International Journal for Housing Science and Its Applications
Publish August 5, 2025. Volume 46, Issue 3 Pages 4524-4535

https://doi.org/10.70517/ijhsa463388

A Study on Posture Recognition and Correction Based on Deep Convolutional Neural Network for Rehabilitation of Lower Limb Motor Function Injury in Sports

Lei Wang[1,*]
[1] Hunan Communications Vocational and Technical College, Changsha, Hunan, 410132, China
Corresponding authors: (e-mail: abc888guan@163.com).

Abstract The rapid development of sports training technology promotes the improvement of physical fitness of the sports population, but sports lower limb functional injuries inevitably occur. In order to realize effective lower limb rehabilitation, the recognition of rehabilitation posture for sports lower limb motor function injury becomes crucial. In this paper, we preprocess the sports lower limb movement action data through the method of error calibration and data normalization. The DeepConvLSTM neural network is proposed by combining convolutional neural network (CNN) and long short-term memory (LSTM) recurrent neural network, and the gesture recognition model based on DeepConvGRU neural network is constructed by integrating gated recurrent unit (GRU). The performance of the model on the lower limb motor function injury rehabilitation task is evaluated through experiments. The DeepConvGRU model in this paper achieved 96.83% and 99.11% accuracy on the UCISmartphone dataset and RehaLab-412 dataset, respectively, demonstrating good model performance.

Index Terms convolutional neural network, long and short-term memory recurrent neural network, DeepConvGRU neural network, sports injury rehabilitation, sports movement

I. Introduction

Lower limb functional injuries in sports are a common type of sports injury, and through rehabilitation training, the rehabilitation process can be accelerated, joint mobility can be improved, and complications can be avoided [1]-[3]. The purpose of rehabilitation training is to help restore the function of the lower limb bones through appropriate movement and exercise, improve muscle coordination by increasing muscle strength, as well as prevent and reduce the risk of recurrent fractures [4]-[6]. Rehabilitation training can promote blood circulation, enhance muscle group strength and joint flexibility, thereby shortening rehabilitation time and improving rehabilitation outcomes [7], [8]. However, in rehabilitation training, the correctness of posture is closely related to the outcome of rehabilitation, and incorrect posture not only does not promote rehabilitation, but also aggravates the injury, so it is crucial to recognize the rehabilitation posture and correct it [9]-[11].

Rehabilitation training for lower limb sports injuries needs to be planned according to the specific situation, including muscle strengthening, balance training, joint mobility exercises, etc [12], [13]. In the process of rehabilitation training, attention should be paid to gradually strengthening the training intensity to avoid excessive fatigue [14]. Muscle strengthening training includes thigh muscle strengthening training such as straight leg raise, lunge squat, and leg curl, calf muscle strengthening such as heel lift and kneeling leg curl, and gluteal muscle strengthening training such as gluteal bridge and lateral gluteal bridge [15]-[17]. Balance training includes single-leg support exercises, walking balance exercises, and dynamic balance exercises [18], [19]. In terms of joint mobility exercises, there is a gradual transition from early bed training, including knee bending, extension and other movements, to the use of aids such as walker crutches and other activities such as going up and down stairs [20]-[22]. In rehabilitation care, not only should we focus on appropriate rehabilitation training, but also on psychological care and analgesic care, which can detach them from the vicious cycle state of pathogenic factors such as psychological stress and pain stress, improve overall comfort, strengthen the confidence in rehabilitation treatment, and accelerate the regression of the condition and the improvement of the prognosis [23]-[26].

Due to the risk of recording error of the action collector and the principle error of the inertial guidance data solving itself, this paper proposes the pre-processing process and method of the sports lower limb sports action data with error calibration and data normalization to ensure the quality of the lower limb sports data set. Under the premise of guaranteeing the quality of sports lower limb movement data, a sports lower limb motor function injury rehabilitation posture recognition model is constructed to realize the recognition of sports lower limb motor function

injury rehabilitation posture and monitor the rehabilitation status of lower limb motor function injury. By combining convolutional neural network (CNN) and long short-term memory (LSTM) recurrent neural network, and for the problem that DeepConvLSTM network is difficult to be trained, a fusion model DeepConvGRU based on convolutional neural network and gated recurrent units (GRUs) is proposed, which automatically extracts the features and models the time-dependence relationship, and improves the convergence speed of the model. The performance of this paper's model on the task of lower limb motor function injury rehabilitation is explored through the sports lower limb motor function injury rehabilitation posture recognition experiments.

II. Methods of pre-processing lower limb movement data in sports

The main content of this chapter is the preparation of sports lower limb motor action data in the study of posture recognition for rehabilitation of sports lower limb motor function injury. The preprocessing of sports lower limb motor action data is achieved through error calibration and data normalization.

II. A.Error calibration

Microelectromechanical system (MEMS)-based inertial sensors have the advantages of easy integration, low price, and low power consumption, which provide the hardware basis for making portable inertial guidance devices; however, at the same time, MEMS devices tend to show the defects of poor stability and insufficient accuracy, which are mainly reflected in the larger drift error and higher randomness of the noise model, which leads to the easy accumulation of error data.

In the zero-input state, the output of the sensor tends to behave as a smooth random process after entering a long time steady state. The statistical properties of a smooth stochastic process are to have a constant mean and a constant variance, i.e., the sensor output behaves as if it varies around a fixed value (the mean) with a fixed degree of dispersion (the mean-variance value). This initial state deviation does not vary with time under the steady-state assumption and is therefore a static error that can be eliminated offline:

$$\begin{aligned} \mu &= E(z_t) = \int_{-\infty}^{\infty} zp(z)dz \\ \sigma_z^2 &= E\left[(z_t-\mu)^2\right] = \int_{-\infty}^{\infty}(z-\mu)^2 p(z)dz \end{aligned} \tag{1}$$

In fact, the accurate error model of inertial devices is very complex, in addition to static error, it also contains various types of error components with time-varying coefficients and cross-coupling terms. Compensation algorithms have been built in the device for these complex error components, so there is no need to carry out additional modeling processing in the data application stage. In this section, only a simplified error model is considered for static error calibration, and this coarse calibration scheme is easy to operate and simple to compute, so it can be easily used by common users. The accuracy of the inertial guidance devices tends to be relatively stable under short-term applications, and the remaining part of the neglected error as an uncontrollable factor accompanies the data into the feature extraction stage, and their impact on the algorithmic model is controlled by the robustness of the feature engineering scheme.

