Hindawi
Mobile Information Systems
Volume 2022, Article ID 6820073, 9 pages
https://doi.org/10.1155/2022/6820073

Research Article

A Reinforcement Learning-Based Basketball Player Activity Recognition Method Using Multisensors

Yang Bo

University of Shanghai for Science and Technology, Shanghai 200093, China

Correspondence should be addressed to Yang Bo; boyang021@usst.edu.cn

Received 16 May 2022; Revised 29 June 2022; Accepted 4 July 2022; Published 9 August 2022

Academic Editor: Xingsi Xue

It is an effective means to use a computer auxiliary system to assist athletes in training. In this paper, we design a technical activity recognition system for basketball players. The system uses the sensing module bound to the basketball player to collect the activity data and uses the proposed Multilayer Parallel Long Short Term Memory (MP-LSTM) algorithm to recognize the activity. Moreover, in order to extend the working time of the system and reduce the energy consumption of the sensing module, we also utilize the classical reinforcement learning algorithm DQN to adaptively control the sampling frequency of the sensing module for making a trade-off between recognition accuracy and energy consumption. Experiment results show that the recognition accuracy of the proposed MP-LSTM algorithm reaches 94%, while the recognition accuracy of the system remains at about 90% after applying the DQN algorithm, and the energy consumption is reduced by 76%.

1. Introduction

The basketball sport was invented in 1891 by an American physical education teacher named James Naismith. As a fun and easy event, this game is popular with the public, and it gradually spreads over the world [1]. Nowadays, basketball has become a world sport with detailed and rigorous rules, and it has higher requirements on the height, physical strength, and skill of the athletes. The National Basketball Association (NBA), which is the most professional league, has top-notch players in all aspects. The famous center, Shaquille O ′′Ne al, who is 214 cm tall, has a speed of 10.7 seconds for 100 meters and 47 seconds for 400 meters. Michael Jordan can reach 1.2 m in spot bounce as an outstanding basketball star. For the players, correct training methods are crucial to the improvement of performance on the court. The teaching ability and personal experience of the coach are very important to the athletes' training. However, with the continuous development of information technology, using computers and other intelligent devices to collect and analyze data can effectively assist athletes in the training [2].

The computer-aided training means are widely utilized in basketball sports. For example, Mieraisan [3] provides prototype implementations of computer vision algorithms in the sports industry, the main objective of his issue is to develop initial algorithms to solve play-field detection and player tracking in basketball game videos. Shah and Romijnders [4] use the deep learning algorithm to analyze the sports data of players, then predict whether a three-point shot is successful. Kizielewicz and Dobryakova [5] construct a kind of multicriteria decision-making method with an expert evaluation mechanism to analyze the players' data in order to accurately rank NBA players.

Inspired by related works, in this paper, we design a sports data acquisition and analysis system based on multisensors for basketball players. In the system, we use several homogeneous sensing modules to collect the technical motion data from basketball players and transmit the data to the server for analysis and processing. Meanwhile, considering that the battery power of the sensing module is limited and needs to be optimized for energy saving, we design an adaptive sampling frequency control algorithm based on reinforcement learning (RL). RL is one of the main methods in the field of machine learning and intelligent control in recent years. Its idea is to construct a strategy that enables the agent to choose behavior based on the environment in order to achieve the maximum cumulative reward. With the RL means, the agent can choose the most appropriate behavior in different states. Therefore, RL is actually the learning of the mapping strategy, which is defined as "Policy," from environment state to agent behavior. The Deep Q-Network (DQN), a classic RL algorithm, involved in this paper was proposed by the Google DeepMind team in a paper published in Nature in 2015 [6]. The algorithm combines reinforcement learning and deep learning (DL), which makes it a major breakthrough in the field of artificial intelligence.

In this paper, we realize a system for data collection and analysis of basketball players based on multisensor architecture. In order to reduce the energy consumption of the sensing module, the system applies the DQN algorithm to adaptively control the sampling frequency of the nodes, so as to prolong the service cycle of the system. The rest of the paper is organized as follows: In Section 2, we introduce more related works on our issue, including information about the sensor-based recognition system and the DQN algorithm. Section 3 depicts the main structure of our system. Experiments are conducted in Section 4. Section 5 gives the conclusion of our work.

2. Related Works

2.1. Sensor-Based Activity Recognition for Basketball Players. With the maturity of computer technology, computer-based auxiliary training methods have become effective ways in professional sports training. The NBA is a highly professional basketball league, and some commercial organizations, e.g., Stats and Second Spectrum, record players' on-court performance with a multicamera system and provide professional data analysis services to the league teams. In order to personalize the training of players' technical movements and for the sake of flexibility, more studies choose to use an inertial sensing module to record and analyze activity data. Sangüesa et al. [7] establish a technical action data set using positioning sensors. The machine learning algorithm is used to recognize 5 classic basketball skill activities, including floppy offset, pick and roll, press break, post-up situation, and fast breaks, with an accuracy rate of 97.9%. Staunton et al. [8] use Magnetic, Angular Rate, and Gravity (MARG) sensors to measure the Counter-Movement Jump (CMJ) performance metric of elite basketball players. The experiment results show that there is a strong correlation between this index and players' competitive level. Mangiarotti et al. [9] design a wearable activity recognition system based on TinyDuino with an accelerometer and gyroscope, which is specially developed for coaches to track the activities of two or more players at the same time. Hasegawa et al. [10] study the wheelchair basketball game by adding inertia sensors to the athletes' wheelchairs to record the movement parameters so as to improve the athletes' skills in this game. Liu et al. [11] adopt a shapelet-based framework to recognize human activities and take daily life activities and basketball games to test the whole model. To summarize, given that sensing modules have a powerful function and wide popularity, it is an important means of using sensing devices to study basketball sports.

