performance in physical education and proposes a solution based on a real-time monitoring system. By combining wearable biosensor technology with advanced algorithms, this approach helps transform physical education methods and improve participation and performance outcomes. Reference [10] proposes a smart sensor-based automatic monitoring and recognition method to achieve the automatic monitoring of physical fitness indicators during sports training. It verifies that this method facilitates the automatic monitoring of physical performance metrics during training, thereby enabling personalized training.

Second, virtual reality (VR) technology can be used to create realistic sports training scenarios, such as simulating environments for alpine skiing and outdoor rock climbing. Students can experience different sports scenarios in a safe classroom environment as if they were actually there, enhancing the sense of immersion. This represents the integration of VR technology with physical education in terms of scenario creation [11–12]. Regarding the application of VR technology, Reference [13] designed and proposed a university physical education virtual reality system comprising the Internet of Things (IoT), a cloud platform, and a mobile client. By delivering virtual reality experiences, this system has demonstrated excellent application and promotion outcomes, providing a scientific reference for physical education reform in higher education. Reference [14] utilized VR technology to construct a virtual physical education teaching environment platform and, through experiments, verified that the platform helps reduce safety incidents during physical activities while increasing students' interest in participating in physical education. Reference [15] reviews the current state of research on VR technology in physical education and, by integrating Web application design with VR technology, constructs an extraction model. Experimental results demonstrate that incorporating VR technology into physical education is significant for enhancing students' learning interest and enthusiasm for sports. Reference [16] emphasizes that the integration of VR technology into university physical education helps prevent sports injuries, push training boundaries, and improve teaching effectiveness. Reference [17] examines the innovative application of VR technology in university physical education, designing and implementing an immersive experience system aimed at enhancing students' motor learning outcomes and training quality through advanced 3D sports training support systems and motion recognition algorithms. Reference [18] discusses researchers' perspectives on the application of VR in physical education. The findings indicate that overcoming the inherent challenges associated with VR technology in educational settings can enrich the teaching process; however, further research on the application of VR in educational contexts is needed. Reference [19] integrates VR technology to explore a novel physical education game-based teaching model in universities, aiming to improve students' exercise efficiency and subjective motivation. Through comparative testing, the study validated the effectiveness and superiority of this VR-integrated physical education game-based teaching model. Reference [20] analyzed the impact of VR-assisted physical education on student performance; correlation analysis indicated that VR-assisted training helps improve students' athletic performance and training quality.

In addition, regarding the management of physical education teaching resources, the system leverages big data analytics to integrate various types of physical education materials, such as instructional videos, lesson plans, and training programs. Based on factors such as students' age, gender, and physical fitness level, and in accordance with the educational psychology principle of teaching tailored to individual needs, the system precisely delivers appropriate learning resources [21–22]. For example, Reference [23] utilized multimedia technology to construct an interactive teaching system based on resource delivery to support physical education instruction, addressing the shortcomings of traditional teaching models. The study validated that this system facilitates the delivery of personalized teaching resources tailored to individual student needs, thereby enhancing learning outcomes. Reference [24] discusses the application of cloud computing in the management of physical education teaching resources in higher education institutions and conducts research on physical education teaching resources in conjunction with virtualization technology. The results indicate that cloud computing improves the utilization rate of physical education teaching resources and helps universities manage and integrate teaching resources more effectively. Reference [25] identifies issues of resource

recommendation imbalance and low resource credibility in traditional online physical education resource recommendations. It proposes a trust-based balanced recommendation algorithm for online physical education resources and verifies that this algorithm improves resource balance and yields recommendations with higher credibility. Reference [26] examined the impact of AI-optimized dynamic physical education teaching materials on physical education instruction. Comparative experiments demonstrated that AI-assisted physical education teaching resources can enhance learning outcomes and user satisfaction, while also opening new avenues for innovation in physical education. Reference [27] proposed a personalized recommendation method based on a cognitive diagnostic model to improve the security of physical education teaching resource recommendations in universities and reduce test overlap and resource exposure rates. Experiments showed that this method enhances the security and accuracy of teaching resource recommendations.

This paper introduces artificial intelligence recognition technology into physical education in higher education. First, the OpenPose human pose estimation model is used to extract skeletal joint data from RGB video footage and preprocess the pose data. Next, considering the varying degrees of reliance on skeletal landmarks across different sports, we propose a sports motion recognition algorithm based on a spatio-temporal convolutional neural network (STCNN) that incorporates a spatio-temporal attention mechanism. Next, an action evaluation method based on DTW (Dynamic Time Warp) distance is employed to objectively assess the similarity between the performed sports movements and template movements, and to provide movement feedback. Finally, experiments are conducted on both public and collected datasets to investigate the effectiveness of the motion recognition model and to evaluate the extent to which the motion evaluation feedback improves the quality of students' sports movements.

2. Sports Pose Estimation and Data Preprocessing

Due to factors such as changes in lighting, occlusions, and variations in camera resolution during video recording, OpenPose may encounter issues such as missing or undetected joints when extracting the skeleton. These issues primarily manifest as missing points and outliers.

2.1. Human Sports Pose Estimation Based on OpenPose

Using the OpenPose pose estimation method, the system extracts 18 skeletal joints from the video and detects the 2D pose of a person in the image in real time by encoding the relationships between joint positions and limb orientations. The OpenPose network employs a multi-stage, cascaded, dual-branch architecture, as illustrated in Figure 1. This framework utilizes the VGG-19 network to extract feature maps F. In the first stage, feature map F is fed into the dual-branch network to extract joint locations (PAM) and the Affinity Vector Field (PAF). These two predicted features are then fused with the original feature map F and fed into the next stage. Through multiple iterations, PAM and PAF that meet the requirements are obtained. Finally, the Hungarian algorithm is used to determine the joint connections that maximize the total weight of all body parts, thereby deriving the pose information corresponding to a single human body.

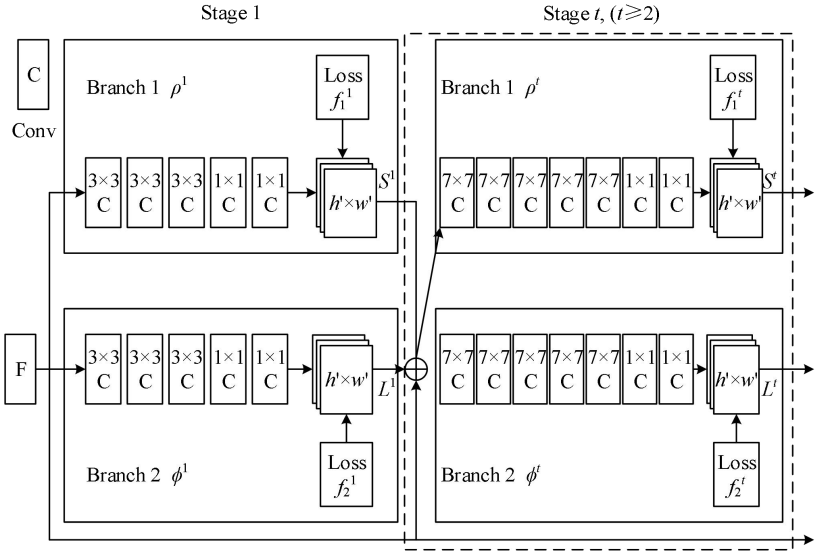


Figure 1. OpenPose network structure

2.2. Handling Missing Points

Missing joint points primarily manifest in two ways: 1) missing joint points between adjacent frames; and 2) missing joint points across multiple consecutive frames. To mitigate the impact of missing joint points on recognition accuracy, we assume that joint points move at a constant speed over short time intervals and interpolate the missing joint point coordinates using the average of the joint point coordinates from adjacent image frames.