1) Gyroscope

The relationship between the actual output and the true angular rate of a single-axis MEMS gyroscope is shown in equation (2):

$$R_t = SF_{gyro} \times \left(R_m - R_0\right) \tag{2}$$

Where R_t - the ideal output, i.e. the true angular rate value, in °/S (dps);

R_m - indicates the actual digital signal value measured by the sensor in lowest significant bits (LSBs);

R_0 - the measured value of the sensor at zero input level, i.e., the zero bias value, in (dps);

SF_{gyro} - the sensitivity, i.e. the ratio between the gyroscope output and the input angular rate, is the conversion factor used for digital-to-analog conversion in (dps/LSBs).

The inertial motion acquisition device was then subjected to a static experiment and the following zero bias data for the MEMS sensor was obtained:

$$\begin{aligned} R_{0x} &= 0.000\quad 089\quad 792\quad 345 \\ R_{0y} &= -0.000\quad 054\quad 875\quad 483 \\ R_{0z} &= 0.000\quad 005\quad 848\quad 978 \end{aligned} \tag{3}$$

Thus, the calibration equation is obtained as shown in equation (4), where checking the table shows that $SF_{gyro} = 2000(dps / LSBs)$. i.e.:

$$\begin{aligned} R_x &= SF_{gyro} \times \left(R_{output-x} - R_{0x} \right) \\ R_y &= SF_{gyro} \times \left(R_{output-y} - R_{0x} \right) \\ R_z &= SF_{gyro} \times \left(R_{output-z} - R_{0x} \right) \end{aligned} \tag{4}$$

2) Accelerometers

Due to the nature of the measurement principle, the accelerometer has a reference calibration value, i.e., the gravitational acceleration value of the earth, and can therefore be calibrated using a simple static six-position calibration method. First the error equation of the accelerometer is shown in equation (5):

$$\begin{aligned} \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} &= \begin{bmatrix} A_{x0} \\ A_{y0} \\ A_{z0} \end{bmatrix} + \begin{bmatrix} SF_x & K_{\alpha x1} & K_{\alpha x2} \\ K_{\alpha y1} & SF_y & K_{\alpha y2} \\ K_{\alpha z1} & K_{\alpha z1} & SF_z \end{bmatrix} \cdot \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \\ &+ \begin{bmatrix} K_{xx} & 0 & 0 \\ 0 & K_{yy} & 0 \\ 0 & 0 & K_{zz} \end{bmatrix} \cdot \begin{bmatrix} a_x^2 \\ a_y^2 \\ a_z^2 \end{bmatrix} \end{aligned} \tag{5}$$

where $\vec{A}\left(A_x, A_y, A_z \right)$ - the actual output value of the MEMS accelerometer;

$\overrightarrow{A_0}\left(A_{x0}, A_{y0}, A_{z0} \right)$ - the zero bias component of the sensor in g ;

$\vec{a}\left(a_x, a_y, a_z \right)$ - the ideal output, i.e., the true acceleration component, whose modulus magnitude is constant to the g value, i.e.:

$$a_x^2 + a_y^2 + a_z^2 = g^2 \tag{6}$$

$SF_{acc}\left(SF_x, SF_y, SF_z \right)$ - scale factor of the accelerometer in g^{-1} ;

$\left(K_{\alpha x1}, K_{\alpha x2}, K_{\alpha y1}, K_{\alpha y2}, K_{\alpha z1}, K_{\alpha z2} \right)$ - Installation error factor;

The third term on the right side of the equation represents the quadratic term error, and the simplified model after omitting this part is shown in equation (7):

$$\vec{A} = \overrightarrow{A_0} + T_{3\times3} \cdot \vec{a} \tag{7}$$

It is known that the output of the accelerometer is $g = 9.81 m / s^2$ when the accelerometer is static on the horizontal plane in the positive vertical axis (facing up), the device is fixed on a stable horizontal surface, respectively, so that its carrier coordinate system of x , y , z three axes along the vertical upward and downward direction for a certain period of time, after the stabilization of the output can be obtained respectively in six states of the output sequence, take the statistical average as a representative of the data, in order to The statistical average is taken as the data representative, which is combined in the form of row vectors as a matrix of 6×3 size, and the real acceleration component can be obtained as shown in Eq. (8):

$$\vec{A}_{1\to6} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \\ a_{51} & a_{52} & a_{53} \\ a_{61} & a_{62} & a_{63} \end{bmatrix} \quad \vec{a}_{1\to6} = \begin{bmatrix} +g & 0 & 0 \\ -g & 0 & 0 \\ 0 & +g & 0 \\ 0 & -g & 0 \\ 0 & 0 & +g \\ 0 & 0 & -g \end{bmatrix} \tag{8}$$

First as shown in equation (9), the zero bias error is calculated as follows:

$$A_{0x} = \frac{A_{11} + A_{21}}{2}, A_{0y} = \frac{A_{32} + A_{42}}{2}, A_{0z} = \frac{A_{53} + A_{63}}{2} \tag{9}$$

Taking the 1st, 3rd and 5th rows from the matrices $\vec{A}_{1\to 6}$ and $\vec{a}_{1\to 6}$ to form $\vec{A}_{1,3,5}$, $\vec{a}_{1,3,5}$, respectively, we can obtain an equation containing 9 unknowns as shown in Eq. (10):

$$\begin{bmatrix} a_{11}-A_{0x} & a_{12}-A_{0y} & a_{13}-A_{0z} \\ a_{31}-A_{0x} & a_{32}-A_{0y} & a_{33}-A_{0z} \\ a_{51}-A_{0x} & a_{52}-A_{0y} & a_{53}-A_{0z} \end{bmatrix} \times T_{3\times 3} = \begin{bmatrix} g & 0 & 0 \\ 0 & g & 0 \\ 0 & 0 & g \end{bmatrix} \tag{10}$$

It is sufficient to solve the matrix $T_{3\times 3}$ by using the solution method of the system of linear equations: $T_{3\times 3} = \left(\vec{A}_{1,3,5}\right)^{-1} \times \vec{a}_{1,3,5}$.

