Hindawi
Scientific Programming
Volume 2022, Article ID 8393612, 12 pages
https://doi.org/10.1155/2022/8393612

Research Article

Construction of Sports Training Management Information System Using AI Action Recognition

Dali Cheng,[1] Hong Wang,[2] and Min Li[1]

[1]*College of Physical Education, Chongqing University of Arts and Science, Chongqing 412160, China*
[2]*Xuanhua Middle School Yongchuan, Chongqing 402160, China*

Correspondence should be addressed to Min Li; 19980006@cqwu.edu.cn

Received 6 May 2022; Revised 4 June 2022; Accepted 10 June 2022; Published 24 June 2022

Academic Editor: Mian Ahmad Jan

With the development of science and technology, more and more fields have begun to use AI to provide convenient services for humans. Artificial intelligence (AI) refers to a new technology that uses human thinking to respond accordingly through computers and robots to assist human beings. Action recognition is an important research project that needs to be broken through in many industries, such as security system, martial arts instruction, and dance training. This paper aims to study a method for action recognition using AI technology and to build a sports training management information system. In this paper, a recognition model and related algorithms using a convolutional neural network (CNN) are proposed, and an intelligent sports training management information system is constructed. The system and the model are tested, the action recognition effect of 60 athletes in a university is tested, and the comparison with the traditional recognition algorithm is carried out. The results show that the CNN recognition accuracy test results used in this paper are generally more than 90%, while the traditional recognition accuracy rate is only about 75%, and the highest is not more than 86%; the training management information system of this paper takes about 15.7 s, and the maximum time is not more than 10 s, while the traditional recognition system takes about 15.7 s, which is about twice the time of the system in this paper. Therefore, it shows that the CNN recognition method in this paper has a significantly better effect on the recognition of athletes' movements, and the sports training management information system constructed in this paper is less time-consuming and faster and has certain feasibility.

1. Introduction

Human action recognition is an important and challenging topic in computer vision. Recently, CNN has established impressive results for many image recognition tasks. CNN typically contains millions of parameters that are prone to overfitting when trained on small datasets. Therefore, CNN does not yield better performance than traditional action recognition methods. Additionally, sparsity has been shown to be one of the most important properties for visual recognition purposes. Adaptive sparse coding is used to capture high-level patterns from the data.

Human motion recognition plays an increasingly important role in many fields such as smart home, human-computer interaction, patient monitoring system, medical, and health care and can be applied to many aspects such as public places, medical care, and security. With the advent of the era of big data, the continuous increase in the amount of data such as various media interactions and chat software has generated a large amount of information about human movements, and the recognition of human movements in videos faces great challenges.

With the boom of low-cost and easy-to-operate depth cameras, neural network-based human action recognition has been extensively studied recently. However, most existing methods partially consider all 3D joints of the human skeleton to be the same. In fact, these 3D joints exhibit different responses to different action categories, and certain joint configurations are more discriminative for distinguishing specific actions.

The traditional CNN structure can maintain a certain degree of translation and rotation invariance for a specific position in space for the input image. This spatial invariance only acts on the local area of the input image, and the entire image cannot achieve the invariance of the overall spatial rotation in the stacked local area. Since the pooling layer in the CNN structure has many limiting factors, for example, when extracting features, a lot of useful information is lost, the input data are only a local operation, and the feature map in the middle of the CNN framework will produce large distortions and will make it difficult for CNN to achieve spatial transformations such as rotation and scaling of images. The feature map generated in the process of CNN extracting features is not an overall transformation of the input data and is more restrictive. At the same time, there are large intraclass differences between the categories of human actions, and different people may perform a class of actions with great differences in amplitude and frequency. And there may be big differences in body size between different people. Human movements cannot be completed in one frame.

With the development of technology, the previous methods can no longer meet the current environment. The innovation of this paper is to propose a three-dimensional CNN recognition algorithm model, which can recognize athletes' movements more quickly and accurately. And the test results of this paper are compared with the traditional identification methods, the data are more intuitive, and the results are more obvious.

2. Related Work

Regarding action recognition technology, many scholars have carried out related research on it. Fanello et al. proposed the use of linear support vector machines for simultaneous online video segmentation and action recognition and showed that sparse representations play an important role in enabling one-shot learning and real-time action recognition. The main contribution of his research was an efficient real-time action modeling and recognition system; the paper highlighted the effectiveness of sparse coding techniques in representing 3D actions [1]. The aim of Nakonechna et al.'s study was to estimate the distribution of phosphatidylserine in the phospholipid bilayer of the liver membrane and the apoptotic stage of rat hepatocytes under the influence of surfactants: ethylene glycol (EG), polyethylene glycol 400 (PEG-400), and polypropylene glycol (PPG). Nakonechna et al. utilized the specific signals of macrophages to specifically recognize and eliminate apoptotic cells [2]. Yu and Yun presented a novel method, which was called the maximum margin heterogeneous information machine (MMHIM), for human action recognition from RGB-D videos. MMHIM fuses heterogeneous RGB visual features and depth features and uses the fused features to learn an efficient action classifier. Rich heterogeneous vision and depth data are efficiently compressed and projected into shared spaces and independent private spaces for learning to reduce noise and capture useful information for identification. Knowledge from various sources can then be shared with others in the learning space to learn cross-modal features [3]. Yanhua et al. put forward a discriminative multiinstance multitask learning (MIMTL) framework to discover intrinsic connections between joint configurations and action classes. Yanhua conducted extensive evaluations on MIMTL using three benchmark 3D action recognition datasets. Experimental results showed that the MIMTL framework proposed by Yanhua had good performance compared with several state-of-the-art methods [4]. However, these action recognition methods are more traditional and can only be used for single human action recognition and are difficult to apply multiple recognition objects and complex environments.

In order to be applicable to the complex environment and the situation of multiple recognition objects, some scholars have proposed new recognition methods. Nguyen et al. proposed a new method for action recognition based on Gaussian descriptors. Experimental evaluations showed that the method achieved very promising results on all datasets [5]. Xiu-Hong et al. used UPLC-MS technology and the pattern recognition method to study the potential biomarkers of endometriosis in rats with cold blood coagulation and blood stasis (ECB) and the effective mechanism of paeoniflorin (PF) [6]. Yu et al. designed a novel two-stream fully convolutional network architecture for action recognition, which can significantly reduce parameters while maintaining performance [7]. Although these methods can be adapted to multitarget recognition and complex environments, the recognition efficiency is not high, and the recognized results are not ideal, so it is necessary to further improve.