(1) Missing vertices between adjacent frames. Suppose that the coordinates $P_i^j\left(x_i^j, y_i^j\right)$ of the j th vertex in frame i are missing. To fill in missing joint coordinates by calculating the average of the joint coordinates in adjacent image frames, the method for calculating the missing point $P_i^j\left(x_i^j, y_i^j\right)$ is as follows:

$$x_i^j = \frac{x_{i+1}^j + x_{i-1}^j}{2}, y_i^j = \frac{y_{i+1}^j + y_{i-1}^j}{2} \tag{1}$$

(2) Missing keypoints between consecutive frames. Suppose the coordinates of the j th keypoint $P_{i-1}^j\left(x_{i-1}^j, y_{i-1}^j\right)$, $P_i^j\left(x_i^j, y_i^j\right)$, $P_{i+1}^j\left(x_{i+1}^j, y_{i+1}^j\right)$ are missing in consecutive frames $i-1$, i, and $i+1$. In this case, the missing keypoint coordinates at the intermediate position are filled in by calculating the average of the keypoint coordinates in the frames immediately preceding and following the missing frame. The missing point $P_i^j\left(x_i^j, y_i^j\right)$ is calculated as follows:

$$x_i^j = \frac{x_{i+2}^j + x_{i-2}^j}{2}, y_i^j = \frac{y_{i+2}^j + y_{i-2}^j}{2} \tag{2}$$

Similarly, the coordinates of the missing points $P_{i-1}^j\left(x_{i-1}^j, y_{i-1}^j\right)$ and $P_{i+1}^j\left(x_{i+1}^j, y_{i+1}^j\right)$ can be determined. Use the method described above to fill in the missing points.

2.3. Exception Handling

To address outliers and spikes in the dataset, exponential smoothing is used to smooth the data. The calculation method is as follows:

$$S_t = ay_t + (1-a)S_{t-1} \tag{3}$$

In the equation: S_t represents the smoothed value of the coordinate data t; y_t represents the true value of the coordinate data t; S_{t-1} represents the smoothed value of coordinate data $t-1$; a is the smoothing constant, set to 0.6. Due to the inherent characteristics of video data, unprocessed raw coordinate data suffers from missing coordinates and severe jitter over time. This leads to missing features during feature extraction and sequence modeling, thereby affecting recognition performance. Although mean-filled coordinates mitigate the information loss to some extent, they remain susceptible to severe video jitter. Therefore, a coordinate smoothing strategy must be introduced to reduce this impact while ensuring information integrity, thereby enhancing the robustness of the extracted features.

3. Motion Recognition and Evaluation Based on Skeletal Pose Points

3.1. Convolutional Neural Networks

(1) Graph Structures

Video and image data are Euclidean data in which pixels are neatly arranged in a matrix, with each pixel having the same number of neighboring pixels. Traditional convolutional neural networks can only process Euclidean data and cannot be used to process non-Euclidean graph data. In mathematical graph theory, graph data refers to topological graphs, which are established by a series of corresponding relationships between nodes and edges. Graph data lacks a regularized structure; each node has a different number of neighboring nodes and lacks translation invariance. Therefore, it is not feasible to use a fixed-size convolution kernel to extract features from nodes in graph data. To effectively process complex graph data, Graph Convolutional Networks (GCNs) can be utilized. Essentially, GCNs update node features based on the connection relationships between nodes, thereby serving as feature extractors.

A graph consists of a finite set of non-empty nodes and a set of edges representing the relationships

between nodes. Graph data is denoted by $G(V,E)$, where $V=\{v_i|i=1,\cdots,n\}$ represents a set of n nodes and $E=\{e_{ij}|v_i,v_j\in V\}$ represents a set of edges connecting the nodes. The adjacency relationships among the nodes in the graph are represented by the adjacency matrix A, defined as shown in Equation (4):

$$A_{ij}=\begin{cases}0, & e_{ij}\in E\\ 1, & otherwise\end{cases} \tag{4}$$

If there is an edge connecting nodes v_i and v_j, the entry A_{ij} in the adjacency matrix A is 1; otherwise, it is 0.

The number of edges connected to each node in the graph is defined as the node's degree d_{v_i}. The degree matrix D represents the degree of each node in the graph. D is a diagonal matrix whose diagonal elements represent the degrees of the individual nodes, as defined in Equation (5):

$$D_{ij}=\begin{cases}d_{v_i}, & i=j\\ 0, & i\neq j\end{cases} \tag{5}$$

(2) Graph Convolutional Networks

There are two types of convolutional methods in GCNs: spectral-domain-based graph convolutions and spatial-domain-based graph convolutions. This section provides a detailed introduction to the latter method.

Spatial graph convolution is used to update node features by aggregating the features of neighboring nodes. In CNNs, aggregating features from a specific region in the previous layer yields the features of neurons in the subsequent layer. By applying this concept of local connectivity to GCNs, the iterative formula for nodes can be derived intuitively, as shown in Equation (6):

$$H^{(l+1)}=AH^{(l)}W^{(l)} \tag{6}$$

In Equation (6), $H^{(l)}$ represents the feature map of the l th layer, A represents the adjacency matrix of the graph, and $W^{(l)}$ represents the weight parameters in the l th layer network. This calculation method has one drawback: it does not take into account the characteristics of the node itself; therefore, it can be replaced with $\tilde{A}$, $\tilde{A}=A+I$, I is a diagonal matrix. This method involves summing the features of neighboring nodes with those of the current node, and the result is proportional to the number of neighboring nodes; the eigenvalue increases as the number of neighboring nodes increases. To address this issue, we normalize the rows of $\tilde{A}$ —that is, the degree matrix of $D^{-1}\tilde{A}$ and D —without altering the feature distribution; we also normalize the columns of $\tilde{A}$ —that is $\tilde{A}D^{-1}$. In this case, Equation (6) can be rewritten as Equation (7):

