ISSN 1330-3651 (Print), ISSN1848-6339(Online)

https://doi.org/10.17559/TV-20240524001659
Original scientific paper

Enhancing User Experience with Human-Computer Interaction Technology: A Digital Terminal Application

Jian DONG

Abstract: The objective is to develop an accurate method for capturing and identifying complex gestures, with the aim of enhancing the user experience of digital media multimedia terminals. A dynamic gesture recognition system based on Leap Motion sensor and bidirectional long and short term memory network is designed. Gesture recognition, classification and control are realized by collecting and analyzing gesture data and combining with deep learning algorithm. The experimental results show that the recognition accuracy of model training set reaches 99%, test set reaches 97%, and the error converges to less than 0.2 in training set and less than 0.01 in test set. The model also shows high accuracy and low error in the craft language database. The dynamic gesture recognition system designed in this study provides users with more intelligent and convenient digital media multimedia terminal experience, and has broad application prospects in the dynamic gesture recognition multimedia terminal.

Keywords: Bi-LSTM; digital multimedia terminal; dynamic gesture recognition; human-computer interaction technology; leap motion sensor

1 INTRODUCTION

With the rapid development of computer science, artificial intelligence, sensor technology, and display technology, human-computer interaction (HCI) technology continues to innovate, providing new possibilities for the functional design of digital media and multimedia terminals. However, as a key component of HCI, dynamic gesture recognition (GDR) faces the challenge of accurately capturing and recognizing complex gesture actions [1]. These gestures are not only diverse, but also varied, making the recognition accurate and real-time become the core issues to improve user experience. The importance of GDR is not only reflected in its contribution to enhancing the naturalness and intuitiveness of multimedia terminal interaction, but also, with the popularization of intelligent devices and the advancement of globalization, it is of great significance for cross-cultural communication and barrier-free access to information of special groups [2, 3]. Although some progress has been made in existing research, most methods still have limitations in dealing with complex dynamic gestures, environmental noise interference, and light changes, and lack adaptability to user habits in different cultural backgrounds. Most of the existing studies focus on gesture recognition in a single data set or a specific environment, but lack extensive verification of diverse gestures and practical application scenarios. In addition, the existing methods still need to be improved in terms of real-time, accuracy and robustness to environmental changes [4, 5]. To overcome these shortcomings, a GDR system based on Leap Motion sensor and bidirectional long and short term memory (Bi-LSTM) network is designed. The system can not only provide high accurate gesture recognition, but also adapt to different operating environments and user habits. The contribution of the research is to propose an innovative system integrating Leap Motion sensor and Bi-LSTM network, which can not only realize the accurate capture and recognition of gestures, but also convert the recognition results into corresponding instructions to control the functions of multimedia terminals. The recognition accuracy, error convergence, and evaluation index of the system are superior to the existing technology, which provides a new solution for the field of GDR. The research not only provides powerful technical support for the intelligent interaction of digital media multimedia terminals, but also has important theoretical and practical significance for promoting the development of HCI technology and meeting the needs of users for efficient and personalized services.

2 RELATED WORKS

In the design of digital multimedia terminal functions, the application of HCI technology aims to enhance user experience, facilitate user operations, and achieve effective interaction between users and terminals. An increasing amount of research on HCI has appeared in recent years. For example, Ge et al. proposed surface electromyography (sEMG) signal gesture recognizing combining deep learning for master-slave operation scenarios. Experimental results showed that their research achieved the highest accuracy [6]. Alashhab et al. developed an interaction system for mobile devices controlled by gestures, specifically designed for visually impaired individuals to switch between applications and perform various operations. Comparative analysis with nearly 50 current ways demonstrated the competitive performance of the system in executing different operations [7]. Wu et al. addressed the issue of poor spatiotemporal information fusion in existing methods and proposed an attention mechanism for traffic police gesture recognition based on an improved spatiotemporal convolutional neural network. Experiment outcomes denoted that this method beat several others on the AVA and Chinese traffic police gesture datasets [8]. Zhang et al. introduced a new deep learning network for gesture recognition, which demonstrated improved robustness compared to other models by enhancing gesture diversity through an augmented dataset [9]. Kumar et al. tackled the challenging problem of achieving high gesture recognition accuracy with low computational time complexity by proposing a dual-camera system with two viewpoints for gesture recognition. Experimental results showed improved recognition rates compared to a traditional single-camera system [10]. Salvador et al. addressed the limitations faced by surgeons when interacting with computer systems and proposed a gesture recognition framework based on deep computer vision to facilitate interaction. Experimental results demonstrated the feasibility of the application and defined a gesture dictionary related to medical image navigation [11].