3. AI-Based Action Recognition Method

3.1. AI-Based Action Recognition. Artificial intelligence (AI for short) is an intelligent technology that simulates the human brain for thinking operations and has been widely used in many fields. With the influx of AI research, AI-related applications such as smart sweeping robots, smart speakers, personal assistants, and face-scanning payments are gradually appearing in people's lives [8, 9]. AI research is almost ubiquitous in people's lives, and related research is being carried out in various fields such as finance, medical care, autonomous driving, and education, as shown in Figure 1. Heavy scientific and engineering calculations are originally undertaken by the human brain. Today, AI can not only complete this calculation but also do it faster and more accurately than the human brain. Therefore, contemporary people no longer regard this kind of computing as a "complex task that requires human intelligence to complete." It can be seen that the definition of complex work changes with the development of the times and the advancement of technology, and the specific goals of the science of AI also naturally develop with the changes of the times.

AI includes a variety of science and technology, as shown in Figure 2. Machine learning is a core technology of AI, and neural network is the most important algorithm of machine learning. At present, for action recognition, CNN neural network has unique advantages in action recognition due to its convolution principle [10, 11]. Machine learning is a multidomain interdisciplinary subject involving probability

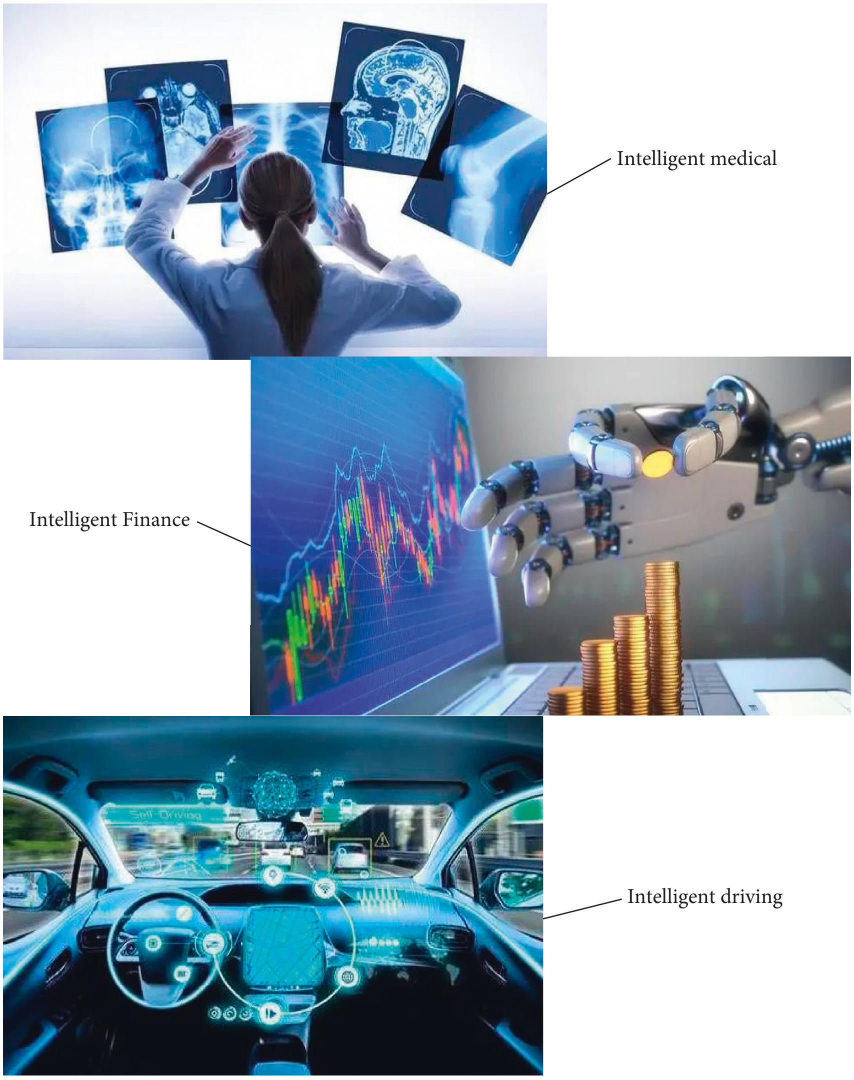
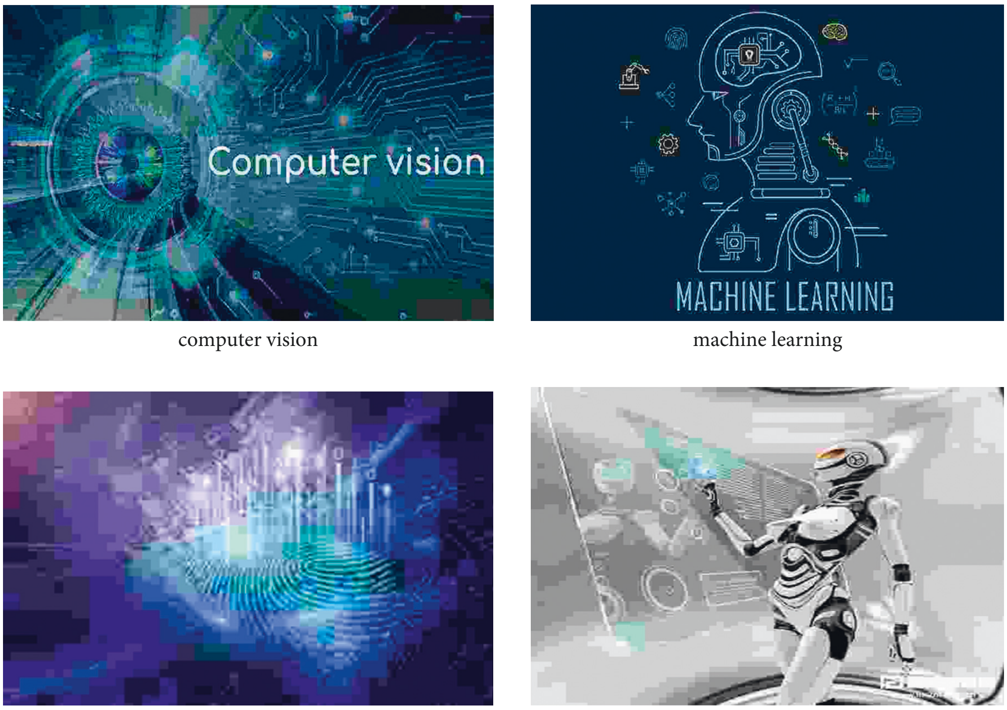


Figure 1: AI application areas.

theory, statistics, approximation theory, convex analysis, algorithm complexity theory, and other disciplines. It specializes in how computers simulate or realize human learning behaviors to acquire new knowledge or skills and reorganize existing knowledge structures to continuously improve their performance.