$$H^{(l+1)}=D^{-\frac{1}{2}}\tilde{A}D^{-\frac{1}{2}}H^{(l)}W^{(l)} \tag{7}$$

3.2. An Action Recognition Algorithm Based on the ST-GCN Attention Mechanism

The specific implementation steps of ST-GCN are divided into the following parts. First, let $G=(V,E)$ denote a spatio-temporal graph of a skeleton sequence with N nodes and T frames, where the set of nodes is denoted by $V=\{v_{ti}|t=1,\cdots,T,i=1,\cdots,N\}$. The feature vector $F(v_{ti})$ of the i th node in the t th frame consists of the node's coordinate vector and the estimated confidence score. The graph structure consists of two parts: first, based on the human body's anatomy, nodes in each frame are connected by edges, forming spatial edges; second, identical nodes in consecutive frames are connected by edges, forming temporal edges. Taking the common two-dimensional convolution of an image as an example, the convolution output for a specific location x can be expressed as follows:

$$f_{ut}(x)=\sum_{h=1}^{K}\sum_{w=1}^{K}f_{in}\left(p(x,h,w)\right)\cdot w(h,w) \tag{8}$$

A feature map f_{in} with c input channels, a convolution kernel of size $K\times K$, a sampling function $p(x,h,w)$, and a weight function with c channels. In the figure, the sampling function

$p(h,w)$ refers to the neighboring pixels centered at pixel x. In the figure, the set of neighboring pixels is defined as:

$$B\left(v_{ti}\right)=\left\{v_{tj}\left|d\left(v_{tj},v_{ti}\right)\le D\right.\right\} \tag{9}$$

Here, $d\left(v_{tj},v_{ti}\right)$ refers to the shortest distance from v_{tj} to v_{ti}, so the sampling function can be written as $p\left(v_{ti},v_{tj}\right)=v_{tj}$. In 2D convolution, neighboring pixels are arranged in a regular pattern around the central pixel, so they can be convolved using a regular convolution kernel based on spatial order. By analogy with 2D convolution, the neighboring pixels obtained from the sampling function in the graph are divided into different subsets, each of which has a numerical label. Thus, $l_{ti}:B\left(v_{ti}\right)\to\{0,\cdots,K-1\}$ maps a neighboring node to the corresponding subset label, and the weight formula is:

$$w\left(v_{tj},v_{ti}\right)=w'\left(l_{ti}\left(v_{tj}\right)\right) \tag{10}$$

Based on the above derivation, the expression for spatial convolution is:

$$f_{out}(v_{ti})=\sum_{v_{tj}\in B(v_{ti})}\frac{1}{Z_{ti}(v_{tj})}f_{in}(p(v_{ti},v_{tj}))\cdot w(v_{ti},v_{tj}) \tag{11}$$

The normalization term $Z_{ti}\left(v_{tj}\right)=\left|\left\{v_{tk}\left|l_{ti}\left(v_{tk}\right)=l_{ti}\left(v_{tj}\right)\right.\right\}\right|$ is equivalent to the basis of the corresponding subset. Substituting the above formula yields:

$$f_{out}(v_{ti})=\sum_{v_{tj}\in B(v_{ti})}\frac{1}{Z_{ti}(v_{tj})}f_{in}(v_{tj})\cdot w(l_{ti}(v_{tj})) \tag{12}$$

After spatial convolution, keypoints must be partitioned into subsets to design the convolutional kernels.

GCN learns local features of adjacent joints in space. Building on this, the behavior recognition network model also needs to learn local features of joint changes over time. How to overlay temporal features onto graph structures is one of the challenges facing graph networks. Research in this area primarily follows two approaches: Temporal Convolutional Networks (TCN) and sequence models (LSTM). ST-GCN employs TCN. Since the shape is fixed, we can use traditional convolutional layers to perform temporal convolution operations. For ease of understanding, this can be analogized to convolutional operations in images. The shape of the last three dimensions of the ST-GCN feature map is (C,V,T), which corresponds to the shape (C,W,H) of the image feature map. Specifically, the number of image channels C corresponds to the number of joint features C. The image width W corresponds to the number of keyframes V. The image height H corresponds to the number of joints T. In spatial convolution, if the kernel size is $w\times1$, then each operation processes w rows and 1 column of pixels. If the stride is s, the kernel moves s pixels at a time, and after processing one row, it proceeds to the next row. In contrast, in temporal convolution, where the kernel size is $tsize$, each operation processes one node across $tsize$ key frames. If the stride is 1, the process moves one frame at a time, performing convolution on one node before proceeding to the next.

Finally, an attention mechanism is incorporated. During movement, different body parts carry varying levels of importance. For example, movements of the legs and arms may be more significant than those of the neck; we can even distinguish between running, walking, and jumping based on leg movements, whereas neck movements often contain less useful information. Therefore, ST-GCN assigns weights to different body parts. Each ST-GCN unit also has its own weight parameters for training.

The overall workflow of ST-GCN is shown in Figure 2. The input data first undergoes batch normalization, then passes through nine ST-GCN units, followed by a pooling layer to obtain a 256-dimensional feature vector for each sequence. Finally, the SoftMax function is used for classification to obtain the final label. When training the sample dataset, two strategies were employed to replace the dropout layer: (1) Random moving: Random affine transformations—fixed angles, translations, and scaling factors—were applied to the skeleton sequences of all frames. (2) Randomly sampling segments of the original skeleton sequences during training and using all frames during testing to prevent the network from overfitting. Since human movements in video data typically consist

of multiple segments, and individual action categories cannot be completely independent of one another, subcategories are classified under parent action categories during classification. Each segment of video data yields different detection results at each consecutive keyframe, which are independent of one another. The test result for the entire video is determined by a vote of the results from each stage, and the action category with the highest score is ultimately output as the overall category for the video. The specific calculation formula for the SoftMax function used to handle this multi-classification problem is as follows:

$$P(i) = \frac{\exp\left(\theta_i^T x\right)}{\sum_{k=1}^{K} \exp\left(\theta_k^T x\right)} \tag{13}$$

Here, θ_i and x are column vectors, and $\theta_i^T x$ can be expressed as the function $f_i(x)$ with respect to x. The SoftMax function maps the probability of the score $P(i)$ to a range between $[0,1]$. In behavioral recognition classification problems, θ represents the parameter to be estimated, and the optimal parameter is determined by finding the value of θ_i that maximizes $P(i)$. Each ST-GCN unit adopts a "one GCN and three TCNs" structure. The first three layers have 64 channels, the middle three layers have 128 channels, and the last three layers have 256 channels. After each pass through the ST-GCN structure, features are randomly dropped out with a probability of 0.5, and the stride of the 4th and 7th temporal convolutional layers is set to 2. Training is performed using SGD with a learning rate of 0.01, which is reduced by 0.1 every 10 epochs. The output channels and structural types of each layer in the ST-GCN are shown in the table below.