So the six-plane calibration experiment is carried out to obtain the steady state output sequence of the triaxial accelerometer on the horizontal plane, and the sequence is randomly sampled, 100 data points are taken to find the mean value, and the final six-position calibration data is obtained.

The calibration parameters can be obtained by bringing the data into Eq. (9), where the accelerometer zero bias is:

$$\begin{aligned} A_{x0} &= -0.000\,892\,388 \\ A_{y0} &= 0.000\,649\,876 \\ A_{z0} &= 0.001\,084\,710 \end{aligned} \tag{11}$$

The installation error factor is:

$$\begin{aligned} K_{\alpha x1} &= -0.007\,815\,476, K_{\alpha x2} = 0.008\,373\,928 \\ K_{\alpha y1} &= 0.000\,725\,927, K_{\alpha y2} = -0.000\,176\,984 \\ K_{\alpha z1} &= -0.000\,421\,032, K_{\alpha z2} = -0.001122\,914 \end{aligned} \tag{12}$$

Substituting the above values into Eq. (12), the true error calibration formula of the triaxial accelerometer under the experimental field conditions can be obtained. In order to check the effect of this calibration model, the difference between the modal value of the output value of the triaxial MEMS accelerometer and the standard g-value is used as the error indicator, and the relationship shown in Eq. (13) exists in the stationary state:

$$error = \left\|\vec{A}\right\|_2 - g = \sqrt{A_x^2 + A_y^2 + A_z^2} - g \tag{13}$$

3) Magnetometer

Magnetometers can convert the magnetic induction strength of the magnetic field into a digital signal for output, however, due to the complexity of the ground's ambient magnetic field, the interference factors are numerous and high randomness, and even the printed circuit board of the integrated sensors itself will exist to a certain extent can not be suppressed by the phenomenon of electromagnetic interference, the magnetometer components also inevitably contain some ferromagnetic material which will produce remanent magnetism.

Therefore, compared with other types of inertial sensors, MEMS magnetometer circuit structure caused by the drift phenomenon tends to be more serious, if the magnetometer's measurement error curve as described in the introduction to this section of a smooth random process, then the circuit drift and remanent magnetization phenomenon on the measurement element performance can be summarized as a zero-bias error model. That is, the relationship between the actual output of the magnetometer and the true magnetic field magnitude component is shown in equation (14):

$$\begin{aligned} B_y &= B_{y0} + b_y \\ B_z &= B_{z0} + b_z \\ B_x &= B_{x0} + b_x \end{aligned} \tag{14}$$

where $\vec{B}\left(B_x, B_y, B_z\right)$ - the actual output value of the MEMS magnetometer;

$\vec{b}\left(b_x, b_y, b_z\right)$ -Magnitude component of the magnetic induction strength of the real magnetic field;

$\overrightarrow{B_0}\left(B_{x0}, B_{y0}, B_{z0}\right)$ - the zero bias component of the sensor.

II. B.Data normalization

The study data contains acquisition data from different sensors, each of which does not correspond to the same units, and under the influence of magnitude, gyroscope (in °/s) data will be literally much larger than accelerometer (in m/s^2) data for the same level of motion amplitude. Since some of the feature extraction methods described later are sensitive to vector modulus, there will be a difference in the contribution of data at different orders of magnitude to the feature expression, i.e., it is manifested that the gyroscope data is more important in motion recognition, which obviously violates the basic knowledge, and thus normalization is needed to convert the inertial data into a purely dimensionless expression of magnitude.

In order to facilitate the calculation, the normalization method used in this section is block mapping: the data of the three sensors, namely gyroscope, accelerometer and magnetometer, are mapped into the range of $(0,1)$, and since they are all three-axis data, the specific expression formulas are shown in Eq. (15):

$$a_{norm} = \frac{a - \min\left(A_x, A_y, A_z\right)}{\max\left(A_x, A_y, A_z\right) - \min\left(A_x, A_y, A_z\right)} \tag{15}$$

III. Posture Recognition Model for Rehabilitation of Lower Extremity Motor Function Injuries in Sports

In the previous chapter, this paper completed the preprocessing of sports lower limb movement action data. In this chapter, the convolutional neural network (CNN) and long short-term memory (LSTM) recurrent neural network will be combined into DeepConvLSTM to propose a posture recognition model based on DeepConvLSTM for rehabilitation of sports lower limb motor function injury.

III. A. CNN Convolutional Neural Network

CNNs have been successfully applied to the field of human motion recognition for wearable sensors [27]. CNNs perform convolutional operations on sensor signals by means of a convolutional kernel and extract features from them, followed by using the extracted features as inputs to a multilayered, fully-connected layer in order to obtain probability distributions of different human motions. The ability of the convolutional kernel to capture temporal signal structure in its kernel window is demonstrated through experiments on several sensor motion public datasets. The experimental results show that the recognition performance of the CNN outperforms traditional machine learning methods based on manual feature extraction.

III. B. LSTM recurrent neural network

LSTM regulates the interaction between the storage unit itself and its environment by introducing the idea of thresholds [28]. The three thresholds of LSTM are forgetting gate, input gate and output gate. The forgetting gate can determine the extent to which existing cell states are forgotten, the input gate can determine the extent to which candidate cell states are added to the original cell states, and the output gate can control the extent to which cell states are output. In the absence of external interference, the cell state of the LSTM can be maintained from one time step to another with a constant time step, thus simplifying its learning of temporal relationships on longer time scales.