2.2. Energy-Saving Strategy. In many applications involving human activity recognition, it is usually a long-term monitoring job for the purpose of mining the subjects' activity patterns. However, the commonly used sensing modules are usually in small size and limited in power. Therefore, amount of studies try to minimize the energy consumption of the whole system and extend the work time of the system while keeping an acceptable recognition accuracy. There are lots of works on energy-saving strategies. For example, Phan [12] design a special algorithm for the use of GPS and turned on the GPS sensing module only when the geographical position of the user changed. The energy consumption of the entire activity recognition can be effectively reduced by utilizing the algorithm. Ling et al. [13] define the concept of "compression," the essence of which is to control the sampling frequency of the data acquisition module. Meanwhile, the paper also finds the optimal combination of sampling frequency and recognition accuracy through an exhaustive method. Wei et al. [14] reduce the sampling frequency of the sensing module to 2 Hz for energy saving. Meanwhile, the hybrid model structure of the decision tree + support vector machine is used to ensure the recognition accuracy. Gordon et al. [15] set the corresponding sensor configuration for different activities and predict the activity of the next window by referring to the history record. Due to the optimal sensor configuration, the recognition accuracy is improved and the power consumption is reduced. Morillo et al. [16] take advantage of different data sampling frequencies that vary from 32 Hz to 50 Hz to collect data and recognize daily activities (including walking, jumping, and cycling). According to the conclusion of these research studies, energy consumption is positively correlated with identification accuracy. However, all recognition systems are designed to achieve the highest accuracy with the least energy consumption, so making a balance between the two indicators is the core issue discussed in our work.

2.3. Reinforcement Learning. The concept of RL comes from the field of psychology, which is a branch of machine learning methods. RL algorithm can learn from interaction, which is similar to the evolution of the human learning process. The learning subject, named "Agent," learns knowledge according to the rewards or punishments obtained in the process for the purpose of adapting to the environment. The DQN is the combination of Q learning algorithm and deep learning. It takes advantage of the perception ability of deep learning to transform the state into a high-dimensional space and then utilizes the decision-making ability of Q learning method to map the high-dimensional state to a low-dimensional action space, thus solving the problem of dimension explosion. DQN is applied in areas such as game AI and resource optimization. Théate and Ernst [17] study trading algorithms in the stock market. They use DQN to determine the best trading timing and propose a more effective evaluation index for stock trading. The proposed approach can significantly improve both the safety and efficiency of online policy optimization based on the simulation experiment results. Xu et al. [18] proposed a multiexit evacuation simulation based on Deep Reinforcement Learning (DRL) in the simulations on multiexit indoor, which is named as MultiExit-DRL. The proposed method presents great learning efficiency while reducing the total number of evacuation frames in all designed experiments. Sun et al. [19] utilize the DRL method to design a game system based on turn-based confrontation. In the model, they use a Q-learning algorithm to achieve intelligent decision-making. The experiments demonstrate the correctness of the proposed algorithm, and its performance surpasses the conventional DQN algorithm. Leng et al. [20] propose a Color-Histogram (CH) model, which combines the Markov decision process with a DQN algorithm, to solve the problem of color reordering in automotive spray painting workshops.

3. Methodology

3.1. Overall Structure. In this paper, we implement a basketball player skill activity recognition system based on a multisensor architecture. The overall structure is shown in Figure 1.

As shown in Figure 1, the main workflow of the recognition system is as follows:

(i) Data collection: the system collects motion data via sensing modules bound on the basketball player, including acceleration and angular velocity

(ii) Activity recognition: the preprocessed and segmented activity data are identified by the classifier

(iii) Recognition results: the recognition results are presented to the user and also passed to the sampling frequency controller as input parameters

(iv) Frequency control: the sampling frequency controller adaptively changes the frequency according to the recognition results to balance the power consumption and recognition accuracy

The main work of this paper lies in “Classifier” and “Sampling Frequency Controller,” and details of both are given in the following parts.

3.2. Multilayer Parallel LSTM-Based Recognition Algorithm

3.2.1. The Structure of LSTM. The Long Short Term Memory (LSTM) is an upgrade structure of a classical deep learning model, the Recurrent Neural Network (RNN). RNN is a set of the model that can process time sequences and extract time-dependent features. In RNN, neurons in the same layer are connected with each other, and the latter neuron can use information from the former one. Therefore, in the process of network iteration, each neuron contains its own previous information, and its output is affected by the previous neurons, that is, the RNN network can remember and output time-dependent information of samples. However, due to gradient explosion or gradient disappearance occurs when processing long time series, in the traditional RNN network, the historical information that can be preserved by the hidden state is limited. The LSTM proposed by Hochreiter and Schmidhuber [21] for the first time in 1997 has become a classic solution to the two issues. The internal structure of the LSTM unit is shown in Figure 2.