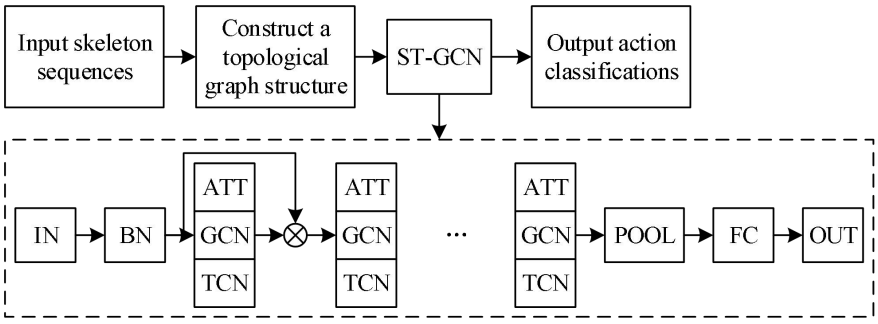


Figure 2. The structure of the GCN topology network

3.3. Sports Movement Analysis Based on the DTW Algorithm

3.3.1. Action Evaluation Metrics

To address the issues of inconsistent and difficult-to-quantify evaluation metrics in traditional motion assessment, the proposed evaluation metrics should be as objective, quantifiable, and unambiguous as possible. In competitive sports such as diving and gymnastics, each judge has a mental template of a standard movement. After an athlete performs a movement, the judge assigns a score based on the degree of similarity between the athlete's performance and the template. Drawing inspiration from this method of movement evaluation in sports, this paper proposes a movement evaluation metric based on motion similarity.

According to the human motion model presented in Chapter 4, each frame of human keypoint data captured by the Kinect is ultimately converted into the relative pose of each joint, expressed in Euler angles. Since this paper primarily focuses on the movement of the human limbs, the selected joint points are the left shoulder joint, left elbow joint, right shoulder joint, right elbow joint, left hip joint, left knee joint, right hip joint, and right knee joint—a total of eight joint points. Consequently, the data structure for each frame of motion data is a 16-dimensional vector:

$$G\{[LS],[LE],[RS],[RE],[LH],[LK],[RH],[RK]\} \tag{14}$$

Each element in vector G is a triple $[\alpha,\beta,\gamma]$, representing the pose of a specific joint relative to its parent joint in the current frame of data.

If each frame of motion data is viewed as a point in a 16-dimensional space, then a motion sequence

can be viewed as a curve in that space. The similarity between two motion sequences can be determined by calculating the distance between the two sets of data; this distance is typically referred to as the Euclidean distance. Suppose $M_1 = (G_0, G_1, G_2, \cdots G_{N-1})$ and $M_2 = (G'_0, G'_1, G'_2, \cdots G'_{N-1})$ are two motion sequences of length N; the Euclidean distance between them is:

$$D(M_1, M_2) = \sqrt{\sum_{i=0}^{N-1} (G_i - G'_i)^2} \tag{15}$$

The smaller the calculated result, the closer the two sets of movements are.

While calculating the Euclidean distance is a simple method for determining the similarity between two motion sequences, it has a very obvious drawback. Since motion sequences are frame-based time series, the number of frames in two sets of motion capture data to be compared is typically unequal. For two sets of data of different lengths, the Euclidean distance cannot be calculated directly; instead, data must be manually added or removed to align the two sets. However, determining which frames to add or remove is a challenging problem when dealing with motion data. Even after manually aligning the two datasets, directly calculating the Euclidean distance to evaluate the similarity between two motions remains inaccurate. This is because, whether a single person performs the same type of motion multiple times or multiple people perform the same type of motion, the speed and starting posture vary each time. Consequently, two sets of motions that appear visually similar may yield a large Euclidean distance value.

3.3.2. DTW Algorithm

To address the limitations of Euclidean distance in evaluating motion similarity, the Dynamic Time Warping (DTW) algorithm is introduced to calculate motion similarity. DTW is widely used in the field of speech recognition. For two audio clips with similar content, their lengths may differ; even if the total lengths of the two clips are the same, the duration of each syllable may vary. Motion data shares similar characteristics with audio data, and DTW can perform time-domain warping to align the two sets of data.

The DTW distance is calculated as follows: Given two motion data sets, $M_1 = (G_0, G_1, G_2, \cdots G_{J-1})$ and $M_2 = (G'_0, G'_1, G'_2, \cdots G'_{K-1})$, first construct a matrix $d(J \times K)$ of size $J \times K$, the element $d(j,k)$ in d represents the Euclidean distance between the points G_j and G'_k, and the DTW-aligned path is shown in Figure 3.

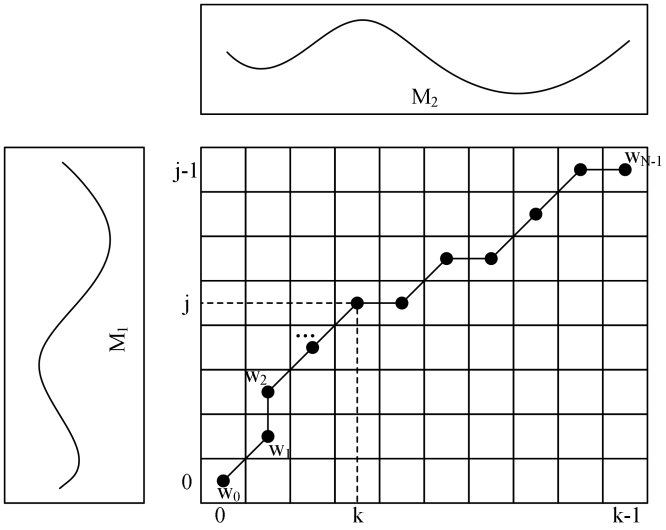


Figure 3. DTW routine path

At this point, the main task of the DTW algorithm is to find a path in the grid shown in Figure 3:

$$W = (w_0, w_1, w_2, \cdots w_{N-1}) \tag{16}$$

Given that $\max(J, K) \le N < J + K - 1$, the following equation holds:

$$DTW\left(M_1, M_2\right) = \min\left\{\sqrt{\sum_{i=0}^{N-1} w_i / N}\right\} \tag{17}$$

To this end, it is necessary to construct a matrix D with the same dimensions as matrix d. The construction method is as follows:

$$\begin{cases} D(0,0) = d(0,0) \\ D(j,0) = d(j,0) + D(j-1,0) \\ D(0,k) = d(0,k) + D(0,k-1) \\ D(j,k) = d(j,k) + \min\{D(j-1,k), D(j,k-1), D(j-1,k-1)\} \end{cases} \tag{18}$$

A path W is called a regular path if it satisfies the following conditions:

(1) Boundary conditions: namely, $w_0 = D(0,0)$ and $w_{N-1} = D(J-1, K-1)$. This is because, although the DTW algorithm can scale time series, it cannot alter the order in which data frames appear; therefore, the aligned path must start at the bottom-left corner and end at the top-right corner;

(2) Continuity: If $w_{i-1} = D(j,k)$, then the next point $w_i = D(j',k')$ on the aligned path must satisfy $(j'-j) \le 1$ and $(k'-k) \le 1$. This ensures that every data frame in both sequences is included in the similarity evaluation without omitting any data;

(3) Monotonicity: If $w_{i-1} = D(j,k)$, then the next point $w_i = D(j',k')$ on the regular path must satisfy both $(j'-j) \ge 0$ and $(k'-k) \ge 0$; that is, once a point on the regular path is determined, its next point can only be the point adjacent to it on the right, above, or to the upper right.