III. C. DeepConvLSTM neural network

Since the existing single neural network algorithms are often limited by the computational speed and accuracy when applied to complex problems, many people have focused on deep learning algorithms that combine multiple neural network structures. Among them, the network model based on the combination of convolutional neural network and LSTM recurrent neural network has received more attention in recent years. This combination network has already achieved better results in robot tactile recognition, speech recognition and other fields.

Deep Conv LSTM consists of a series of convolutional and recurrent layers, where the convolutional layer acts as a feature extractor and is responsible for converting the input sensor data into an abstract feature map. The recurrent layer, on the other hand, models the temporal dependencies in the input feature map data.

III. D. DeepConvGRU Neural Network Modeling

The network structure proposed in this paper has 9 layers, including 3 convolutional layers, 2 pooling layers, 2 looping layers and 1 output layer. Firstly, the preprocessed sensor data will be used as the input of the convolutional layer, then the feature output extracted from the convolutional pooling layer will be used as the input of the recurrent layer, and finally the output of the recurrent layer will be sent to the output layer, which is a fully connected layer with the activation function of Softmax. The Softmax layer can calculate the probability of the samples belonging to

each class, and at the same time, it is convenient for the later computation of the loss function by solving the cross entropy. Through the above description of the network structure, the final structure of the DeepConvGRU network is constructed as shown in Figure 1.

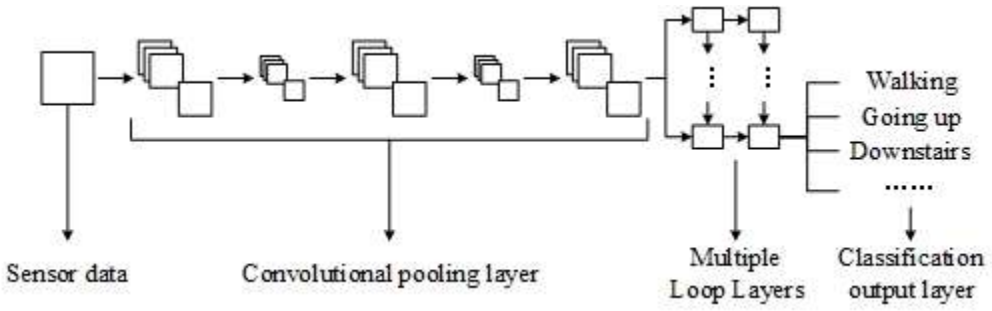


Figure 1: DeepConvGRU network structure

In order to show the advantages of Deep Conv GRU, the performance of Deep Conv GRU is compared with that of general CNN without loop layer and Deep Conv LSTM with LSTM loop unit in the loop layer. In order to fairly compare the three models, the convolutional pooling layer part of the experiment adopts the same structure, and the same size and number of convolutional kernels are selected. The main difference between the three networks lies in the recurrent layer part, where Deep Conv GRU chooses GRU recurrent units, Deep Conv LSTM chooses LSTM recurrent units, and the CNN replaces the recurrent layer all the time with a fully connected layers [29]. Thus, the difference in the final performance of the models depends entirely on the strengths and weaknesses of the models themselves, and not due to better optimization or preprocessing, for example.

III. D. 1) Convolutional pooling layer

The convolutional pooling layer part of this network consists of alternating stacks of convolutional and pooling layers. It includes 3 convolutional layers and 2 pooling layers. The number of convolution kernels of the 3 convolutional layers are 16, 32 and 64 respectively, the size of convolution kernels are 1×8, the step size is 1 and the padding mode is complementary 0. In the convolutional layer, each time, multiple convolution kernels can be used to perform convolution operation on the input information to generate multiple feature maps, and its mathematical model can be described as:

$$x_j^{l+1} = f\left(\sum_{i=1}^{M^i} x_i^l \times k_i^l j + b_j^l\right) \tag{16}$$

where: x_j^{l+1} is the j-feature map of the l-layer; f is a nonlinear activation function; M^l is the number of feature maps in the l-layer; k_{ij}^l is the convolution kernel that maps the i-feature map of the l-layer to the j-feature map of the $(l+1)$-layer by convolution operation. The nonlinear activation usually chooses the modified linear unit (ReLU), the advantage of ReLU is that it can reduce the dependency between the parameters and reduce the generation of overfitting problems.The formulaic expression of ReLU is as follows:

$$a_i^{l+1}(j) = f\left(y_i^{l+1}(j)\right) = \max\left\{0, y_i^{(l+1)}(j)\right\} \tag{17}$$

where: $y_i^{(l+1)}(j)$ denotes the output value of the convolution operation; $a_i^{l+1}(j)$ is the activation value of $y_i^{(l+1)}(j)$.

The 2 pooling layers are distributed between the 3 convolutional layers, and the pooling filters all have a size of 2 × 1, a step size of 2, and a padding method of complementary 0. The pooling layers are sandwiched in between consecutive convolutional layers, and are used to compress the amount of data and parameters, and to reduce overfitting. The commonly used pooling methods are divided into maximum pooling and average pooling, average pooling can take the average value of all feature points in the window, and maximum pooling takes the largest feature point in the window. In order to extract the most obvious features in the feature map, this network structure uses the maximum pooling strategy.

III. D. 2) GRU Circulation Layer

The 2 recurrent layers of the network both use GRU as the recurrent unit with 128 recurrent neurons.GRU is a much altered variant of LSTM, first proposed by Cho et al. Similar to the structure of LSTM, GRU has different gates in its structure, but it combines the forget gate and the input gate into one into an update gate, and also makes some other changes, such as merging the cell state and the hidden state.

Assuming that the hidden state (i.e., the output) of the GRU at time t is h_t, it can be represented by the hidden state h_{t-1} at the previous moment and the candidate hidden state $\tilde{h}_t$:

$$h_t = (1 - z_t) h_{t-1} + z_t \tilde{h}_t \tag{18}$$

The update gate z_t determines the extent of the hidden state update, which is computed by equation (19).