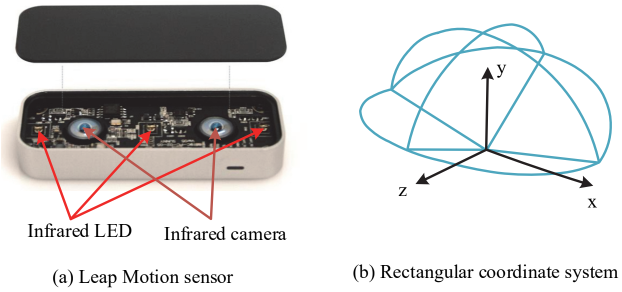
In recent years, the research of Leap Motion and Bi-LSTM have been extensive. Liu et al. addressed the issue of most existing robot arms being controlled by wired or wearable controllers by developing a six-axis robot arm with a leap motion controller. Experimental results showed that the six-axis robot arm achieved high accuracy [12]. Ovur et al. invented an adaptive multi-sensor fusion method to improve the success of sensors in subtle interaction scenarios. Experimental results demonstrated a significant improvement in the smoothness of pose estimation without affecting the occlusion of individual sensors [13]. Xia et al. tackled the time-consuming, expensive, and inflexible nature of traditional development methods for gesture recognition systems by using reconstructed hand motions to generate simulated signals with a virtual distance sensor and training a convolutional neural network model. Experiment outcomes said that it offered correct deployment suggestions, and the model could precisely identify real gestures [14]. Han et al. addressed the size limitation by proposing a novel fault classification method based on Bi-LSTM and convolutional neural network. Experimental results demonstrated good performance and noise resistance of the proposed method [15]. Bai et al. proposed a Bi-LSTM model to estimate short-term tidal levels, addressing the problem of poor short-term tidal estimation. Experimental results showed that the Bi-LSTM model had excellent capabilities in short-term tidal estimation [16]. Du et al. addressed the lack of consideration for the sequential patterns of patent acquisition activities and the potential diversity of company business interests by proposing a knowledge-aware attention-based Bi-LSTM network method for patent transaction recommendation. Experimental results showed that this method could provide accurate recommendations [17].

To sum up, the existing research in the field of GDR mainly focuses on improving the accuracy and real-time of recognition, relying on simple image processing and template matching. Therefore, based on the existing work, a GDR multimedia terminal system based on Leap Motion sensor and Bi-LSTM network is proposed to meet people's needs for intelligent interaction. Compared to traditional rule-based approaches and a single machine learning model, the Leap Motion sensor and Bi-LSTM-based approach adopted in this study demonstrates a stronger ability to process continuous dynamic gestures. In particular, the bi-directional information processing mechanism of Bi-LSTM can better understand the context information of gestures. Compared with the existing work, this study is enhanced in terms of the diversity of the dataset, the generalization ability of the model, and the comprehensiveness of the experimental design.

3 RESEARCH MODEL

3.1 Model Settings

In this paper, GDR data is collected by Leap Motion sensor, which leads to the model hypothesis and environment of the study. The Leap Motion sensor is known for its high precision and low latency properties, enabling the real-time tracking of hand movements and gestures, which provides the technical basis for the accurate capture of dynamic gestures. Leap Motion offers a rich software development kit that makes it easy for developers to integrate sensors into a variety of applications and systems to accelerate the development process. Due to its portability and compatibility, the Leap Motion sensor is suitable for a variety of devices, including computers, virtual reality devices, etc., which opens the possibility for a wide range of applications of GDR technology [18, 19]. The architecture of Leap Motion is shown in Fig. 1.