4. Experimental Results and Analysis

4.1. Experiments on Public Datasets

4.1.1. Experimental dataset

This paper conducts experiments using two datasets, NTU-RGBD and Kinetics. All experiments were performed on a Windows 10.0 system with an NVIDIA GeForce GTX 1080 Ti GPU. The PyTorch deep learning framework was used, with PyCharm serving as the integrated development environment.

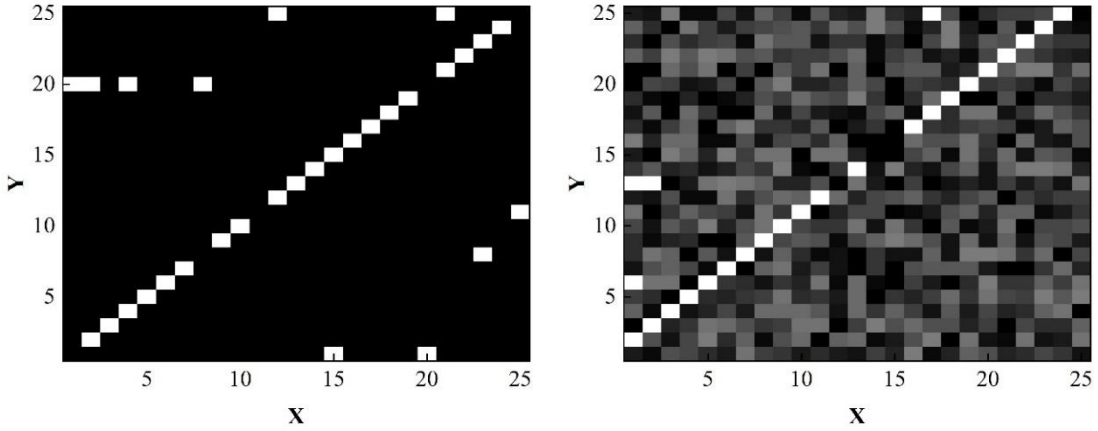
The NTU-RGBD dataset is one of the most widely used and challenging datasets for action recognition tasks. The data in this dataset was collected using three Kinect V2.0 devices, with the three cameras positioned at the same height but at different horizontal angles (−45°, 0°, 45°). The NTU-RGBD dataset comprises a total of 60 distinct action categories, with 56,880 action sample sequences recorded by 40 volunteers aged 10–35 in a controlled laboratory setting. Each sample sequence features no more than two participants and consists of the 3D coordinates of 25 human joints across all frames, encompassing both single-person and two-person everyday actions.

Kinetics is a large-scale human motion dataset comprising 300,000 video clips across 400 action categories. These clips are sourced from YouTube videos and represent a wide variety of activities. This dataset provides only raw video sequences without skeleton data. The open-source OpenPose toolkit was used to estimate the positions of 18 human joint points in each video frame, and the authors' published dataset (Kinetics) was used to evaluate the model described in this paper. The dataset is divided into two parts: a training set (240,000 samples) and a test set (20,000 samples). The model was trained on the training set according to the evaluation methodology, and the Top-1 and Top-5 accuracies on the test set are presented.

4.1.2. Data-Driven Graph Matrix

This paper designs a data-driven graph matrix, A', within the graph attention module. We compare it with the original ST-GCN adjacency matrix to demonstrate the performance of (ST-GCN+A). The adaptive learning of the graph topology benefits action recognition. Figure 4 illustrates an example of the data-driven graph matrix, showing the initial and trained states of its second subset. The matrix shown in Figure 4a is initialized based on the natural physical connections of the human body, while the matrix shown in Figure 4b is the adjacency matrix dynamically adjusted during training; the grayscale of each element in the matrix represents connection strength. From this, it can be seen that

after training, the data-driven graph matrix alters the connections of the original adjacency matrix, making it more suitable for representing the topological structure of the human body in action recognition tasks.



(a) Initialization (b) After training

Figure 4. Examples of the data driver diagram matrix

4.1.3. Comparison of Results and Experimental Analysis

The batch size was set to 64 for all experiments. The optimization strategy employed stochastic gradient descent with a momentum of 0.9, using cross-entropy as the loss function for backpropagation, with a weight decay of 0.0001. For the NTU-RGBD dataset, the maximum number of frames per sample was set to 300. The initial learning rate was set to 0.1, reduced by 10% at the 30th and 40th training iterations, with the training process concluding at the 50th iteration. For the Kinetics dataset, 150 frames were randomly selected from the input skeleton sequence; the initial learning rate was also set to 0.1, reduced by 10% at the 45th and 55th training iterations, with training concluding at the 65th iteration.

On the NTU-RGBD and Kinetics datasets, the final model was compared with the latest skeleton-based action recognition methods; the comparison results are shown in Tables 1 and 2, respectively. The methods included in the comparison are those based on hand-crafted features, RNNs, CNNs, and GCNs. Our model achieved higher accuracy on both datasets, with CS and CV values of 88.5% and 94.9% on the NTU-RGBD dataset, and Top-1 and Top-5 values of 34.8% and 57.8% on the Kinetics dataset, demonstrating superior performance. This effectively validates the superiority of our model.

Table 1. Accuracy in the NTU-RGBD data set %

Method	CS	CV
CNN	50.3	83.3
HBRNN	59.2	63.8
ST-LSTM	69.7	77.2
VA-LSTM	79.5	87.4
SRN+TSL	85.5	92.3
TCN	74.2	82.7
Clips+CNN+MTLN	79.9	85.1
CNN+Motion+Trans	82.5	89.1
3scale ResNet152	85.1	92.1
ST-GCN	81.6	88.4
DPRL+GCNN	83.2	89.8
AS-GCN	87.1	94.1
Ours	88.5	94.9

Table 2. Accuracy in kinetics data sets %

Method	Top-1	Top-5
Feature Enc	15.2	26.1
Deep LSTM	16.2	35.4
TCN	20.4	40.1

ST-GCN	30.7	53.1
AS-GCN	34.3	56.5
Ours	34.8	57.8

4.2. Sports Recognition Results and Analysis

Based on the 50 sports movements collected from OpenPose—including basketball, diving, weightlifting, horizontal bar exercises, and horseback riding—which feature complex backgrounds and widely varying visual perspectives, a total of 6,525 samples were gathered. Of these, 4,000 samples will be selected as the training set, with the remaining samples serving as the test set. The average recognition rate will be used as the metric for evaluating sports movement recognition results, and the ST-LSTM algorithm will be employed for comparative testing.