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{19}$$

This computation of solving the linear sum of old hidden states and candidate hidden states is very similar to LSTM, but GRU does not have any way to control the extent of its cell state output, meaning that it outputs the complete cell state at every step.

The exact formula for calculating the candidate hidden states is as follows:

$$\tilde{h}_t = \tanh(W x_t + U(r_t \square h_{t-1})) \tag{20}$$

where: r_t is a set of reset gates; $\square$ is a dot-multiplication (i.e., vector elements correspond to multiplication) symbol. When the reset gates are closed (i.e., when r_t approaches 0), the reset gates cause the cell to forget its previous state.

The reset gate r_t is computed similarly to the update gate:

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{21}$$

IV. Sports lower limb motor function injury rehabilitation posture recognition experiment

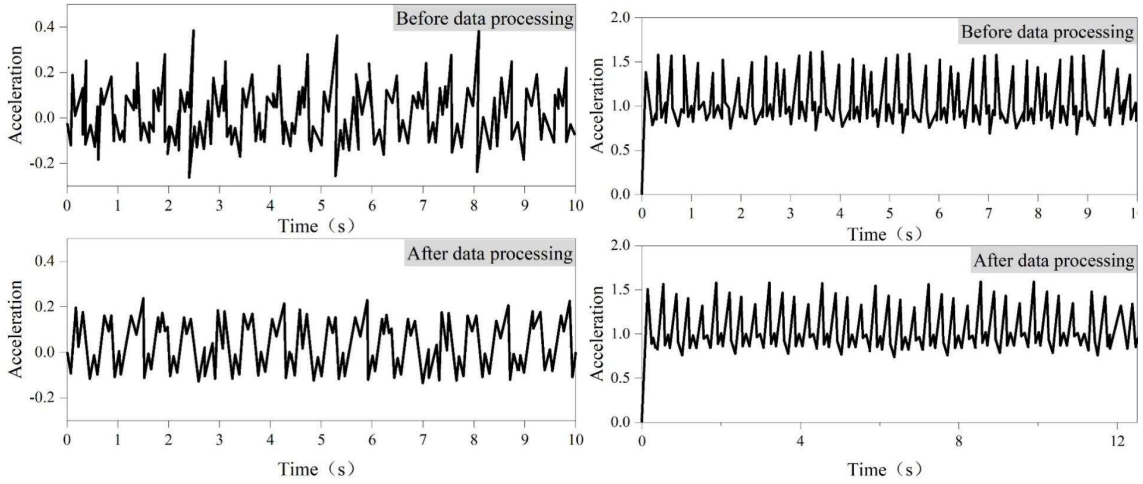
In this chapter, the performance of DeepConvGRU, the posture recognition model proposed in this paper for sports lower limb motor function injury rehabilitation, on the task of lower limb motor function injury rehabilitation will be evaluated through experiments, which firstly process the collected sports lower limb motor action data. Then the model performance is validated by comparison experiments. The comparison models used in the model comparison experiment include the time interval series construction model LSTM, the warp volume network CNN, the residual volume network ResNet, the iSPL Inception model and the system depth perception network Rehab-Net for fitness equipment monitoring.

IV. A. Pre-processing of sports lower limb movement action data

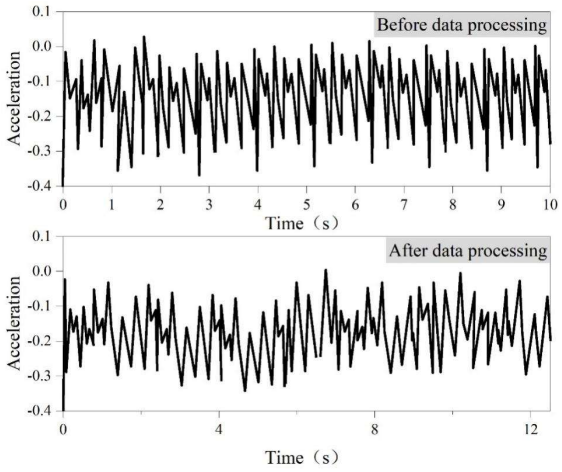
In the process of human body data acquisition, it is not possible to get the data of the target movement directly even if the human movement is standardized and sophisticated acquisition equipment is used. In this section, the preprocessing method of sports lower limb movement action data proposed in this paper will be used to suppress the noise of the data so as to make it closer to the real data. The data used in this experiment comes from the homemade lower limb rehabilitation posture dataset RehaLab-412 and the public posture dataset UCISmartphone.

IV. A. 1) Data noise suppression

Noise will inevitably be generated due to the abnormal jitter during movement and the influence of the measurement environment. The frequency of human movement is generally concentrated in 0-20Hz, and the change is more obvious in 0-5Hz, while the frequency of noise is certainly greater than this frequency. Randomly selected a group of lower limb movement data as data preprocessing sample data, a total of 10s, the acquisition frequency is set to 50Hz, a total of 500 groups of data. Using the data preprocessing method in this paper for noise processing, the denoising results are specifically shown in Figure 2. Figures (a)~(b) are the comparison graphs of the time domain before and after the x, y, z-axis acceleration filtering, respectively. As can be seen from the figure, some of the more prominent cusps of the processed data curve are effectively removed, and the whole becomes much smoother, achieving the effect of noise suppression.