(a) Leap Motion sensor (b) Rectangular coordinate system

Figure 1 Leap Motion's architecture diagram

As shown in Fig. 1, the Leap Motion sensor is equipped with units in millimeters. The coordinate system originates from the center of the rectangular black screen on the sensor. Within the black screen, the *x*-axis and *z*-axis run parallel to the longer and shorter sides of the sensor, respectively. The *y*-axis is perpendicular to the black screen. A complete dynamic gesture comprises various hand poses at different time points. Each hand pose can be represented by a series of hand information, including palm coordinates, palm velocity, fingertip velocity, finger length, width, and interjoint angles. The hand model in Leap Motion is shown in Fig. 2.

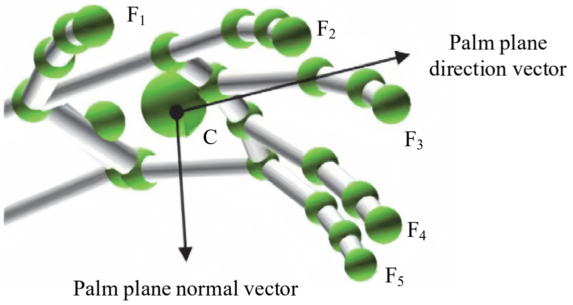


Figure 2 Hand modeling in Leap Motion

O is defined as the origin of the Leap Motion sensor's coordinate of three-dimension, and *C* is defined as the three-dimensional coordinates of the Leap Motion coordinate system palm center. The gap of the palm center and fingertip is given by Eq. (1).

$$D_i = \left\| \overrightarrow{OF_i} - \overrightarrow{OC} \right\| (i = 1, 2, 3, 4, 5) \tag{1}$$

In Eq. (1), D_i represents the Euclidean distance between the three-dimensional coordinates of the fingertip and the palm center, and F_i represents the three-dimensional coordinates of the fingertip in the Leap Motion coordinate system. The angle between the finger and the palm plane is given by Eq. (2).

$$\theta_i = \angle(\overrightarrow{OF_i^p} - \overrightarrow{OC}, \overrightarrow{OF_i} - \overrightarrow{OC})(i = 1, 2, 3, 4, 5) \tag{2}$$

In Eq. (2), θ_i represents the angle between lines connecting the palm center and the projection point of the fingertip on the palm plane, and F_i^p represents the projection point of the fingertip F_i on the palm plane. The height of the fingertip is given by Eq. (3).

$$H_i = \operatorname{sgn}(\overrightarrow{F_i^p F_i} \cdot \vec{N}) \times \left\| \overrightarrow{F_i^p F} \right\| (i = 1, 2, 3, 4, 5) \tag{3}$$

In Eq. (3), H_i represents the distance from the fingertip point F_i to the palm plane, and $\vec{N}$ represents the normal vector of the palm plane. The distance between adjacent fingertips is given by Eq. (4).

$$DD_i = \left\| \overrightarrow{F_i F_{i+1}} \right\| (i = 1, 2, 3, 4) \tag{4}$$

In Eq. (4), DD_i represents the distance between adjacent fingertips. The angle between the line connecting adjacent fingertips and the palm center is given by Eq. (5).

$$\Theta_i = \angle(\overrightarrow{CF_i}, \overrightarrow{CF_{i+1}})(i = 1, 2, 3, 4) \tag{5}$$

In Eq. (5), Θ_i represents the angle between the line connecting adjacent fingertips and the palm center. The palm center coordinates are $(x_{palm}, y_{palm}, z_{palm})$. These gesture features form the final 26-dimensional feature vector, as shown in Eq. (6).

$$X_t = \begin{Bmatrix} D_1, \dots, D_5, \theta_1, \dots, \theta_5, H_1, \dots, H_5, DD_1, \dots, DD_4, \\ \Theta_1, \dots, \Theta_4, x, y, z \end{Bmatrix} \tag{6}$$