4.2.1. Analysis of Accuracy in Sports Motion Recognition

Table 3 shows the sports motion recognition results of the proposed method and the comparison methods. As can be seen from the experimental results in Table 3, by leveraging the advantages of spatial aggregation, the proposed method achieves a significantly higher recognition accuracy for sports motions than the comparison methods. The average recognition accuracy of the proposed method on the training and test sets is 90.67% and 88.42%, respectively. Since the random projection algorithm reduces the dimensionality of sports motion features based on maximum contribution, it requires a large number of training samples. Furthermore, it necessitates uniform dimensionality reduction for all training feature samples of sports motions, which can easily disrupt the intrinsic relationships among important features and result in high redundancy of feature information. In contrast, the method proposed in this paper employs a spatio-temporal convolutional module to project features into a low-dimensional subspace, effectively ensuring the reliability of sports motion recognition. At the same time, the experimental results show that for all sports actions, both methods exhibit unsatisfactory recognition results; for example, the recognition accuracy for basketball actions is relatively low. The primary cause of this issue is the complex background of the sports actions. During the movement of the target, the camera is subject to certain interferences, which affect the extraction of sports action features and consequently reduce the recognition accuracy of the sports actions.

Table 3. The accuracy of sports action recognition %

Action type	Training sample		Test sample	
	ST-LSTM	Ours	ST-LSTM	Ours
Play basketball	67.29	70.56	68.13	71.51
Weight lifting	83.89	92.25	84.48	92.73
Golf	87.54	88.84	87.13	88.55
Diving	81.37	88.58	81.48	88.94
Riding	79.74	83.9	80.6	84.81
Horizontal bar	84.73	90.62	83.56	91.03
Mean	87.13	90.67	83.28	88.42

4.2.2. Analysis of the Efficiency of Sports Motion Recognition

On the MATLAB R2014b platform, the recognition efficiency of the two methods for sports motion recognition was tested, with recognition efficiency evaluated based on runtime. The computational time for different feature dimension reduction methods (in seconds) is summarized in Table 4. As shown in Table 4, compared to the baseline method, the proposed method significantly improves the recognition efficiency of sports movements. This is primarily because the baseline method employs the stochastic projection algorithm for feature dimension reduction, which requires matrix factorization, resulting in high time complexity. As the feature dimension increases, the time required for dimension reduction increases dramatically. In contrast, the proposed algorithm utilizes a spatio-temporal convolutional network, which only requires simple matrix operations, thereby greatly improving the efficiency of feature extraction.

Table 4. Feature extraction time of sports action method

Action type	30 Dimension		60 Dimension	
	ST-LSTM	Ours	ST-LSTM	Ours

Play basketball	5.21	4.56	22.21	8.53
Weight lifting	5.26	3.46	24.25	8.75
Golf	6.12	4.74	22.12	9.49
Diving	5.96	3.82	20.15	8.81
Riding	6.63	2.99	23.78	8.99
Horizontal bar	5.07	4.65	23.03	9.97
Mean	6.22	4.32	22.95	9.35

4.3. Assessment and Analysis of Physical Education Movements

4.3.1. Experimental Setup

For motion evaluation, the experiments in this section are divided into two steps: First, the DTW algorithm is used to align the training motion skeleton sequence with the standard motion skeleton sequence, yielding quantified motion results, namely joint scores; then, based on the comparison results of the sports training motions, key frames are selected to provide personalized feedback and guidance on how to improve the sports training motions. The standard motion videos were filmed under the guidance of professional instructors and sports specialists, while the training motion videos were recorded by participants after studying the standard motion videos. The experiments in this chapter were conducted on a system equipped with an Intel® Xeon® CPU E5-2609 v2 @ 2.50 GHz, 16 GB of RAM, an NVIDIA GeForce GTX 1080 Ti graphics card, and running Windows 10. Python was used to implement personalized motion evaluation, while Blender and Unity 3D were used to visualize the guided motions.

4.3.2. Experimental Results

This chapter uses a motion evaluation method to compare training motion data with standard motion data, assess the quality of the training motions, and provide personalized guidance that can effectively improve the accuracy of the training motions. This section validates the personalized motion evaluation method proposed in this paper using the standing hand-clasping swing motion as an example. Since all joints move relative to the root joint, the root joint is not scored in the table, and scores are rounded to integers. The personalized motion evaluation method in this paper provides feedback on the primary areas of the body where the training motion deviates from the standard, offering personalized feedback recommendations (guided motions) to improve the standardization of the motion. To validate the effectiveness of the method, this experiment compared motion data from university students after receiving feedback with data collected before feedback. The results of the improvement in motion quality following personalized evaluation feedback are shown in Figure 5. It can be observed that after receiving feedback, the motion quality of university students improved significantly, with overall score fluctuations becoming more stable, and scores generally tending toward the 90–100 range. The experimental results indicate that the feedback provided by the motion evaluation method described in this paper effectively helps users improve motion quality and enhance compliance with standard motion patterns.

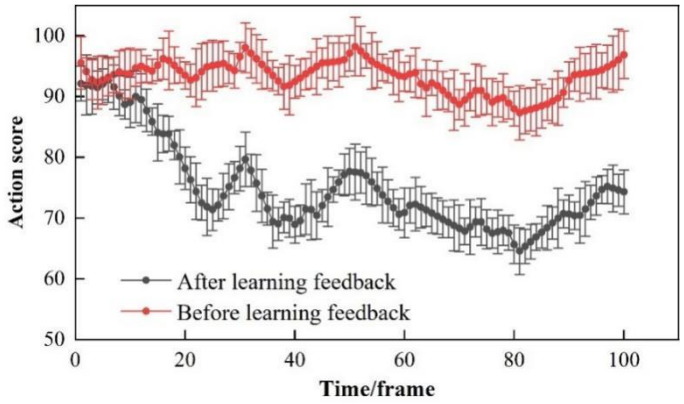


Figure 5. Improved action quality of personalized feedback feedback

4.3.3. Comparative Experimental Analysis

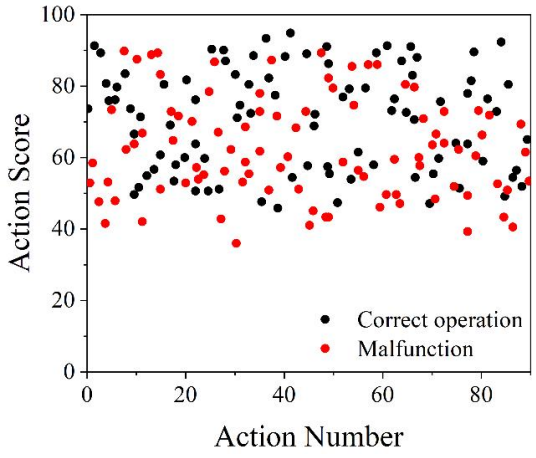
(1) Assessment of Scoring Accuracy

Quantitative motion evaluation serves as the foundation for feedback tasks; the accuracy of joint scoring influences both the selection of personalized labels and the calculation of motion feedback. In this section, we use discriminant power as a performance metric to evaluate the scoring accuracy of our motion evaluation method by comparing it with three classic motion quantification methods—Euclidean distance, cosine similarity, and traditional DTW—on a self-built sports dataset.