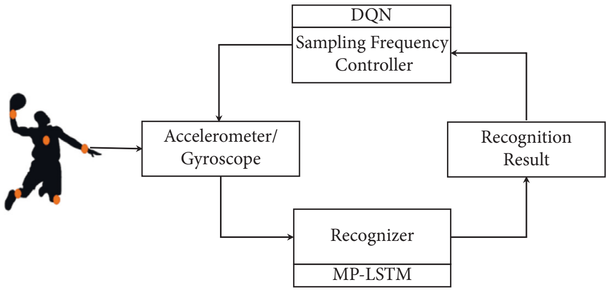

Figure 1: Overall structure of the system.

LSTM is composed of 1 self-connected memory storage unit Ct and 3 gates that control memory information to be saved or forgotten. The Forget Gate determines what kind of information is allowed to pass through or kept. This structure ensures that errors will propagate in the network as a constant and prevents gradient explosion or gradient disappearance. It takes the current input X_t and the previous output state $h_t - 1$ of the hidden layer as input and calculates according to

$$f_t = \sigma\left(W_f X_t + U_f h_{t-1} + b_f\right), \tag{1}$$

where W_f is the weight matrix that maps the hidden layer input to the forget gate, U_f is the weight matrix that connects the output state of the previous moment to the forget gate, b_f is the bias vector, and σ is the activation function, which generally uses the sigmoid function.

The *Input Gate* determines whether the new information can be retained and updated to Ct for storage. The new information is controlled by X_t and h_{t-1}. A new state output s_t is obtained through the tanh function, then the input gate assigns a weight between 0 and 1 to each component of s_t to control how much new information is added to the network. The formulas are shown as follows:

$$\begin{aligned} i_t &= \sigma\left(W_i X_t + U_i h_{t-1} + b_i\right), \\ s_t &= \tanh\left(W_s X_t + U_s h_{t-1} + b_s\right). \end{aligned} \tag{2}$$

The output of the two gates is jointly calculated and updated to C_t after the information passes through the forget gate and the input gate:

$$C_t = f_t * C_{t-1} + i_t * s_t. \tag{3}$$

The *Output Gate* determines what information is output to the next unit. It uses X_t and $h_t - 1$ to calculate o_t, which is the state of the output information, and uses the tanh function to adjust the value of C_t to range [−1, 1].

$$\begin{aligned} o_t &= \sigma\left(W_o X_t + U_o h_{t-1} + b_o\right), \\ h_t &= o_t * \tanh\left(C_t\right). \end{aligned} \tag{4}$$

3.2.2. Multilayer Parallel LSTM. In this part, we introduce the multilayer parallel LSTM proposed in this paper, namely, MP-LSTM. Define an activity sequence $S = [S_1, S_2, \ldots, S_n]$, $S_i \in S$. The details in S_i are shown as follows:

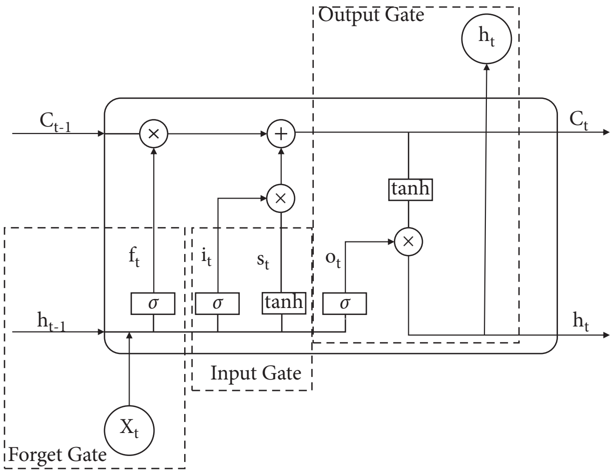

Figure 2: Structure of LSTM unit.

$$S_i = \{1 \le t \le m | [ACC\_X_t, ACC\_Y_t, ACC\_Z_t, GYO\_X_t, GYO\_Y_t, GYO\_Z_t]\}, \tag{5}$$

where ACC_X_t represents the X-axis acceleration data of the tth sampling point in S_i and GYO_X_t represents the angular velocity data. After the input sequence format is defined, the entire network structure is shown in Figure 3. Each input subsequence corresponds to an LSTM unit in the Parallel Layer, and each LSTM unit iterates the sample data in time order, so that the time-dependent information of each sample subfragment can be retained. Meanwhile, each LSTM unit has the same hyperparameters, such as a number of neurons and matrix shape, which ensures that each subfragment is processed in an equal manner.

In the Fuse Layer, there is only one LSTM unit that carries out the iterative calculation on the output matrix of the Parallel Layer and fuses the features of each sub-segment into a complete feature vector h. The elements in h are expanded and put into the Dense Layer for dimension reduction, so as to obtain the feature vectors of higher levels with lower dimensions of the sample. Finally, the softmax function is used to map the feature vector to the final probability results. The softmax function is as follows:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_c^C e^{z_c}}. \tag{6}$$

3.3. DQN-Based Adaptive Sampling Frequency Control Algorithm. When using wearable devices for activity recognition, changes of sampling rate affect the recognition accuracy and energy consumption, and the optimal sampling frequency is also different for activities. The algorithm proposed in this part carries out adaptive sampling frequency control on the sensing modules. It adjusts the sampling frequency according to the current activity, balances the recognition accuracy and energy consumption, and improves the overall performance of the system.