In Eq. (6), this feature vector combines the single finger features, double finger, and palm center features. It solves the wrong classification caused by executing dynamic gestures and also distinguishes the differences between adjacent fingers. The entire dynamic gesture process can be represented by a series of continuous feature vectors, as shown in Eq. (7).

$$X = \left\{ X_1, X_2, \dots, X_j, \dots, X_T \right\} \tag{7}$$

In Eq. (7), T represents the time required to complete the entire dynamic gesture. Since different individuals may perform the same gesture at different speeds, T varies. The real-time and accuracy of gesture acquisition are important factors in improving the DGR accuracy. The research uses thresholds for palm rotation angles and fingertip velocity magnitudes for getting the start and end of dynamic gestures. The rotation angles of the palm in the three coordinate axes need to be obtained by comparing and calculating the historical and current frames, while the fingertip velocity can be directly obtained by calling the Finger class of Leap Motion, as shown in Eq. (8).

$$\begin{cases} \theta_x = frame.RotationAngle(LastFrame, x) \\ \theta_y = frame.RotationAngle(LastFrame, y) \\ \theta_z = frame.RotationAngle(LastFrame, z) \end{cases} \tag{8}$$

In Eq. (8), θ represents the rotation angle of the palm relative to the sensor coordinate system in the current frame and historical frames. The palm rotation angle is calculated as shown in Eq. (9).

$$\theta_{palm} = \sqrt{\theta_x^2 + \theta_y^2 + \theta_z^2} \tag{9}$$

In Eq. (9), the final palm rotation angle is obtained by calculating the Euclidean distance of the angles. This value can be used to detect dynamic gestures. For example, flipping the palm when the palm center does not move. The motion velocity of the fingertips is shown in Eq. (10).

$$V_{finger} = Finger.tipVelocity() \tag{10}$$

In Eq. (10), the motion velocity of the fingertips can be obtained by calling the tipVelocity function of the Finger class in the Leap Motion library. Then, the Euclidean distance of the velocity is calculated to obtain the magnitude of the motion velocity for each fingertip. When the palm moves, the fingertips also move. This value can detect dynamic gestures where the palm center moves, as well as dynamic gestures where the palm center does not move but the fingertips move. The process of dynamic gesture acquisition determination is shown in Fig. 3.

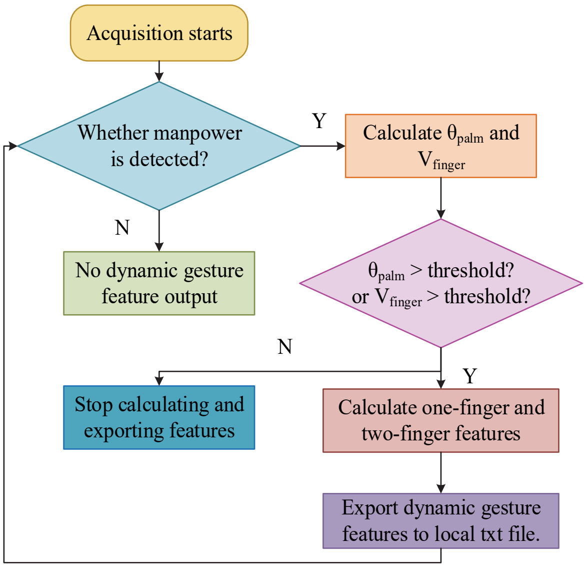

Acquisition starts
Whether manpower is detected?
Y
N
Calculate θ_{palm} and V_{finger}
No dynamic gesture feature output
θ_{palm} > threshold? or V_{finger} > threshold?
N
Y
Stop calculating and exporting features
Calculate one-finger and two-finger features
Export dynamic gesture features to local txt file.