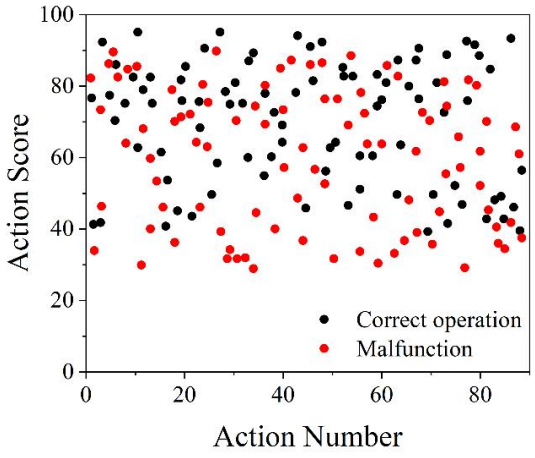
Ten individual students repeated each movement 10 times in front of two motion capture systems: a Vicon optical tracker and a Kinect camera. The dataset provides position and angle data for body joints in the skeletal models of both the Vicon and Kinect systems, with 39 joint points in the Vicon data and 22 joint points in the Kinect data. Data for movements performed both correctly and incorrectly are provided; the correctly performed movements represent training movements by healthy individuals, while the incorrectly performed movements simulate training movements by students with simulated musculoskeletal injuries or physical constraints.

To ensure fairness in the experimental comparison, the numerical values of the measurement results in this section are scaled to the range [0, 100]. In this section, a set of correct movements performed by Participant 6 is selected as the standard movement data, while the data from other participants' correctly and incorrectly repeated movements serve as the training movement data.

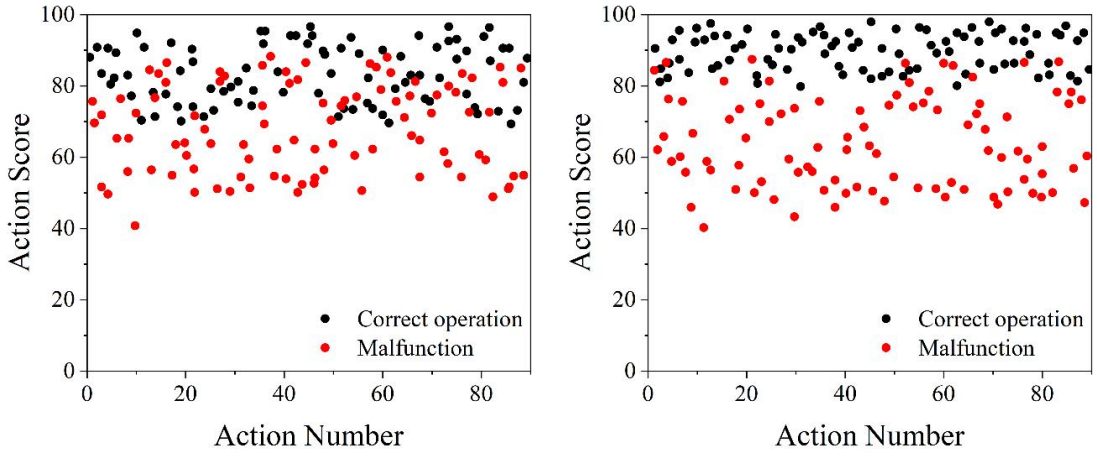
The scoring results for action E8 are shown in Figure 6, where black circles indicate correct actions and red circles indicate incorrect actions. As shown in the figure, a higher discrimination index indicates a stronger ability to distinguish between correct and incorrect actions, while a lower index indicates a weaker ability. The experimental results demonstrate that the action scoring method proposed in this paper is more effective at distinguishing between correct and incorrect actions and achieves higher scoring accuracy. The results show that under the Vicon system, the average separation values for Euclidean distance, cosine similarity, the traditional method, and DTW were 0.075, 0.052, 0.133, and 0.212, respectively. Under the Kinect system, the average separation values for Euclidean distance, cosine similarity, the traditional method, and DTW were 0.055, 0.044, 0.095, and 0.122, respectively. Overall, the separation of data collected by the Kinect device was smaller than that of the Vicon device. This is because, when motion occlusion occurs, the data collected by the Kinect device exhibits instability in certain joint measurements, leading to inaccurate motion quantification. This further underscores the importance of accurate human skeleton data for motion evaluation.



(a)Euclidean distance S_D =0.084



(b)Cosine similarity S_D =0.064



(c) Traditional S_D =0.154 (d) Methodology of This Article S_D =0.218

Figure 6. Action E8 Evaluation Results

(2) Evaluation of Movement Improvement Effects

This experiment involved 20 participants, none of whom had previously learned the sports movements described above. Since the experiment aimed to verify the effectiveness of different types of feedback in correcting non-standard movements, and to eliminate the influence of other factors, all participants were in good physical health.

First, motion data was collected from the participants' initial training sessions using videos of the standard movements. Next, the 20 participants were randomly divided into four equal groups, with each group assigned a specific type of feedback. Participants adjusted their movements based on the feedback, and motion data reflecting these improvements was collected once daily. Finally, the improved motion data was compared with the standard motion data to obtain pre- and post-improvement motion scores. The results of movement improvement across different feedback types are shown in Table 5. The personalized sports feedback described in this study demonstrated superior movement improvement capabilities compared to the other three feedback types, with noticeable improvements observed as early as the first day. Among the other three feedback types, score-based feedback yielded the poorest results in terms of movement improvement, while semantic feedback and traditional sports feedback effectively improved movement quality, albeit to a lesser extent.