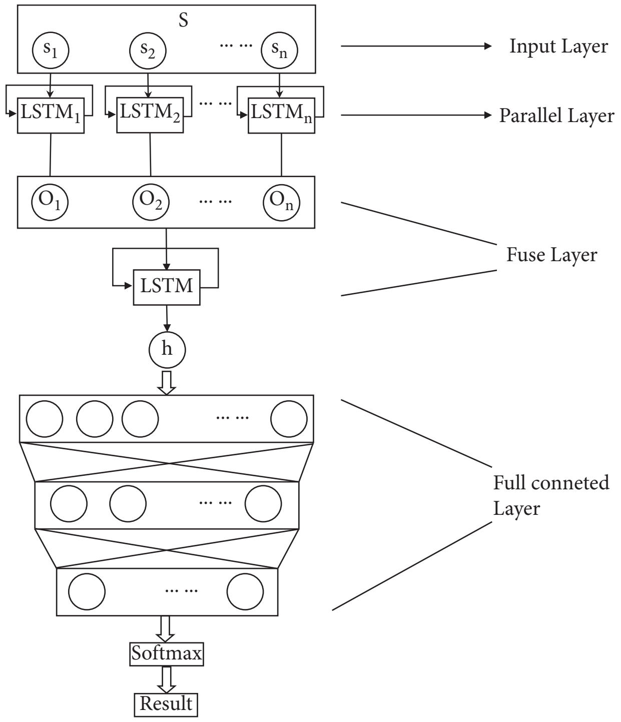

Figure 3: Structure of MP-LSTM.

3.3.1. Problem Mapping. Assume that the sensing module used can support k sampling rates, denoted as $F = (f_1, f_2, \ldots, f_k), f_1 < f_2 < \cdots < f_k$. The energy consumption of the module varies at different sampling rates. Let $P = (P_1, P_2, \ldots, P_k)$ denotes the sampling power set, the larger the sampling frequency f is, the larger the corresponding power consumption is, which means that $P_1 < P_2 < \cdots < P_k$. Suppose that there are m kinds of activities in the target activity set, denoted as $Y = (y_1, y_2, \ldots, y_m)$. Define a contiguous sample sequence with the label as Q, which is shown as follows:

$$Q = \{1 \le i \le n | (X_i, Y_i)\}, \tag{7}$$

where X_i is the activity data of the ith window in the sequence and Y_i is the corresponding activity label. We hope to design an activity recognition model to achieve high recognition accuracy, or a low recognition error rate, and meanwhile, the model has a sampling frequency selection strategy to minimize the overall energy consumption. The issue is depicted as follows:

$$\min_{\theta, f_1, \ldots, f_N} \sum_{t=1}^{N} l\{\hat{y}_t \neq y_t\} + \lambda \sum_{t=1}^{N} P_t, \tag{8}$$

where $\hat{y}_t = \underset{y}{\text{argmax}}\, p(y|x_t; \theta)$ is the recognition results, $l\{\hat{y}_t \neq y_t\}$represents the possibility that the output result is inconsistent with the actual label, and λ refers to the weight of energy consumption. For the convenience of calculation, $l\{\hat{y}_t \neq y_t\}$ can be replaced by the cross entropy of predicted probability distribution and behavior label, which is

$$l\{\hat{y}_t \neq y_t\} = -\log\ pb(y_t|x_t;\theta), \tag{9}$$

where $pb(Y|X)$ is the probability that sample X belongs to category Y. Therefore, formula (8) can be changed as

$$\min_{\theta, f_1,\ldots,f_N} \sum_{t=1}^{N} -\log\ p(y_t|x_t;\theta) + \lambda \sum_{t=1}^{N} P_t. \tag{10}$$

3.3.2. Algorithm Description. In the traditional Q learning algorithm, the state space and the action space are usually discrete and finite, and the value function $Q(s,a)$ can be stored in tables. But in our work, the state space S is continuous and infinite, which means that the $Q(s,a)$ cannot be stored in tables. Therefore, DQN is used to train the model. The algorithm is shown in Figure 4.

The frequency controller adjusts the sampling rate according to the activity record. This process can be abstracted into a Markov decision process, so the sampling rate selector can be realized by RL. Here, we give the elements of the RL-based sampling frequency controller, including state space, action space, reward function, and action strategy:

(i) State space

The output activity probability distribution vector from the classifier is taken as the current state, which is

$$S = \{(pb_1, pb_2, \ldots, pb_i, \ldots, pb_m)|pb_i \in [0,1]\}, \tag{11}$$

where m is the number of activity types and pb_i is the probability that the sample label is i.

(ii) Action space

According to the previous assumption, different kinds of activities correspond to different sampling frequencies. The action space A is defined as follows:

$$A = (a_1, a_2, a_3, \ldots, a_i, \ldots, a_k), \tag{12}$$

where a_i means that the sampling frequency f_i is chose when collecting data with activity label i.

(iii) Reward function

Rewrite formula (10):

$$\begin{aligned} &\min_{\theta, f_1,\ldots,f_N} \sum_{t=1}^{N} -\log\ p(y_t|x_t;\theta) + \lambda \sum_{t=1}^{N} P_t \\ &\quad = \min_{\theta, f_1,\ldots,f_N} \sum_{t=1}^{N} -\log\ p(y_t|x_t;\theta) + \lambda P_t \\ &\quad = -\max_{\theta, f_1,\ldots,f_N} \sum_{t=1}^{N} \log\ p(y_t|x_t;\theta) - \lambda P_t. \end{aligned} \tag{13}$$