(a)x (b)y



(c)z

Figure 2: Comparison of time domain before and after acceleration filtering

IV. A. 2) Data calibration process

Place the data acquisition module flat on a flat desktop, and use the data acquisition module to collect three-axis acceleration and three-axis angular velocity data in the stationary state. The three-axis angular velocity data should be zero in the stationary state, the x- and y-axis acceleration should be zero, and the z-axis acceleration should be +1g or -1g. However, the data measured by the sensor will drift during the actual measurement process, which needs to be compensated for the measured data. The calibration formula of the sensor is shown in equation (22), which needs to calculate the scale factor ruler and error $error$, limited by the experimental conditions, the z-axis acceleration data are accurately calibrated, and the other 5-axis data are fuzzy calibrated, and the fuzzy calibration doesn't calculate the scale factor, and the mean value is considered as the zero-bias error. When the sensor is placed in the forward direction, the z-axis acceleration is +1g, and when the sensor is placed in the reverse direction, the z-axis acceleration is -1g. Combined with the z-axis acceleration data measured by the sensor, the scale factor K and the error $error$ can be calculated:

$$G=K\times ADC+error \tag{22}$$

where: G is the actual output value; ADC is the theoretical output value; K is the scale factor; $error$ is the zero bias error. The scale factor and zero bias error are calculated for the sports lower limb movement action data, and the lower limb movement action data are obtained as shown in Table 1. In the table, it can be seen that the standard deviation of the data is stable on the order of 10-3, the error is stable on the order of 10-2, the acceleration measurement is stable on the order of 0.001g, and the standard deviation of the angular velocity data is stable on

the order of 10-1, so the zero-bias error has a certain degree of interference with the measured data, which is compensated for by the actual measurement and corrected for the error.

Table 1: Lower limb movement data

-	Gyro x(°/s)	Gyro y(°/s)	Gyro z(°/s)	Accel x(g)	Accel y(g)
standard deviation	0.0362	0.121	0.0677	0.0018	0.0025
Mean value	-0.0201	1.0875	-0.4192	0.0186	0.0158
Error	0.0201	-1.0875	0.4192	-0.0186	-0.0158

IV. B. Analysis of the results of the comparison experiment

The model performance is affected by many factors, in addition to its own structural design factors, it is also affected by the size of the data segmentation window, loss function, optimization algorithms and so on. In this paper, the hyperparameters of the proposed model are optimized for the lower limb rehabilitation posture recognition task to achieve the best recognition results. Finally, the performance of the model is verified by comparison experiments.

IV. B. 1) Hyperparameter optimization

The execution time of different lower limb postures is inconsistent, resulting in different sizes of the time span of the complete data, in order to explore the optimal time window for data segmentation, the sensor multivariate sequence data streams were segmented with window sizes of 300, 400 and 600, respectively, with an importance of 50%, to explore the impact of sample dimensionality on the recognition performance, and to find the optimal segmentation window. At the same time, different hyperparameter combinations are also experimented, based on the lower limb rehabilitation posture dataset for training and testing, to explore the optimal combination of hyperparameter settings with the model recognition accuracy as the evaluation index, and the experimental results are shown in Table 2. The experimental results show that the activation function will have an impact on the recognition results, compared with ERelu as the activation function model overall accuracy of 96.94%, the replacement of the Relu activation function can improve the accuracy of the model by nearly two percentage points. In the experiments with the optimization algorithm, the Adam optimization algorithm achieved optimal results, and Relu in terms of the activation function showed stronger applicability to the task. Segmentation window size has the most obvious effect on the task of posture recognition for rehabilitation of lower limb motor function injury in sports, the data with window size 300 segmentation only achieved 84.35% accuracy, with the expansion of the segmentation window, the accuracy was subsequently increased to 99.17%, the completeness of the data samples is a key step in achieving a high recognition rate.

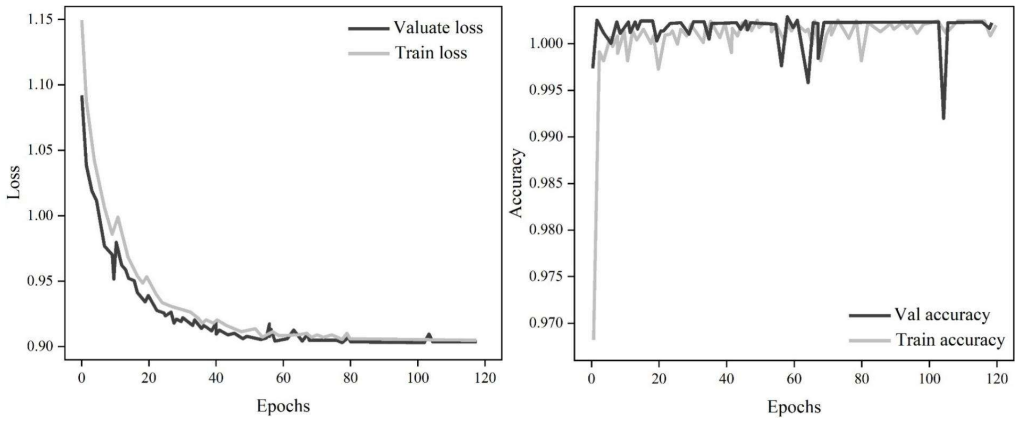
Table 2: Model performance test based on different parameter settings

Hyperparameter term	Value	Accuracy
Activation function	Erelu	96.94%
	LeakuRelu	97.31%
	Relu	99.17%
Optimization algorithm	SGD	95.03%
	Adamax	95.61%
	Adam	99.13
Split window	300	84.35%
	400	93.11%
	600	99.17%

IV. B. 2) Comprehensive model performance validation

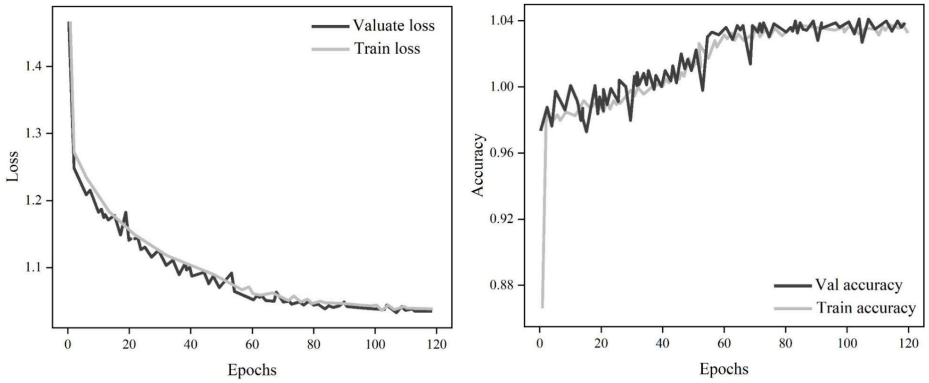
The model comprehensive performance testing experiments are based on the self-made lower limb rehabilitation posture dataset RehaLab-412 and open posture dataset UCISmartphone respectively. The training process of the model in this paper on RehaLab-41 and UCISmartphone datasets is specifically shown in Figure 3. Figures (a) and (b) show the loss values and accuracy on RehaLab-41 dataset, while Figures (c) and (d) show the loss values and accuracy on UCISmartphone dataset. From the figures, it can be seen that the model in this paper completes convergence and model training within the set number of iterations. The training loss values of the model in the first