Table 5. Changes in different feedback forms

	Graded feedback	**Semantic feedback**	**Traditional motion feedback**	**Ours**
Day 0	74	69	71	69
Day 1	72	76	80	85
Day 2	74	78	83	88
Day 3	76	82	85	90
Day 4	72	82	86	91
Day 5	77	82	86	91

5. Conclusion

Smart technologies have revitalized the physical education system in higher education, demonstrating multifaceted value and application potential. Research indicates that by leveraging key technologies such as sports posture estimation, motion recognition, and motion evaluation, it is possible to effectively achieve personalized training guidance, scientific teaching management, and precise performance assessment. Specifically, the method proposed in this paper achieved average sports motion recognition accuracy rates of 90.67% and 88.42% on the training and testing sets, respectively. Furthermore, the feedback provided by the motion evaluation method can effectively help users improve motion quality and meet performance standards, thereby comprehensively enhancing the quality and efficiency of physical education and training. However, practical challenges remain in areas such as data security, technological adaptation, and the transition of teaching staff. In the future, a multi-dimensional, coordinated approach—including strengthening platform development, improving teacher training, establishing robust data security mechanisms, and creating a dynamic evaluation

system—is needed to build a modern university physical education system supported by AI technology, thereby ensuring the cultivation of innovative talent with sound physical and mental health and well-rounded development.

References

1. Zhong, Q., Jiang, J., Bai, W., Yin, Z., Liao, Z., & Zhong, X. (2025). Application of digital-intelligent technologies in physical education: a systematic review. Frontiers in public health, 13, 1626603.
2. Al-Attabi, K., & Raju, V. V. R. (2025). Enhancing Sports Performance through the Integration of Smart Equipment and Sensors. Advances in Sports Science and Technology, 127-131.
3. Srivastava, P. K., Pandey, R. K., Srivastava, G. K., Anand, N., Krishna, K. R., Singhal, P., & Sharma, A. (2024). Intelligent Integration of Wearable Sensors and Artificial Intelligence for Real-time Athletic Performance Enhancement. Journal of Intelligent Systems & Internet of Things, 13(2).
4. Sun, J. (2026). Personalized sports training recommendation system based on motion sensors and data mining. International Journal of System Assurance Engineering and Management, 1-9.
5. Babu, A., Thuau, D., & Mandal, D. (2023). AI-enabled wearable sensor for real-time monitored personalized training of sportsperson. MRS Communications, 13(6), 1071-1075.
6. Chen, Y., & Tian, X. (2024, December). Generation Algorithm of Personalized Sports Training Program Based on Sensor Data. In 2024 IEEE 2nd International Conference on Electrical, Automation and Computer Engineering (ICEACE) (pp. 1291-1296). IEEE.
7. MD, D. A. J., & KOLLI, D. E. T. (2025). The future of sports training: Integrating artificial intelligence and wearable technology in performance enhancement. TPM–Testing, Psychometrics, Methodology in Applied Psychology, 32(S2 (2025): Posted 09 June), 2145-2153.
8. Zhao, J., Yang, Y., Bo, L., Qi, J., & Zhu, Y. (2024). Research progress on applying intelligent sensors in sports science. Sensors, 24(22), 7338.
9. Tian, T. (2025). Wearable sensor-based real time monitoring system for physical education teaching and training. Molecular & Cellular Biomechanics, 22(1).
10. Shi, X., & Zou, H. (2024). Data collection and analysis based on sensor technology in sports training. Scalable Computing: Practice and Experience, 25(5), 4399-4406.
11. Zhang, K., & Liu, S. J. (2016, July). The application of virtual reality technology in physical education teaching and training. In 2016 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI) (pp. 245-248). IEEE.
12. Hamizi, M. A. A. B. M., Mokmin, N., & Ariffin, U. (2022). Virtual reality technology in physical education: A systematic review in instructional design & implementation. Advanced Journal of Technical and Vocational Education, 6(1), 6-12.
13. Ding, Y., Li, Y., & Cheng, L. (2020). Application of Internet of Things and virtual reality technology in college physical education. Ieee Access, 8, 96065-96074.
14. Meng, J. (2021). College physical education teaching aided by virtual reality technology. Mobile Information Systems, 2021(1), 3052895.
15. Feng, Y., You, C., Li, Y., Zhang, Y., & Wang, Q. (2022). Integration of computer virtual reality technology to college physical education. Journal of web engineering, 21(7), 2049-2071.
16. Yang, Y. (2018). The innovation of college physical training based on computer virtual reality technology. Journal of Discrete Mathematical Sciences and Cryptography, 21(6), 1275-1280.
17. Zhao, Z. (2024, August). Research on immersive experience system of virtual reality technology in college physical education teaching. In 2024 International Conference on Computers, Information Processing and Advanced Education (CIPAE) (pp. 833-837). IEEE.

18. Bores-García, D., Cano-de-la-Cuerda, R., Espada, M., Romero-Parra, N., Fernández-Vázquez, D., Delfa-De-La-Morena, J. M., ... & Palacios-Ceña, D. (2024). Educational research on the use of virtual reality combined with a practice teaching style in physical education: A qualitative study from the perspective of researchers. Education Sciences, 14(3), 291.
19. Pan, Y. (2024). Sports game teaching and high precision sports training system based on virtual reality technology. Entertainment Computing, 50, 100662.
20. Turdaliyev, R., Botagariyev, T., Ryskaliyev, S., Doshybekov, A., & Kissebaev, Z. (2024). Virtual reality technology as a factor to improve university sports. Retos, 51, 872-880.
21. Liu, G. (2022). Physical education resource information management system based on big data artificial intelligence. Mobile information systems, 2022(1), 3719870.
22. Wang, Y., & Yu, L. (2022). Multisource Analysis of Big Data Technology: Accessing Data Sources for Teacher Management of Sports Training Institutions. Mobile Information Systems, 2022(1), 5115184.
23. Zhang, L., Wang, F., & Qi, A. (2017). Construction of interactive teaching system for exercise training based on education video resource push technology. International Journal of Emerging Technologies in Learning (Online), 12(7), 158.
24. Wang, C., & Wang, D. (2023). Managing the integration of teaching resources for college physical education using intelligent edge-cloud computing. Journal of Cloud Computing, 12(1), 82.
25. Liang, X., & Yin, J. (2022). Recommendation algorithm for equilibrium of teaching resources in physical education network based on trust relationship. Journal of Internet Technology, 23(1), 133-141.
26. Wu, J. H. (2025). Design and development of artificial intelligence dynamic physical education teaching resources in human-computer interaction mode. Journal of Educational Computing Research, 63(5), 1219-1248.
27. Guo, H., & Cheng, X. (2022). Individual recommendation method of college physical education resources based on cognitive diagnosis model. EAI Endorsed Transactions on Scalable Information Systems, 9(5).

About the Author

Lizhang Cheng (born February 1983), male, Han ethnicity, native of Wenshang, Shandong Province. Lecturer at Jining Normal University, Doctoral Candidate. Research interests: Physical Education Teaching and Training, Exercise Prescription.