3.2. Neural Networks. 3D CNN is the most popular method used in human action recognition tasks. Figure 3 shows an example of a 3D CNN, where a typical 3D CNN consists of multiple layers of basic structures stacked. Each basic structure includes a 3D convolutional layer, a batch normalization layer, an activation layer, and a 3D pooling layer. If it is used for human action recognition tasks, the input of the network is a video frame cube composed of video frames intercepted from the original video, and a fully connected layer and a classification layer will be added at the end of the network [12, 13]. The approximate calculation process of 3D CNN is that the video frame cube is input from the input layer; the 3D convolution layer convolves the input feature cube with a 3D convolution kernel to obtain the output feature cube; the batch normalization layer roughly normalizes the pixel values of the output feature cube into a distribution with 0 mean and unit variance; the output feature cube is subjected to the nonlinear transformation of the activation function of the activation layer to obtain the activated feature cube; the three-dimensional pooling layer performs three-dimensional pooling operation on the activated feature cube to reduce the size of the feature cube; the fully connected layer further learns the features; the classification layer obtains the probability of each category [14].

The core part of the 3D CNN is the 3D convolution operation, which is also the most computationally intensive part of the entire network. The convolution process is shown in Figure 4. The input of the 3D convolution operation is the input feature cube and the 3D convolution kernel, and the output is the output feature cube. The input feature cube in the first layer refers to the first input video frame cube. The middle layer refers to the output of the previous layer. Generally speaking, feature cubes are multichannel, which means that in addition to the height, width, and depth



computer vision

machine learning

Biometrics

Robot

Figure 2: AI core technology.

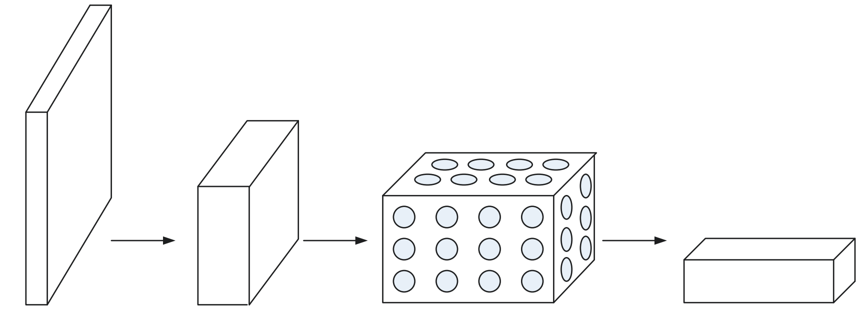


Figure 3: Schematic diagram of CNN.

dimensions of the feature cube itself, there are also channel dimensions. The 3D convolution kernel is a feature extractor used to extract features in the input feature cube, which also has four dimensions. Its height, width, and depth dimensions are smaller than the corresponding dimensions of the input feature cube, but the fourth dimension is equal to the number of channels of the input feature cube. Its parameters, called kernel weights, can be trained to make it a better feature extractor [15, 16]. The output feature cube is the extracted features. The process of extracting features is convolution, and the convolution is similar to the convolution in the field of digital signal processing, but it is not the same [17].

The size of each dimension of the output feature cube of the 3D convolutional neural network will be reduced according to the multiple of the corresponding dimension size of the pooling cube. Usually, the three dimensions of the pooling cube are all 2, so the size of each dimension of the output feature cube after pooling is generally one-half of the input feature cube. The pooling operation is similar to the convolution operation, in which a small cubic filter slides on the input feature cube to obtain the output feature cube. However, the sliding step size during pooling is usually the side length of the pooling cube, and the small cube filter during single pooling and the part of the input feature cube framed by it are not multiplication and accumulation calculations performed. Instead, the average or maximum value is calculated according to the different pooling methods [18].

Grouped convolutions can also reduce computation like depthwise separable convolutions. However, it avoids the shortcomings of the depthwise separable convolution and does not increase the binary activation operation at the algorithm level, nor does it increase the memory access operation at the hardware implementation level. Not only that but it also has a big advantage at the hardware implementation level. Since its convolution is only performed within each group, the addition between channels is also performed only between channels within each group. Therefore, the pixel value range of its output feature map will be reduced by the number of groups, which will greatly reduce the consumption of storage resources. Despite these advantages, grouped convolution also has its disadvantage, which is the problem of poor information flow between groups as explained earlier. However, this shortcoming can be eliminated by using the channel rearrangement method at low cost, so this paper uses grouped convolution for lightweight design [19, 20].

This paper firstly analyzes the most widely used depthwise separable convolution method, which is characterized by decomposing conventional convolution into two steps: channel-by-channel convolution and point-by-point convolution. The advantage is that the number of multiplication calculations of convolution is reduced. Although it works well in real-valued CNN, it is not suitable for the algorithm model of this paper. Because the algorithm model in this paper has a binary activation operation before each convolution calculation, the depthwise separable convolution is used. It divides one layer of convolution into two and changes the binary activation operation into two, and more activation operations will bring about the loss of information. In particular, the value range of the output of the binary channel-by-channel convolution is small because there is no addition between channels. That is to say, the included feature information is relatively weak, and the subsequent insertion of a binary activation will make the feature information weaker. This part is the analysis from the perspective of the algorithm. From the perspective of hardware implementation, the depthwise separable convolution method is not suitable for this study. The purpose of hardware implementation is to use the parallel computing characteristics of hardware to make neural network computing faster, but not infinitely stacked computing units can accelerate infinitely. This is because the computing speed is also limited by the speed of the data supply; that is, it is limited by the storage bandwidth. Therefore, in order to design the operation of a hardware-accelerated neural network, both computing unit design and memory access mode must be taken into account.

The research of this paper not only stays at the algorithm level but also designs its dedicated hardware accelerator for the proposed algorithm. Because it involves the design of the hardware circuit, the design of the algorithm model should not only consider the performance of the algorithm but also consider the consumption of logic resources such as gate units, flip-flops, and on-chip storage when the algorithm is implemented in hardware. The binary CNN model can well meet these requirements, and it not only greatly reduces the storage requirements but also has no multiplication calculation. It is a hardware-friendly CNN model, so the algorithm model in this paper also adopts binarization technology. The difference between the convolution operation in the binarization technique and the conventional convolution operation is that its input has only two values of -1 and 1. In this case, the multiplication calculation in the conventional convolution can be simplified to the XNOR calculation in the logical operation, and the addition calculation can be simplified to the bit count calculation. The result of the binary convolution is not a single bit, but some integers with a small range of values can be represented by a small number of bits, which is why the binary activation operation is used to binarize it again later.