Then, the reward function is set as

$$R_t(s) = \log pb(y_t|x_t;\theta) - \lambda P_t. \tag{14}$$

Given that it is difficult to directly obtain the energy consumption value at different sampling rates in the experimental environment, we take a method of approximation instead. Assume that the energy consumption of a sensing device for single data sampling is fixed and denoted as P_0, then the power consumption of the sampling rate f_t is $P_t = f_t * P_0$. So the reward function is

$$R_t(s) = \log pb(y_t|x_t;\theta) - \lambda f_t P_0. \tag{15}$$

(iv) Action strategy

The ε-greedy is widely used in RL as an action strategy. It is an extension of the traditional greed mechanism. The agent chooses the action in the space A by

$$a = \operatorname*{argmax}_{a\in A} Q(s,a). \tag{16}$$

The agent can select the actions randomly and freely under the ε-greedy mechanism. The specific strategies are as follows:

(1) Generating a random number in the [0, 1], i.e., num = Random (0, 1)

(2) If num < ε, then select the action according to the greedy mechanism, i.e., $a = \operatorname*{argmax}_{a\in A} Q(s,a)$

(3) Else randomly select an action in A, i.e., $a = \text{Random}(A)$

ε is the greed degree and in the range of (0, 1). The agent may choose the random action easily if the ε is a small value, and the convergence speed gets slow correspondingly. Therefore, the choice of ε should be based on careful consideration.

We use pseudo code to demonstrate the DQN in Algorithm 1.

4. Experiment and Results

4.1. Dataset

4.1.1. Data Collection. We introduce the dataset used in the following experiments. We have 20 volunteers in the data collection, 10 of whom are basketball players and others are college students. Their physical information is given in Table 1. The whole activity dataset consists of 6 activities, including standing, standing dribble, penalty shot, jump shot, running, and running dribble. In order to unify these movements, we give a rigorous definition of these activities. For example, we define "standing dribble" as "Subjects should stand with their feet naturally apart, and try not to move, while keeping control of the basketball with one hand." These definitions can reduce the data diversity of the same activity.

Each subject is bound with 5 sensing modules, which are in the left thigh, right thigh, left leg, right leg, and torso in the data collecting process. The original sampling frequency is set to 50 Hz. According to experience from related work [22], we set the size of the sliding window as 3s, i.e., 150 data sampling points. Each subject performs each activity for 5 times, 3 min for each time.

4.1.2. Data Downsampling. In our work, we need to discuss the influence of different sampling frequencies on the recognition accuracy and energy consumption. The original

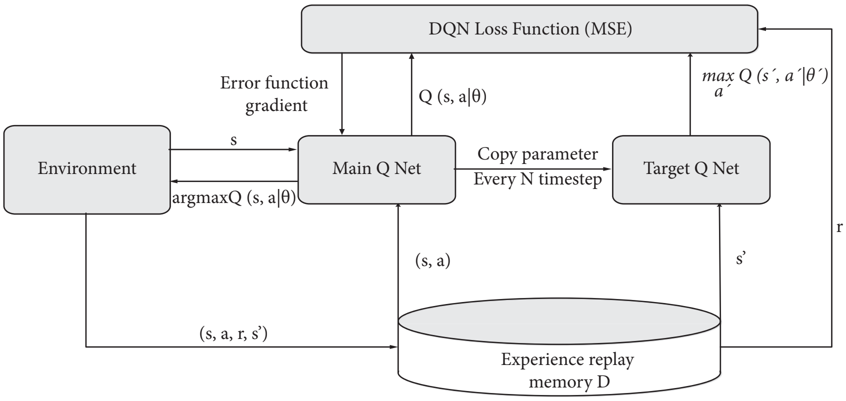

Figure 4: The DQN algorithm.

```
Init: $D$.size: = $N$, Main_Qnet.parameter: = $\theta$, Target_Qnet.parameter: = $\theta'$
For each episode:
    Init first sample $x_t = x_0$, Init first state $s_t = s_0$
    For each period:
        Choose the proper $a_t$ according to the strategy $\varepsilon$, $a_t$: $= \underset{a}{\operatorname{argmax}} Q(s_t, a|\theta)$
        Execute $a_t$, update reward $r_t$, and next sample $x_{t+1}$
        Update state $s_{t+1}$
        Put experience sample $(s_t, a_t, r_t, s_{t+1})$ to $D$
        Random select experience sample $(s_j, a_j, r_j, s_{j+1})$
        If end_of_episode: $y_j = r_j$
        Else: $y_j = r_j + \gamma \max_{a'} Q(s_{j+1}, a'|\theta')$
        Update $\theta$ using loss function: $(y_j - Q(s_j, a_j|\theta))^2$
        Every $C$ timestep: $\theta' = \theta$
    Until period end
End episode end
```

Algorithm 1: The DQN algorithm.

Table 1: Physical information of the volunteers.

Item	Value
Height	173.3 (±10.5)cm
Weight	68.2 (±9.7)kg
Age	26.5 (±3.5)
Sex	15 males : 5 females

frequency is fixed (50 Hz), so it is necessary to conduct the downsampling to obtain data with different sampling frequencies. We downsample all the active data for 5 times to compose new datasets as given in Table 2.

4.2. Performance of MP-LSTM Recognition Algorithm. In this part, we test the recognition ability of the proposed MP-LSTM algorithm through experiments. We use the original data set described above for the recognition experiment with10 fold cross-validation. First, model parameters are set through experiments, including the number of LSTM units in the parallel layer, the number of hidden neurons in each LSTM unit, and the number of hidden neurons in the fuse layer. The experiment results are shown in Table 3.