Figure 3 Dynamic gesture acquisition decision flowchart

In Fig. 3, first, the function of the Hand class is called to return 0 or 1, which determines whether there is a hand or a hand-like object within the capture range of Leap Motion. If there is, the palm rotation angle and fingertip velocity magnitude are further calculated. If not, the gesture features are not calculated and output. Then, the calculated values are compared and determined against the thresholds for palm rotation angle and fingertip velocity magnitude. If either the palm rotation angle or the fingertip velocity is greater than its corresponding threshold, the hand is considered to be in motion, and the six mentioned features are immediately calculated and output. If both values are less than or equal to their corresponding thresholds, the calculation and output of features are stopped. The research uses a standardization method to normalize the same gesture features of different samples in each gesture library, mapping the same gesture features of different gesture samples to a unified standard. After standardization, the average value of each feature becomes 0, the variance becomes 1, the data follows a normal distribution, and it is dimensionless. In the same gesture library, for the 26 features in the gesture feature vector, the standardization method for each feature is shown in Eq. (11).

$$\begin{cases} mean_k = \dfrac{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{T} X_{i,j,k}}{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{T} 1} \\ variance_k = \sqrt{\dfrac{1}{(\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{T} 1)-1}\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{T}(X_{i,j,k}-mean_k)^2} \\ X'_{i,j,k} = \dfrac{X_{i,j,k}-mean_k}{variance_k} \end{cases} \quad (11)$$

In Eq. (11), i represents the i-th sample, j represents the j-th feature vector in the sample, k represents the k-th feature value in the feature vector, ranging from 1 to 26, $X_{i,j,k}$ represents the k-th feature value of the j-th sequence of the i-th sample. m represents the number of samples, and T represents the length of a single sample sequence. For each partitioned gesture library, the data in the training set is first normalized to obtain 26 sets of mean and variance. Then, the mean and variance obtained from normalization are hired to normalize the test set. This ensures the test and training set independence, while also aligning the test set data with the format of the training set data for subsequent experiments. It is assumed that the Leap Motion sensor can provide accurate gesture data, user gestures have certain regularity, and the selected features can effectively represent gesture information. In terms of the research environment, the hardware and software configuration required for the experiment included the Leap Motion sensor, the detailed configuration of the laptop, and the software development tools used to ensure the controllability of the experiment and the reliability of the results. In summary, the model setting in this section provides a clear and systematic framework for the study of GDR tasks. Through these settings, a solid foundation has been laid for the next model proposal and verification, and to ensure the scientific and systematic research.

3.2 The Proposed Model

Based on the model setting in the first section, the model proposed in this study will be elaborated in detail. Bi-LSTM network, as a special recurrent neural network, performs well in processing data with time series characteristics, and can capture the changes of gestures over time. Compared to traditional one-way LSTM, Bi-LSTM is able to consider both past and future information, which allows the network to better understand and predict the context and development trends of dynamic gestures. Bi-LSTM networks have demonstrated superior performance in multiple sequence prediction tasks, especially in scenarios requiring long and short term memory and complex pattern recognition [20, 21]. Leap Motion sensors can provide rich hand motion data, and the Bi-LSTM network can effectively extract key features from these data, providing strong data support for GDR. Through the deep learning capability of the Bi-LSTM network, the system can learn the internal patterns of complex gestures, improve the robustness to environmental noise and change, and enhance the generalization ability of the model. In this study, a two-layered Bi-LSTM was constructed by stacking two bidirectional structures, as shown in Fig. 4.