4. CNN Action Recognition Sports Training Management Information System

4.1. CNN Computational Model. The CNN structure is shown in Figure 5. The convolution operation includes continuous and discrete convolution. For continuous convolution, the calculation is

$$y(t) = \int_{-\infty}^{\infty} x(p)h(t-p)dp = x(t) * h(t). \tag{1}$$

The calculation for discrete convolution is

$$y(n) = \sum_{i=-\infty}^{\infty} x(i)h(n-i) = x(n) * h(n). \tag{2}$$

Additionally, 2D convolutional feature extraction is calculated as

$$v^{xy} = \sum_{p=0}^{P-1}\sum_{q=0}^{Q-1} w^{pq} u^{(x+p)(y+q)}. \tag{3}$$

The 3D convolutional feature extraction is calculated as

$$v^{xyz} = \sum_{p=0}^{P-1}\sum_{q=0}^{Q-1}\sum_{r=0}^{Q-1} u^{(x+p)(y+q)(z+r)} w^{pqr}. \tag{4}$$

The sampling of 2D graphics can be expressed as

$$y_{mn} = \frac{1}{S_1 S_2}\sum_{j=0}^{S_2-1}\sum_{i=0}^{S_1-1} x_{m\cdot S_1+i,n\cdot S_2+j}. \tag{5}$$

S is the figure size, y is the output, and x is the input. The maximum sampling of 3D video can be expressed as

$$y_{mnl} = \max_{i,j,k;S_1,S_2,S_3}\left(x_{m\cdot S_1+i,n\cdot t+j,l\cdot r+k}\right). \tag{6}$$

The back-propagation process of the convolutional neural network is shown in Figure 6, in which the loss function is introduced.

$$J(W,b;x,y) = \frac{1}{2}\left\| h_{W,p}(x) - y^2 \right\|. \tag{7}$$

where W, b represents the weight of the CNN.

For a dataset with m samples, the loss function is

$$J(W,b) = \left[\frac{1}{m}\sum_{i=1}^{m} J\left(W,b;x^i,y^i\right)\right]. \tag{8}$$

The way to update the weights using the gradient descent method (gradient descent is a first-order optimization

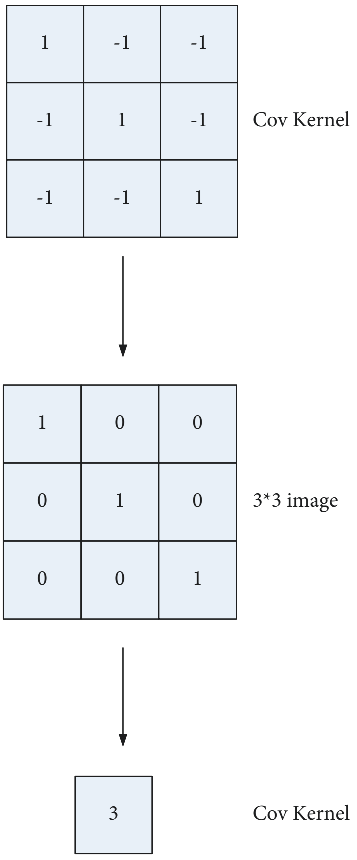


Figure 4: CNN convolution operation.

algorithm, also commonly known as steepest descent. To find the local minimum of a function using gradient descent, an iterative search must be performed to a point at a specified step distance in the opposite direction of the gradient corresponding to the current point on the function.) is

$$
\begin{aligned}
W_{ij}^{l} &= W_{ij}^{l} - \alpha \frac{\partial}{\partial W_{ij}^{l}} J(W, b), \\
b_i^l &= b_i^l - \frac{\partial}{\partial b_i^l} J(W, b).
\end{aligned}
\tag{9}
$$

For the n_l-th layer network, the residuals of each node are

$$
\delta_i^{n_l} = \frac{\partial}{\partial Z_i^{(n_l)}} \frac{1}{2} y - h_{w,b}(x)^2. \tag{10}
$$

The residual for the i-th neuron node of the l-th layer is

$$
\delta_i^l = \left(\sum_{j=1}^{S_{l+1}} W_{ji}^{l} \delta_j^{l+1} \right) f'\left(z_i^l\right), \tag{11}
$$

where z_i^l is the weight increment.

From this, the partial derivatives of the loss function and the nodes and biases of each layer can be obtained.

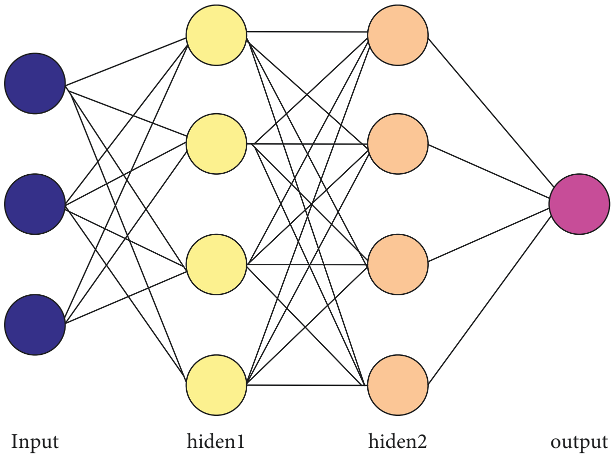


Figure 5: CNN network structure.

$$
\begin{aligned}
\frac{\partial}{\partial W_{ij}^{l}} J(W, b; x, y) &= a_j^l \delta_i^{l+1}, \\
\frac{\partial}{\partial b_i^l} J(W, b; x, y) &= \delta_i^{l+1}.
\end{aligned}
\tag{12}
$$

After the identification and fusion of the graphics, the regression analysis is carried out. In this paper, the softmax regression model is selected, and its expression is

$$
h_\theta\left(x^i\right) = g\left(\theta^T x^i\right) = \frac{1}{1 + e^{-\theta^T x^i}}. \tag{13}
$$

(x, y) represents the sample training set.

With the training parameter θ, the loss function is calculated

$$
J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y_i \log h_\theta\left(x^i\right) + \left(1 - y^i\right)\log\left(1 - h_\theta\left(x^i\right)\right)\right]. \tag{14}
$$

Let h_θ be

$$
h_\theta\left(x^i\right) = \frac{1}{\sum_{j=1}^{k} e^{\theta^T x^i}}. \tag{15}
$$

Then, the training model is iteratively calculated, and the gradient can be obtained by derivation

$$
\nabla_{\theta_j} J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[\left(x^i 1\left\{y^i = j\right\} - p\left(y^i = j | x^i; \theta\right)\right)\right]. \tag{16}
$$

$\nabla_{\theta_j} J(\theta)$ represents the partial derivative value.