The results above suggest that the increase in the number of LSTM units in the parallel layer improves the recognition accuracy. However, the accuracy rate is about 94.5%, which does not increase with more LSTM units. So we finally set this parameter as 6. The number of hidden neurons in each parallel LSTM affects the number of feature units sent to the next layer. Generally, a high feature dimension means powerful representation ability. However, according to the experiment, the best result is achieved when the number of hidden neurons is 24, and higher feature dimensions bring no significant promotion. Based on the same reason, we set the number of hidden neurons in the fusion layer as 64.

We use the above parameters to build the MP-LSTM model and compare it with similar algorithms. Table 4 gives the comparison results.

According to the comparison results, the recognition accuracy of the algorithm proposed in this paper reaches 94.77%, which is better than other similar algorithms. Benefit from the parallel structure of the model, the recognition time is effectively reduced and the recognition efficiency is improved.

4.3. Performance of DQN Sampling Frequency Control Algorithm

4.3.1. Sampling Frequency vs Recognition Accuracy. In order to simplify the problem, we simplify the relation between frequency and energy consumption in the former parts. We

Table 2: Information of data downsampling.

Downsampling rate	New frequency (Hz)	Size of single sample (sample points)
2	25	75
4	12.5	38
5	10	30
8	6.3	19
10	5	15

Table 3: Relation between the MP-LSTM parameter and recognition accuracy.

No. of LSTM units in parallel layer	2	4	6	8	10
Recognition accuracy	93.88	94.19	94.57	94.00	92.67
No. of hidden neurons in parallel LSTMs	8	16	24	32	40
Recognition accuracy	89.14	93.08	94.33	93.91	93.85
No. of hidden neurons in fuse layer	16	32	48	64	80
Recognition accuracy	92.61	92.87	93.26	94.32	94.03

Table 4: Comparison with other algorithms.

Methods	Accuracy (%)
Dynamic time warping [23]	87.26
LSTM [24]	90.55
CNN [25]	90.78
PCA + SVM [26]	92.31
MP-LSTM	94.77

assume that the energy consumption is proportional to the sampling frequency. Therefore, the problem between energy consumption and recognition accuracy is transformed into making trade-off between sampling frequency and accuracy. We explore the influence of different sampling frequencies on the recognition accuracy in this part. Datasets of different sampling frequencies are recognized using the 10-fold cross-validation method, and the average recognition accuracy is given in Table 5. The $A1$–$A6$ in the table refers to the 6 activities, which are standing, standing dribble, penalty shot, jump shot, running, and running dribble.

It can be seen from Table 5 that the recognition accuracy of each activity decreases with the sampling frequency. However, the model has different performances on recognizing different activities. It achieves better results on standing, standing dribble, and free throw than on running and running dribble. It is supposed that the first 3 activities have smaller movements and concentrate on the upper limbs compare with the other 3 ones.

4.3.2. λ Determination.

In this part, we realize the sampling frequency controller using the DQN algorithm, and the main parameter to be determined is λ. During training, 50 Hz is selected as the default frequency for the first sample of each sequence to ensure that enough features can be extracted. Meanwhile, we set the energy consumption of single sampling as $P_0 = 1$ to restrict the reward function in [0, 1].

After choosing training the sampling frequency controller, we evaluate the performance on the test set, focusing on the overall recognition accuracy and energy savings. Moreover, we define the Energy Saving Rate (ESR) to measure the energy saving effect. ESR refers to the percentage of energy saved by the currently selected sampling frequency compared to the original frequency. For example, the currently selected sampling frequency of 10 Hz provides an 80% ESR compared to the original sampling frequency of 50 Hz.

Figure 5 shows the relationship among recognition accuracy, ESR, and weight parameter λ. According to the figure, with the decrease of the weight parameters λ, the recognition accuracy continuously increases, while the ESR gradually reduces. This is because when λ is large, the model pays more attention to energy saving and tends to choose a lower sampling frequency, which also leads to a lower recognition accuracy. Correspondingly, when λ is small, the model pays more attention to the recognition accuracy, and the energy consumption is relatively high. In addition, when $\lambda > 0.5$, the accuracy rate increases obviously, while the energy saving rate decreases slightly, and vice versa. $\lambda = 0.5$ is an equilibrium state of the model, which makes the accuracy and ESR balanced. Therefore, λ is set to 0.5 in the subsequent experiments.

4.3.3. Overall Performance.

Table 6 shows the recognition accuracy and ESR of each activity when $\lambda = 0.5$. According to the statistics, activities such as standing, standing dribble, and penalty shot have relatively high recognition accuracy compared with other 3 activities. Moreover, the ESRs of the first 3 activities overcome the values of A4–A6. We believe that the intensity of A1–A3 is lower and their data barely fluctuates, which means that data with a low sampling frequency is enough for recognizing these activities.

In addition, we compare the performance of the proposed model with other related works in Table 7. According to the results, the accuracy of the proposed model is lower than the methods in Reference [28], but with little difference. However, our model performs much better than the other 2 works in energy saving. To summarize, the proposed model can effectively reduce the energy consumption of the equipment, while ensuring a certain recognition accuracy.

Finally, in order to verify the performance of the DQN-based frequency control algorithm in the real scene, we

Table 5: Sampling frequency vs recognition accuracy.