30% of epochs are decreasing steadily and gradually converge to 0. In the accuracy curve, the fast learning ability of the model can be seen. After nearly 10 epochs of training, the model achieves more than 90% validation recognition accuracy on both tasks, and the training and validation accuracies of the model increase steadily and slowly with the increase in the number of iterations, and finally converge stably.



(a)Loss value of the model on the RehaLab-41 dataset

(b)Accuracy of the model on the RehaLab-41 dataset



(c)Loss value of the model on the UCISmartphone dataset

(d)Accuracy of the model on the UCISmartphone dataset

Figure 3: Visualization of training process

The performance of the different models on the pose recognition task for lower limb functional impairment rehabilitation is specifically shown in Table 3.Both the LSTM model and the bidirectional LSTM model showed poor performance on the pose recognition task, obtaining only 65.62% and 69.64% recognition accuracies. Compared to the LSTM family of models, the convolutional network demonstrated superior results in the lower limb rehabilitation posture recognition task, in which the base 1DCNN network achieved a combined recognition accuracy of 87.95%. The Rehab-Net model deep residual network 1DResNet achieves a combined recognition accuracy of 91.52%, but its relatively large number of network parameters consumes resources. The iSPL Inception model achieves 95.98% on the lower limb rehabilitation posture recognition task and also achieves more accurate recognition on overt posture and on. DeepConvGRU, the sports lower limb motor function injury rehabilitation posture recognition model proposed in this paper, achieves 99.11% accuracy on the RehaLab-412 lower limb rehabilitation posture task with the smallest model parameter scale, and it can be seen that the model exhibits a more balanced performance on this task based on the F1 score. On the UCISmartphone dataset, the DeepConvGRU model in this paper also achieves an accuracy of 96.83%, demonstrating good model performance.

Table 3: Comparative experiment of lower limb rehabilitation posture recognition task

Model	Data set	Iteration times	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
LSTM	RehaLab-412	128	65.62	65.93	67.2	63.77
	UCISmartphone		89.98	89.9	89.96	89.86
Bi-LSTM	RehaLab-412	128	69.64	74.43	71.99	71.9
	UCISmartphone		89.74	89.75	89.77	89.73
1DCNN	RehaLab-412	128	87.95	91.68	87.73	85.98
	UCISmartphone		89.52	89.31	88.64	89.31
1DResNet	RehaLab-412	128	91.52	92.54	91	90.83
	UCISmartphone		91.88	92.1	91.86	91.69
Rehab-Net	RehaLab-412	128	91.52	91.47	91.64	91.47
	UCISmartphone		90.66	90.7	90.77	90.72
ISPL Inception	RehaLab-412	128	95.98	96.47	95.71	95.66
	UCISmartphone		95.09	95.9	95.72	95.67
DeepConvLSTM	RehaLab-412	128	99.11	99.23	99.09	99.16
	UCISmartphone		96.83	95.23	95.23	95.2

V. Conclusion

In this paper, the data preprocessing method based on error calibration and data normalization is used to enhance the quality of sports lower limb movement action data. After completing the data preprocessing work, a posture recognition model based on DeepConvGRU neural network is proposed for the rehabilitation of sports lower limb motor function injury. The gesture recognition experiment of sports lower limb motor function injury rehabilitation is carried out to test the recognition performance of the DeepConvGRU model of this paper in the task of lower limb motor function injury rehabilitation. The processed data curve is overall smoother and effectively achieves the effect of noise suppression, the standard deviation and error of the data are stabilized on the order of 10-3 and 10-2, the acceleration measurements are stabilized at 0.001g, and the standard deviation of the angular velocity data is stabilized on the order of 10-1, and the error is effectively corrected. In the comparison experiments, the Adam optimization algorithm in hyper-parameter optimization achieves the optimal effect, and with the expansion of the segmentation window, the accuracy is subsequently improved to 99.17%. The DeepConvGRU model in this paper has a steady decrease in the training loss value and gradually tends to 0 within the first 30% epochs on the RehaLab-41 and UCISmartphone datasets, and achieves more than 90% validation recognition accuracy in more than 2 tasks. Meanwhile, the DeepConvGRU model in this paper achieves 99.11% and 96.83% accuracy on RehaLab-41 and UCISmartphone datasets, respectively, demonstrating good model performance.

Funding

2022 Hunan Provincial Vocational College Education and Teaching Reform Research Project, Hunan Communication Polytechnic,Project No.: ZJGB2022236, "Research on the Construction of a 'Two Characteristics and One Degree' Golden Course System for Health Qigong in Colleges under the Digital China Initiative".

References

[1] Larsen, L. R., Kristensen, P. L., Junge, T., Møller, S. F., Juul-Kristensen, B., & Wedderkopp, N. (2016). Motor performance as risk factor for lower extremity injuries in children. Medicine & Science in Sports & Exercise, 48(6), 1136-1143.

[2] Nanbancha, A., Mawhinney, C., & Sinsurin, K. (2023). The effect of motor imagery and action observation in the rehabilitation of lower limb injuries: A scoping review. Clinical Rehabilitation, 37(2), 145-161.