The update method of the parameters is

$$
\theta_j = \theta_j - \alpha \nabla_{\theta_j} J(\theta). \tag{17}
$$

The loss function is updated to

$$
J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} \sum_{j=1}^{k} 1\left\{y^i = j\right\} \log \frac{e^{\theta^T x^i}}{\sum_{j=1}^{k} e^{\theta^T x^i}}\right] + \frac{\lambda}{2} \sum_{i=1}^{m} \sum_{j=1}^{k} \theta_{ij}^2. \tag{18}
$$

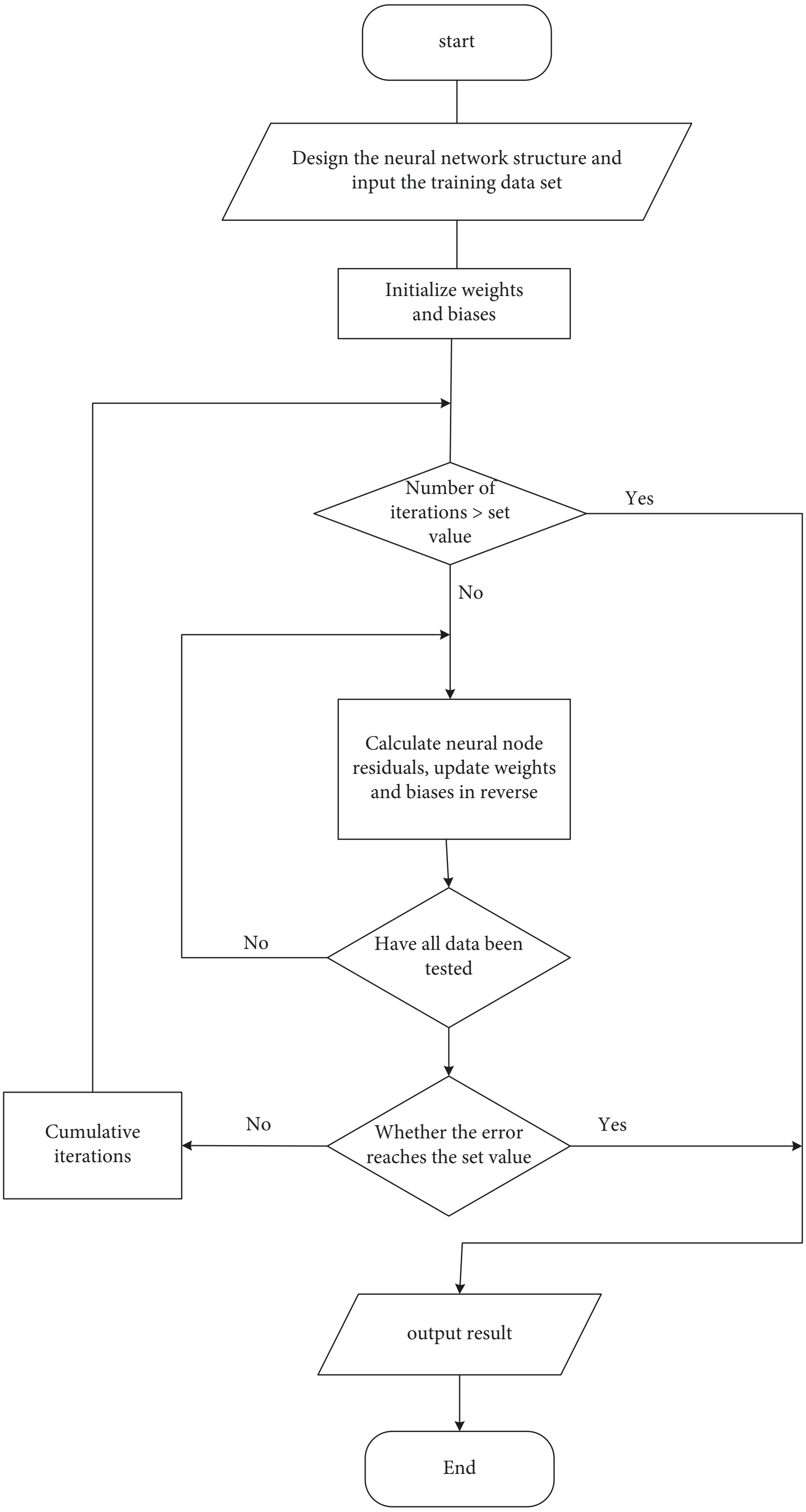

start
Design the neural network structure and input the training data set
Initialize weights and biases
Number of iterations > set value
Yes
No
Calculate neural node residuals, update weights and biases in reverse
No
Have all data been tested
Cumulative iterations
No
Whether the error reaches the set value
Yes
output result
End

Figure 6: CNN back-propagation process.

After filtering the term, it is transformed into a convex function of the optimal solution, and the derivative of it can finally be obtained

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[x^i\left(1\left\{y^i = j\right\} - p\left(y^i = j|x^i;\theta\right)\right)\right]. \quad (19)$$

Finally, $J(\theta)$ is minimized, and the softmax regression model (the softmax logistic regression model is a generalization of the logistic regression model for multiclassification problems. In a multiclassification problem, the class label y can take more than two values. Softmax regression models are useful for problems such as MNIST handwritten digit classification, where the goal is to identify 10 different single digits) is obtained, and then, the classification and recognition of actions are realized.

4.2. Experimental Design

4.2.1. Management System. The sports training management system designed in this paper is shown in Figure 7, and its objects mainly include athlete, referee, and planner.

In this paper, the research at the level of the 3D CNN algorithm is carried out from the perspective of reducing the data bit width of the network and reducing the amount of network computation. Binarization and lightweight techniques are applied to optimize the design of 3D CNN. In order to pave the way for the next hardware accelerator design at the algorithm level at the cost of a slight loss of recognition accuracy, a hardware accelerator with less resource consumption, lower power consumption, and faster speed can be designed. The research content of this paper at the level of 3D CNN hardware accelerator design is to design its dedicated hardware accelerator for the algorithm model of this paper. On the premise of correct function, it can achieve a high processing frame rate and high computing energy efficiency, in order to use it to build an action recognition system with practical use value.

4.2.2. Training Method. Untrained CNN cannot be used directly, because the initialization parameters of CNN do not have the ability to extract features from the input, so its output has no value. The back-propagation algorithm is now used for training. Back-propagation is a method that uses the chain rule to obtain the gradient of the parameters of each layer of the network layer by layer. After the gradient is obtained, each parameter is updated based on it. The continuous cycle of forward propagation and back propagation can gradually update the parameters to obtain parameters with better feature extraction ability. The whole process is training. At present, there are some back-propagation optimization algorithms that can make the training effect better, among which the SGD algorithm is often used. The difference between it and the original BP algorithm is that the SGD algorithm only selects a small batch in the training set to calculate the gradient instead of selecting the entire training set to calculate the gradient like the original BP algorithm. The SGD algorithm with momentum is an improvement to SGD. Unlike SGD, it not only uses the gradient of the currently calculated mini-batch to update the weight but also uses the gradient of the training set that has been calculated to update the weight. Before constructing the 3DCNN structure, the grayscale features, motion features, and edge features have been extracted by the operations of the training samples, as shown in Figure 8.