Sampling frequency	Recognition accuracy (%)						
	*A*1	*A*2	*A*3	*A*4	*A*5	*A*6	Sum
50	98.92	97.80	95.75	91.41	93.30	89.37	94.43
25	98.51	97.12	95.44	91.01	92.91	88.44	93.91
12.5	98.10	95.31	94.26	89.90	90.08	86.27	92.32
10	96.25	94.87	91.78	88.52	89.29	83.37	90.68
6.3	90.71	87.52	87.22	86.08	86.27	80.56	86.39
5	88.27	83.15	85.74	84.70	83.75	79.46	84.18

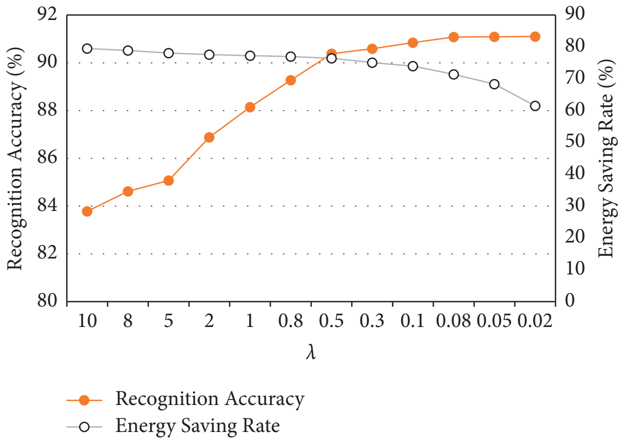

Figure 5: Influence on the recognition accuracy and ESR from different λ.

Table 6: RA vs ESR.

Item	*A*1 (%)	*A*2 (%)	*A*3 (%)	*A*4 (%)	*A*5 (%)	*A*6 (%)	Sum (%)
RA	96.25	95.84	93.21	89.46	85.69	81.83	90.38
ESR	79.20	80.71	79.94	77.52	72.10	68.81	76.38

RA: recognition accuracy; ESR: energy saving rate.

record the change of the remaining battery of sensoring module over time in 3 situations. Figure 6 gives the test results, where we have the following:

(i) Without DQN means to execute the recognition without DQN

(ii) DQN means to execute the recognition with DQN

(iii) NAN means no recognition

At the beginning of the experiment, the equipment battery is 100%. It takes 2.75 hours for the battery to decrease by 10% in "Without DQN," while the DQN algorithm slows down the speed of power decline by 39%. Compared with the "Without DQN" situation, the energy consumption of the sensing module is significantly reduced by 31%–39% in "DQN." At the end of these 2 situations, due to the fast speed of energy consumption in "Without DQN," the remaining power is 50% less than that in "DQN," indicating that DQN-based method is energy-efficient in the real scene.

Table 7: Compare with related works.

Related works	Recognition accuracy (%)	Energy saving rate (%)
[27]	89	50
[28]	92	28
Our work	90	76

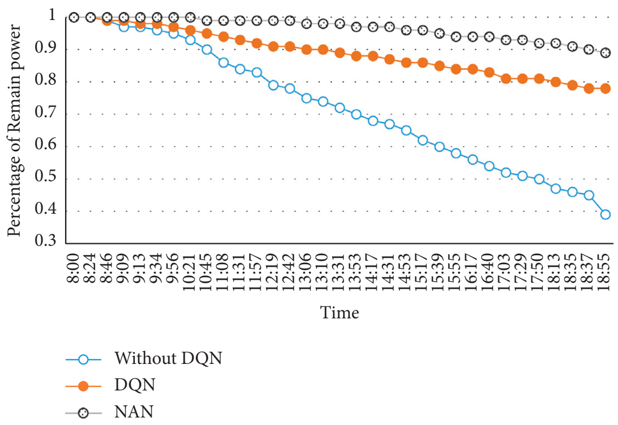

Figure 6: Test results under real scene.

5. Conclusions

In this paper, we implement an activity recognition system for basketball players using multisensor architecture. The MP-LSTM is utilized in the system for activity recognition, and the overall accuracy reaches 94.77%. Meanwhile, in order to prolong the working time of the sensing module, the DQN-based sampling frequency strategy is applied to adaptively control the sampling frequency of the module for energy saving. The experiment results show that the proposed method can reduce 76% of the energy consumption while maintaining the recognition accuracy at about 90%, which outperforms other related works.

The accuracy of activities such as jump shot, running, running dribble is relatively low. It is supposed that a great range of movements brings more noise to the activity data, which is a great challenge for the recognition model. In future work, we focus on this issue to improve the recognition ability of the model on activities with large movements.

Data Availability

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

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] J. Furey, "Basketball in America: a history," 2017, https://www.factmonster.com/sports/basketball/basketball-america-history.

[2] Y. J. Chang, "The use of computer-aided teaching software in the teaching of basketball tactics," *Applied Mechanics and Materials*, vol. 347-350, pp. 2777–2780, 2013.

[3] M. Mieraisan, *Basketball game analyzing based on computer vision*, Doctorial dissertation, Tampere University, Tampere, Finland, 2013.

[4] R. Shah and R. Romijnders, "Applying deep learning to basketball trajectories," *Computer Science: Neural and Evolutionary Computing*, vol. v1, 2016.

[5] B. Kizielewicz and L. Dobryakova, "MCDA based approach to sports players' evaluation under incomplete knowledge," *Procedia Computer Science*, vol. 176, pp. 3524–3535, 2020.

[6] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[7] A. A. Sangüesa, T. B. Moeslund, C. H. Bahnsen, and R. B. Iglesias, "Identifying basketball plays from sensor data; towards a low-cost automatic extraction of advanced statistics," in *Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 894–901, New Orleans, LA, USA, November 2017.