[3] Brumitt, J., Heiderscheit, B. C., Manske, R. C., Niemuth, P., Mattocks, A., & Rauh, M. J. (2016). The lower-extremity functional test and lower-quadrant injury in NCAA division III athletes: a descriptive and epidemiologic report. Journal of sport rehabilitation, 25(3), 219-226.

[4] Vereijken, A., Aerts, I., Jetten, J., Tassignon, B., Verschueren, J., Meeusen, R., & van Trijffel, E. (2020). Association between functional performance and return to performance in high-impact sports after lower extremity injury: a systematic review. Journal of sports science & medicine, 19(3), 564.

[5] Rejc, E., & Angeli, C. A. (2019). Spinal cord epidural stimulation for lower limb motor function recovery in individuals with motor complete spinal cord injury. Physical Medicine and Rehabilitation Clinics, 30(2), 337-354.

[6] McKeon, P. O., & Fourchet, F. (2015). Freeing the foot: integrating the foot core system into rehabilitation for lower extremity injuries. Clinics in sports medicine, 34(2), 347-361.

[7] Tsvyakh, A. I., & Hospodarskyy, A. J. (2017). Telerehabilitation of patients with injuries of the lower extremities. Telemedicine and e-Health,

23(12), 1011-1015.
[8] Palermi, S., Massa, B., Vecchiato, M., Mazza, F., De Blasiis, P., Romano, A. M., ... & Sirico, F. (2021). Indirect structural muscle injuries of lower limb: Rehabilitation and therapeutic exercise. Journal of Functional Morphology and Kinesiology, 6(3), 75.
[9] Madsen, A., Sharififar, S., Oberhaus, J., Vincent, K. R., & Vincent, H. K. (2022). Anxiety state impact on recovery of runners with lower extremity injuries. PLoS One, 17(12), e0278444.
[10] Taylor, C. R., & Knobloch, A. C. (2018). Lower-extremity rehabilitation. Current Sports Medicine Reports, 17(12), 405-406.
[11] Marusic, J., Dolenc, P., & Sarabon, N. (2020). Psychological aspect of rehabilitation and return to sport following lower limb injuries. Montenegrin Journal of Sports Science and Medicine, 9(2), 59.
[12] Zhu, Z., & Ma, Y. (2021). Sports injury and rehabilitation of lower limb soft tissue. Revista Brasileira de Medicina do Esporte, 27, 804-806.
[13] DeJong, A. F., & Hertel, J. (2018). Gait-training devices in the treatment of lower extremity injuries in sports medicine: current status and future prospects. Expert Review of Medical Devices, 15(12), 891-909.
[14] Bulat, M., Can, N. K., Arslan, Y. Z., & Herzog, W. (2019). Musculoskeletal simulation tools for understanding mechanisms of lower-limb sports injuries. Current Sports Medicine Reports, 18(6), 210-216.
[15] Kahanov, L., Eberman, L. E., Games, K. E., & Wasik, M. (2015). Diagnosis, treatment, and rehabilitation of stress fractures in the lower extremity in runners. Open access journal of sports medicine, 87-95.
[16] Joyce, D., & Lewindon, D. (2016). Sports injury prevention and rehabilitation. integrating medicine and science for performance solutions, 2016.
[17] Halverson, D., & Hirth, C. J. (2024). Rehabilitation of lower leg injuries. In Rehabilitation Techniques for Sports Medicine and Athletic Training (pp. 697-730). Routledge.
[18] Wilke, C., Pfeiffer, L., & Froböse, I. (2017). Return to sports after lower extremity injuries: Assessment of movement quality. Health, 9(10), 1416.
[19] Silfies, S. P., Ebaugh, D., Pontillo, M., & Butowicz, C. M. (2015). Critical review of the impact of core stability on upper extremity athletic injury and performance. Brazilian journal of physical therapy, 19, 360-368.
[20] Gokeler, A., McKeon, P. O., & Hoch, M. C. (2020). Shaping the functional task environment in sports injury rehabilitation: a framework to integrate perceptual-cognitive training in rehabilitation. Athletic Training & Sports Health Care, 12(6), 283-292.
[21] Comfort, P., Jones, P. A., Smith, L. C., & Herrington, L. (2015). Joint kinetics and kinematics during common lower limb rehabilitation exercises. Journal of athletic training, 50(10), 1011-1018.
[22] Nas, K., Yazmalar, L., Şah, V., Aydın, A., & Öneş, K. (2015). Rehabilitation of spinal cord injuries. World journal of orthopedics, 6(1), 8.
[23] Jeong, L., & Li, D. (2024). Psychological well-being from sports injuries in adolescence: a narrative review. Cureus, 16(7).
[24] Gledhill, A., Forsdyke, D., & Murray, E. (2018). Psychological interventions used to reduce sports injuries: a systematic review of real-world effectiveness. British Journal of Sports Medicine, 52(15), 967-971.
[25] Gennarelli, S. M., Brown, S. M., & Mulcahey, M. K. (2020). Psychosocial interventions help facilitate recovery following musculoskeletal sports injuries: a systematic review. The Physician and Sportsmedicine, 48(4), 370-377.
[26] Truong, L. K., Mosewich, A. D., Holt, C. J., Le, C. Y., Miciak, M., & Whittaker, J. L. (2020). Psychological, social and contextual factors across recovery stages following a sport-related knee injury: a scoping review. British journal of sports medicine, 54(19), 1149-1156.
[27] Parul Sharma & Pawanesh Abrol. (2025). Multi-component image analysis for citrus disease detection using convolutional neural networks. Crop Protection,193,107181-107181.
[28] Lijian Wei,Sihang Chen,Junqin Lin & Lei Shi. (2025). Enhancing return forecasting using LSTM with agent-based synthetic data. Decision Support Systems,193,114452-114452.
[29] Maciej Ławryńczuk & Krzysztof Zarzycki. (2025). LSTM and GRU type recurrent neural networks in model predictive control: A Review. Neurocomputing,632,129712-129712.