4.2.3. Channel Rearrangement. The channel rearrangement operation is usually used in conjunction with the grouped convolution. It reorders the output feature maps in each group after the grouped convolution operation according to certain rules, just like shuffling cards, so it is called channel rearrangement. The meaning and specific operation of channel rearrangement will be described with reference to Figure 8, and various input feature data are exemplified in the figure. In order to solve the problem of poor information exchange, the output feature maps of the first grouping convolution can be scrambled and reassigned to each group of inputs of the second grouping convolution operation. In this way, each group of inputs of the second grouped convolution will contain the information of each group of the previous first grouped convolution. Then, the output of each group after the second grouping convolution will contain the information of the first three groups of feature maps input, and the information will flow smoothly among the groups.

Normalization: 0/1 binary convolution brings many benefits, but it introduces a new problem compared to $-1/+1$ binary convolution. Then, the value range of the output of the 0/1 binary convolution is greater than or equal to 0, and the pooling operation will not change the value range. Directly using this as the input for the next binary activation will result in 0 as the threshold. All outputs of the step function for binary activation are 1. Obviously, such binarization is incorrect. In order to solve this problem, the batch normalization operation in the conventional CNN can be added before the binary activation, and the input of the binary activation can be roughly normalized into distribution with 0 as the mean and 1 as the variance. In this way, the output of the binary activation will have both 0 and 1. But in the binary network, the purpose of normalization is only to change the input of the binary activation into a distribution of 0 mean value to cooperate with the subsequent binary activation, so it is only necessary to introduce the mean value for calculation. Even if the variance term is introduced, since the value of the denominator formed by the variance term is always positive, dividing by a positive number will not change the positive or negative nature of the result, which means that it has no effect on the result of subsequent binary activation. Therefore, it is not necessary to introduce a variance term in the normalization of the binary network. In addition, in the batch normalization of the inference process of the conventional network, a fixed mean of the training set is used, and such a method can be directly used in the binary network without changing [21]. The problem is that the mean value of the feature map of the training set cannot represent the mean value of the corresponding position feature map during the actual inference, so the fixed mean value of the training set is used to normalize the feature map during the inference. The mean of the normalized distribution is not necessarily 0. This phenomenon exists in

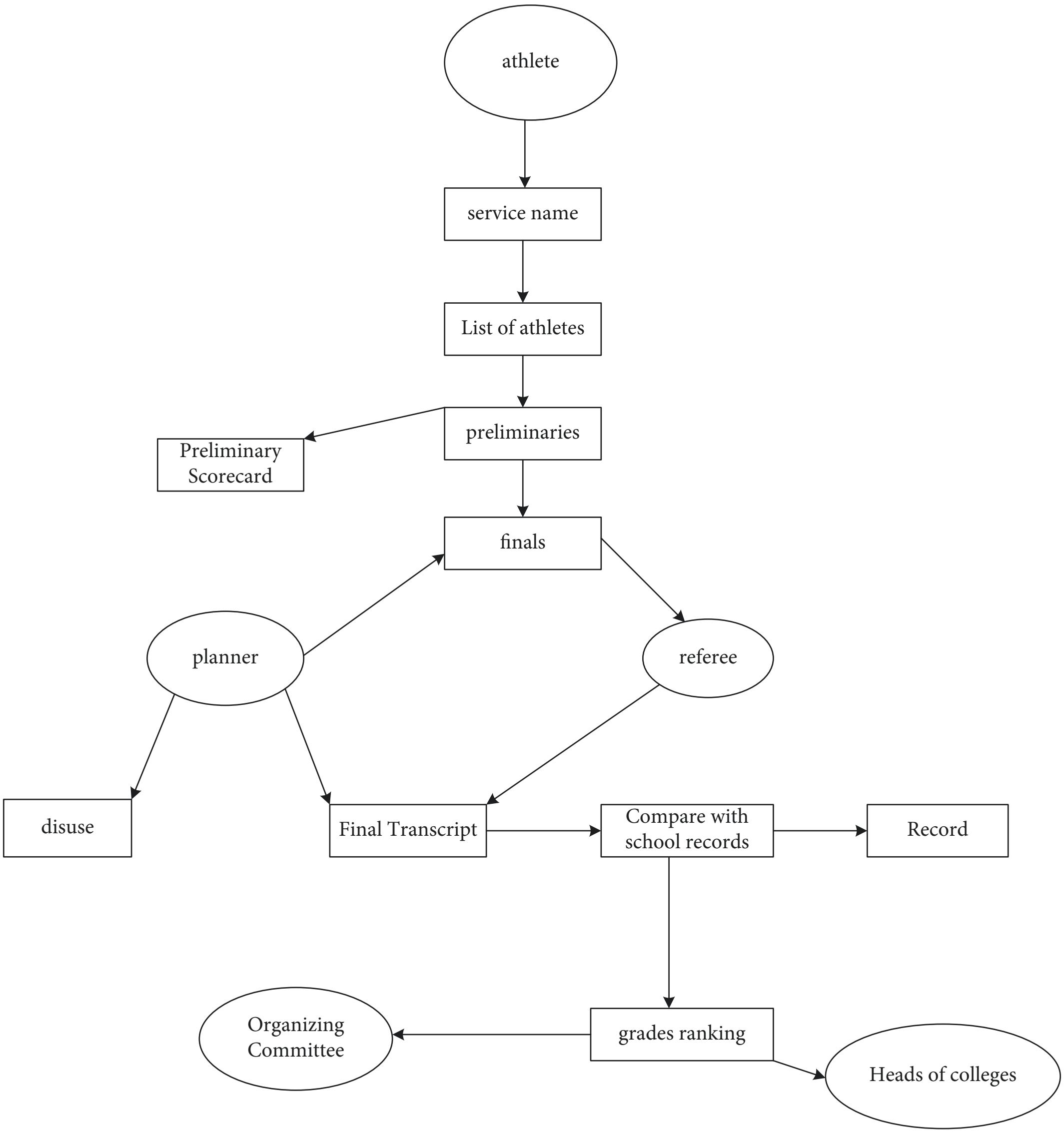


Figure 7: Sports training management information system.

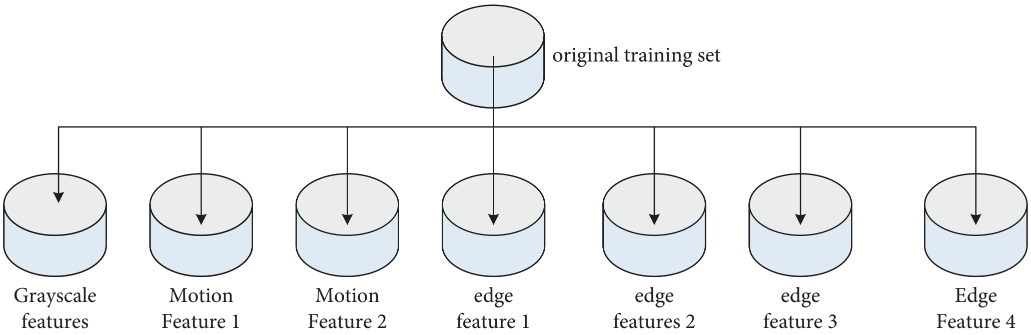


Figure 8: CNN training channel.

regular networks but has little effect because a strictly 0 mean and 1 variance distribution is not necessary for nonlinear activation.