[8] C. A. Staunton, J. J. Stanger, W. Wundersitz, A. Gordon, E. Custovic, and I. Kingsley, "Criterion validity of a MARG sensor to assess countermovement jump performance in elite basketballers," *The Journal of Strength & Conditioning Research*, vol. 35, no. 3, pp. 797–803, 2021.

[9] M. Mangiarotti, F. Ferrise, S. Graziosi, F. Tamburrino, and M. Bordegoni, "A wearable device to detect in real-time bimanual gestures of basketball players during training sessions," *Journal of Computing and Information Science in Engineering*, vol. 19, no. 1, Article ID 11004, 2018.

[10] R. Hasegawa, A. Uchiyama, and T. Higashino, "Maneuver classification in wheelchair basketball using inertial sensors," in *Proceedings of the Twelfth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, pp. 1–6, IEEE, Kathmandu, Nepal, November 2019.

[11] L. Liu, Y. Peng, M. Liu, and Z. Huang, "Sensor-based human activity recognition system with a multilayered model using time series shapelets," *Knowledge-Based Systems*, vol. 90, pp. 138–152, 2015.

[12] T. Phan, "Intelligent energy-efficient triggering of geolocation fix acquisitions based on transitions between activity recognition states," in *Proceedings of the International Conference on Mobile Computing, Applications, and Services*, pp. 104–121, Springer International Publishing, Paris, France, November, 2013.

[13] X. Ling, M. Yu, and W. Kai, "Adaptive compressed classification for energy efficient activity recognition in wireless body sensor networks," in *Proceedings of the 2018 4th International Conference on Big Data Computing and Communications*, pp. 41–45, Chicago, IL, USA, August 2018.

[14] Z. Wei, Y. Yoshihara, D. Tang, N. Kubota, and N. Tay, "Energy-efficient activity recognition on smartphone," in *Proceedings of the Third International Conference on Computing Measurement Control & Sensor Network*, pp. 1–4, IEEE, Matsue, Japan, May 2016.

[15] D. Gordon, J. Czerny, T. Miyaki, and M. Beigl, "Energy-efficient activity recognition using prediction," in *Proceedings of the 2012 16th International Symposium on Wearable Computers*, pp. 29–36, Newcastle, UK, June 2012.

[16] L. Morillo, L. Gonzalez-Abril, J. Ramirez, and M. de la Concepcion, "Low energy physical activity recognition system on smartphones," *Sensors*, vol. 15, no. 3, pp. 5163–5196, 2015.

[17] T. Théate and D. Ernst, "An application of deep reinforcement learning to algorithmic trading," *Expert Systems with Applications*, vol. 173, no. 4, Article ID 114632, 2021.

[18] D. Xu, X. Huang, J. Mango, X. Li, and Z. Li, "Simulating multi-exit evacuation using deep reinforcement learning," *Machine Learning*, vol. 25, pp. 1542–1564, 2020.

[19] Y. Sun, B. Yuan, T. Zhang, B. Tang, W. Zheng, and X. Zhou, "Research and implementation of intelligent decision based on a priori knowledge and DQN algorithms in wargame environment," *Electronics*, vol. 9, no. 10, pp. 1–21, 2020.

[20] J. Leng, C. Jin, A. Vogl, and H. Liu, "Deep reinforcement learning for a color-batching resequencing problem," *Journal of Manufacturing Systems*, vol. 56, pp. 175–187, 2020.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] S. Bersch, D. Azzi, R. Khusainov, E. Achumba, and J. Ries, "Sensor data acquisition and processing parameters for human activity classification," *Sensors*, vol. 14, no. 3, pp. 4239–4270, 2014.

[23] S. Seto, W. Zhang, and Y. Zhou, "Multivariate time series classification using dynamic time warping template selection for human activity recognition," in *Proceedings of the 2015 IEEE Symposium Series on Computational Intelligence*, pp. 1399–1406, Cape Town, South Africa, December 2015.

[24] Y. Chen, K. Zhong, Z. Ju, Q. Sun, and X. Zhao, "LSTM networks for mobile human activity recognition," in *Proceedings of the 2016 International Conference on Artificial Intelligence: Technologies and Applications*, pp. 50–53, Bangkok, Thailand, January, 2016.

[25] C. A. Ronaoo and S. B. Cho, *Evaluation of Deep Convolutional Neural Network Architectures for Human Activity Recognition with Smartphone Sensors*, in *Proceedings of the Korea Information Science Society*, pp. 858–860, Montreal Canada, December 2015.

[26] C. A. Ronao and S. B. Cho, "Recognizing human activities from smartphone sensors using hierarchical continuous hidden Markov models," *International Journal of Distributed Sensor Networks*, vol. 13, Article ID 155014771668368, 2017.

[27] Z. Yan, V. Subbaraju, C. Dipanjan, M. Archan, and A. Karl, "Energy-efficient continuous activity recognition on mobile phones: an activity-adaptive approach," in *Proceedings of the International Symposium on Wearable Computers*, pp. 17–24, IEEE Computer Society, Newcastle, UK, June 2012.

[28] V. Q. Viet, H. M. Thang, and D. Choi, "Adaptive energy-saving strategy for activity recognition on mobile phone," in *Proceedings of the IEEE International Symposium on Signal Processing & Information Technology*, pp. 95–100, IEEE, Ho Chi Minh City, Vietnam, December 2012.