4.3. Experimental Results. In this paper, 60 students from a university are selected as training objects. During the training process, the students' elbows, head, legs, waist, and

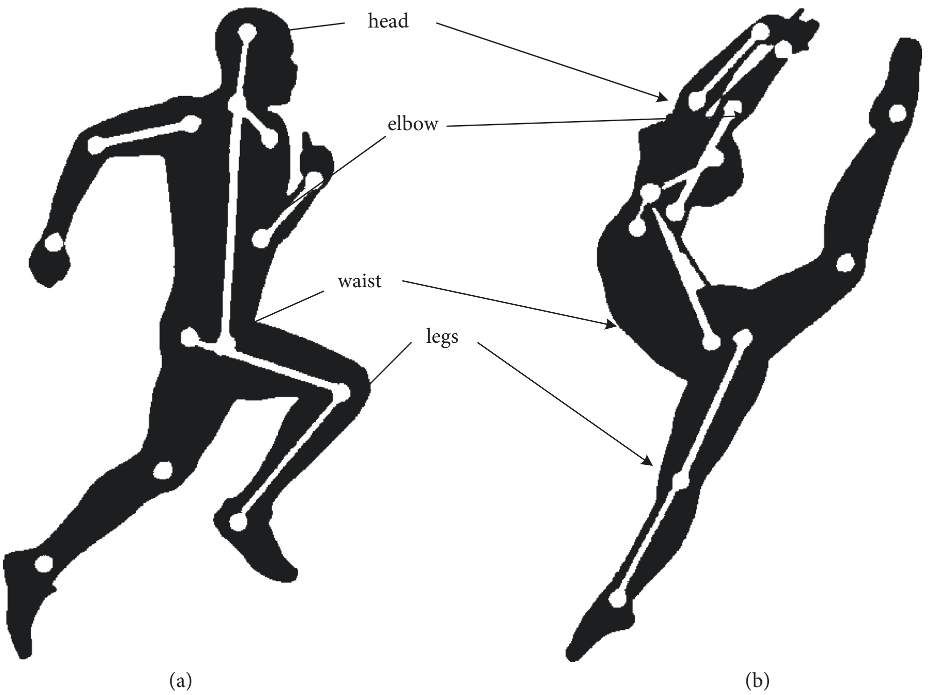


Figure 9: Action recognition feature points.

Table 1: Head training results.

Data	Deviation	Average recognition accuracy (%)
200	0.795 ± 0.027	79.7
400	0.781 ± 0.032	86.6
600	0.736 ± 0.063	92.5

Table 2: Elbow training results.

Data	Deviation	Average recognition accuracy (%)
200	0.865 ± 0.036	76.7
400	0.857 ± 0.042	83.9
600	0.743 ± 0.056	91.6

Table 3: Waist training results.

Data	Deviation	Average recognition accuracy (%)
200	0.665 ± 0.019	81.6
400	0.658 ± 0.022	89.7
600	0.613 ± 0.036	95.1

other parts are selected as identification feature points, as shown in Figure 9.

Firstly, 600 samples are selected from the MSR Action3D dataset for training, and the training results of each part are shown in the tables. Table 1 shows the head training result, Table 2 shows the elbow training result, Table 3 shows the waist training result, and Table 4 shows the leg training result.

Table 4: Leg training results.

Data	Deviation	Average recognition accuracy (%)
200	0.855 ± 0.012	71.7
400	0.848 ± 0.029	79.6
600	0.811 ± 0.038	90.1

It can be seen from Tables 1–4 that after 600 sample training, the accuracy rate of head training is 92.5%, and the accuracy rate of elbow training is 91.6%. The accuracy of waist training is 95.1%, and the accuracy of leg training is 90.1%. Its accuracy rate is more than 90%, so it meets the test requirements.

Finally, this paper conducts actual tests on 60 selected athletes, compares them with traditional calculation methods, and conducts performance tests on the training sports management information system. The overall average recognition accuracy and system recognition rate are obtained, and the results are shown in Figure 10.

It can be seen from Figure 10 that the CNN recognition accuracy test results used in this paper are generally more than 90%, while the traditional recognition accuracy rate is only about 75%, and the highest is not more than 86%. It shows that CNN has a significantly better recognition effect on athletes' movements. And the training management information system in this paper takes about 15.7 s, and the maximum is not more than 10 s. The traditional recognition system takes about 15.7 s, which is about twice the system in this paper. Therefore, it is concluded that the calculation speed of the motion management information system in this paper is faster.

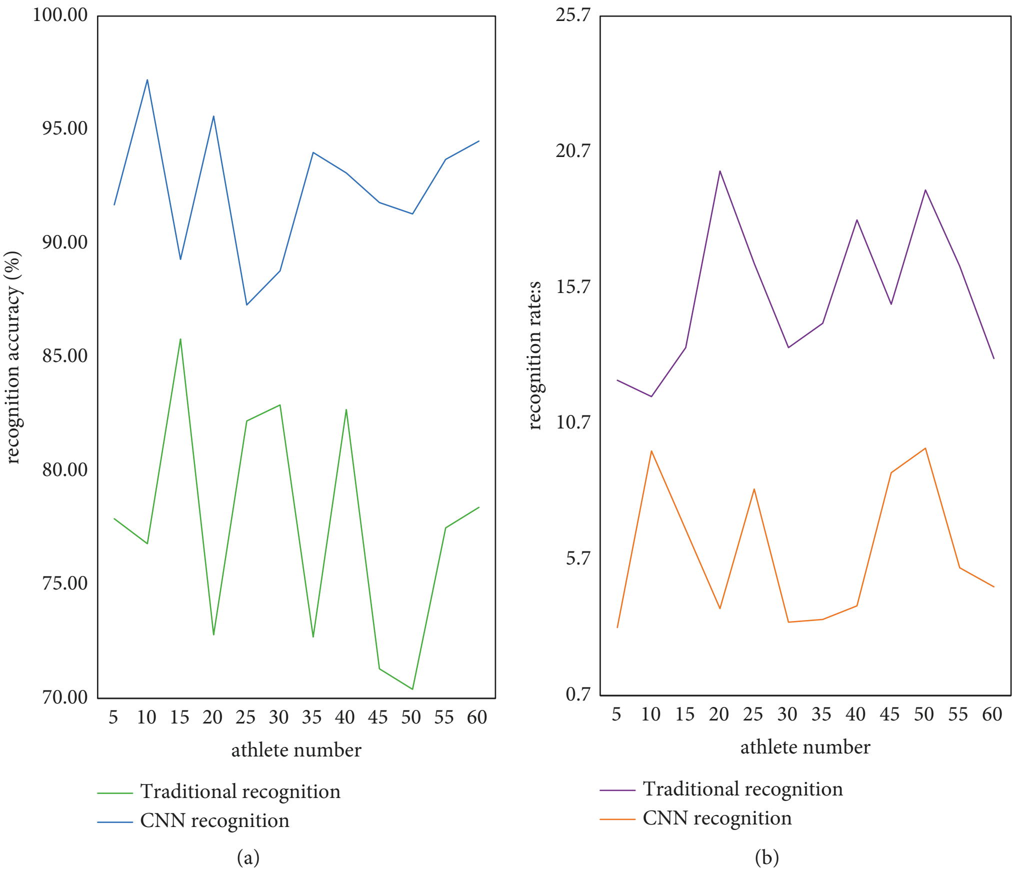


FIGURE 10: Comparison of recognition results. (a) Recognition accuracy. (b) Recognition rate.

5. Conclusions

In the abstract, this paper firstly gave an overview of the overall content of the full text and then introduced the era background of AI in the introduction, introduced the relevant content of action recognition, and summarized the innovations of this paper. The related work part exemplified some related researches, in order to understand the current situation of the related content researched in this paper. Then, in the theoretical research part, the AI-based action recognition was firstly introduced, including the application of AI, the core technology, and the characteristics of neural networks. Finally, the calculation method of the neural network and the content of the experiment were explained in the experimental part, and the movement characteristics of different parts of the athlete were recognized. The results showed that the artificial neural network method in this paper could recognize significantly better, and the calculation time of the sports management information system was lower.

Data Availability

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] S. R. Fanello, I. Gori, G. Metta, and F. Odone, “Keep It Simple and sparse: real-time action recognition,” *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2617–2640, 2017.

[2] O. A. Nakonechna, L. A. Babijchuk, L. A. Babijchuk, and A. I. Bezrodna, “Disturbance of the transmembrane phosphatidylserine asymmetry in hepatocytes as an apoptosis marker under the action of xenobiotics on rats,” *Ukrainian Biochemical Journal*, vol. 90, no. 6, pp. 82–88, 2018.

[3] K. Yu and F. Yun, “Max-margin heterogeneous Information machine for RGB-D action recognition[J],” *International Journal of Computer Vision*, vol. 123, no. 3, pp. 350–371, 2017.

[4] Y. Yanhua, D. Cheng, G. Shangqian, L. Wei, T. Dapeng, and G. Xinbo, “Discriminative multi-instance Multitask learning for 3D action recognition[J],” *IEEE Transactions on Multimedia*, vol. 19, no. 3, pp. 519–529, 2017.

[5] X. S. Nguyen, A. I. Mouaddib, and T. P. Nguyen, “Hierarchical Gaussian descriptor based on local pooling for action recognition[J],” *Machine Vision and Applications*, vol. 30, no. 2, pp. 321–343, 2019.

[6] W. Xiu-hong, S. Xiao-lan, Z. Chuang et al., “Exploring the pharmacological effects and potential targets of paeoniflorin on the endometriosis of cold coagulation and blood stasis model rats by ultra-performance liquid chromatography tandem mass spectrometry with a pattern recognition approach[J],” *RSC Advances*, vol. 9, no. 36, pp. 20796–20805, 2019.

[7] S. Yu, Y. Cheng, L. Xie, and S. Z. Li, “Fully convolutional networks for action recognition,” *IET Computer Vision*, vol. 11, no. 8, pp. 744–749, 2017.

[8] S. P. Yadav, “Emotion recognition model based on facial expressions [J],” *Multimedia Tools and Applications*, vol. 80, no. 6, pp. 1–23, 2021.

[9] H. Wang, O. Dan, and J. Verbeek, “A robust and efficient video representation for action recognition a robust and efficient video representation for action recognition a robust and efficient video representation for action recognition[J],” *International Journal ofuter Vision*, vol. 119, no. 3, pp. 219–238, 2019.

[10] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars, “Rank pooling for action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 773–787, 2017.

[11] S. Laraba, M. Brahimi, and J. Tilmanne, “3D skeleton-based action recognition by representing motion capture sequences as 2D-RGB images[J],” *Computer Animations and Virtual Worlds*, vol. 28, no. 3-4, Article ID e1782, 2017.

[12] T. Revell, “AI can hear a cardiac arrest,” *New Scientist*, vol. 237, no. 3160, p. 6, 2018.

[13] Y. Hou, Z. Li, P. Wang, and W. Li, “Skeleton Optical Spectra-based action recognition using convolutional neural networks,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 3, pp. 807–811, 2018.

[14] W. Peng, Y. Cao, and C. Shen, “Temporal Pyramid pooling based convolutional neural networks for action recognition [J],” *IEEE Transactions on Multimedia*, vol. 27, no. 12, pp. 2613–2622, 2017.

[15] C. Li, Y. Hou, P. Wang, and W. Li, “Joint distance maps based action recognition with convolutional neural networks,” *IEEE Signal Processing Letters*, vol. 24, no. 5, pp. 624–628, 2017.

[16] Q. Ke, S. An, M. Bennamoun, F. Sohel, and F. Boussaid, “SkeletonNet: Mining Deep Part Features for 3-D action recognition,” *IEEE Signal Processing Letters*, vol. 24, no. 6, pp. 731–735, 2017.

[17] J. Liu, G. Wang, and L. Y. Duan, “Skeleton-based human action recognition with Global Context-Aware Attention LSTM networks[J],” *IEEE Transactions on Image Processing*, vol. 27, no. 99, pp. 1586–1599, 2018.

[18] X. Wang, L. Gao, and J. Song, “Beyond frame-level CNN: Saliency-Aware 3-D CNN with LSTM for video action recognition[J],” *IEEE Signal Processing Letters*, vol. 24, no. 99, pp. 510–514, 2017.

[19] A.-A. Liu, N. Xu, W.-Z. Nie, Y.-T. Su, Y. Wong, and M. Kankanhalli, “Benchmarking a Multimodal and Multiview and Interactive dataset for human action recognition,” *IEEE Transactions on Cybernetics*, vol. 47, no. 7, pp. 1781–1794, 2017.

[20] D. Carbonera Luvizon, H. Tabia, and D. Picard, “Learning features combination for human action recognition from skeleton sequences,” *Pattern Recognition Letters*, vol. 99, no. 1, pp. 13–20, 2017.

[21] X. Wang, L. Gao, P. Wang, X. Sun, and X. Liu, “Two-stream 3-D convNet fusion for action recognition in videos with Arbitrary size and length,” *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 634–644, 2018.