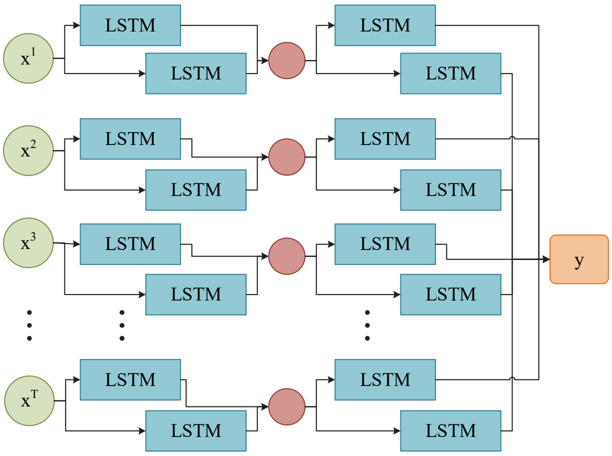


Figure 4 Two-Layered Bi-LSTM

In Fig. 4, the study uses the two-layered Bi-LSTM and each layer of Bi-LSTM consists of two directional LSTM units, namely, forward propagation unit and backward propagation unit. The input data for each time step is processed by LSTM units in both directions at the same time, and the information from both directions is then combined to provide more comprehensive contextual information for the next layer. The Softmax activation function is used in the final layer of the Bi-LSTM network to map each element of the output vector to between 0 and 1, thus making predictions of classes. The first layer Bi-LSTM sets up 256 hidden units to capture as much raw features and information as possible from the input data. The second layer Bi-LSTM sets up 128 hidden units to abstract and refine information in the deep layers of the network, while reducing the complexity of the model and avoiding overfitting. The sequence is input frame by frame into the first layer of Bi-LSTM, and then the results of the forward and backward units at the same time step are combined and output to the corresponding units in the second layer. The basic units of the two layers of Bi-LSTM can be seen as a forward-propagating LSTM unit and a backward-propagating LSTM unit. The first layer output is defined as shown in Eq. (12).

$$x_2^t = \overrightarrow{a}_1^t + \overleftarrow{a}_1^t \tag{12}$$

In Eq. (12), x_2^t represents the input to the second layer network, $\overrightarrow{a}_1^t$ represents the output of the forward-propagating LSTM unit at time step t in the first layer, and $\overleftarrow{a}_1^t$ represents the output of the backward-propagating LSTM unit at time step t in the first layer. Finally, the final output of the entire two-layer Bi-LSTM is defined as shown in Eq. (13).

$$\hat{y}_2 = softmax(\overrightarrow{W}_{y,2} \cdot \sum_{t=1}^{T} \overrightarrow{a}_2^t + \overleftarrow{W}_{y,2} \cdot \sum_{t=1}^{T} \overleftarrow{a}_2^t + b_{y,2}) \tag{13}$$

In Eq. (13), *softmax* is the softmax output function, commonly used in the output layer of classification models. For the two-layer Bi-LSTM, after the error propagates to the two unidirectional networks in the second layer, it will also propagate to the two unidirectional networks in the first layer, resulting in four directions of error backpropagation. Therefore, the model needs to be trained. The essence of training a neural network model is to iteratively change the weights *W* and biases *b* to acquire a min loss function. However, one problem in training these parameters is how to initialize them. Arbitrary initialization of weights can slow down or even stop the convergence process, i.e., the error decreases slowly or remains unchanged. This is because arbitrary initialization of weights may result in a very small variance of the input received by deeper network layers, slowing down the backpropagation process and delaying the entire convergence process. When the model has many layers, Xavier initialization or MSRA initialization can be chosen [22, 23], as these two initializations can alleviate the problem of gradient vanishing caused by too many layers. When there are fewer layers or only one layer, Gaussian distribution initialization, small random number initialization, and Positive unitball initialization can be chosen. In neural networks, gradient descent is usually used to update various weights or bias parameters, as shown in Eq. (14).

$$X = X - \alpha \cdot \frac{\partial L}{\partial X} \tag{14}$$

In Eq. (14), *L* is the loss function, *α* is the learning rate, and *X* is any parameter in the neural network. For training deep neural networks, underfitting and overfitting are two common problems. The principle of dropout is to randomly deactivate the activation values of neurons in the neural network with a certain probability. This can improve the generalization ability and prevent overfitting. The main implementation of dropout is to multiply the output of each layer, except the last layer, in the deep neural network by a vector that follows a Bernoulli distribution, as shown in Eq. (15).

$$X_{\text{out}} = X_{\text{out}} * D \tag{15}$$

In Eq. (15), each element of the *D* vector is either 1 or 0, following a Bernoulli distribution. The probability of 1 appearing is 1 minus the dropout rate, and the probability of 0 appearing is the dropout rate. The dropout of long short-term memory networks is shown in Fig. 5.

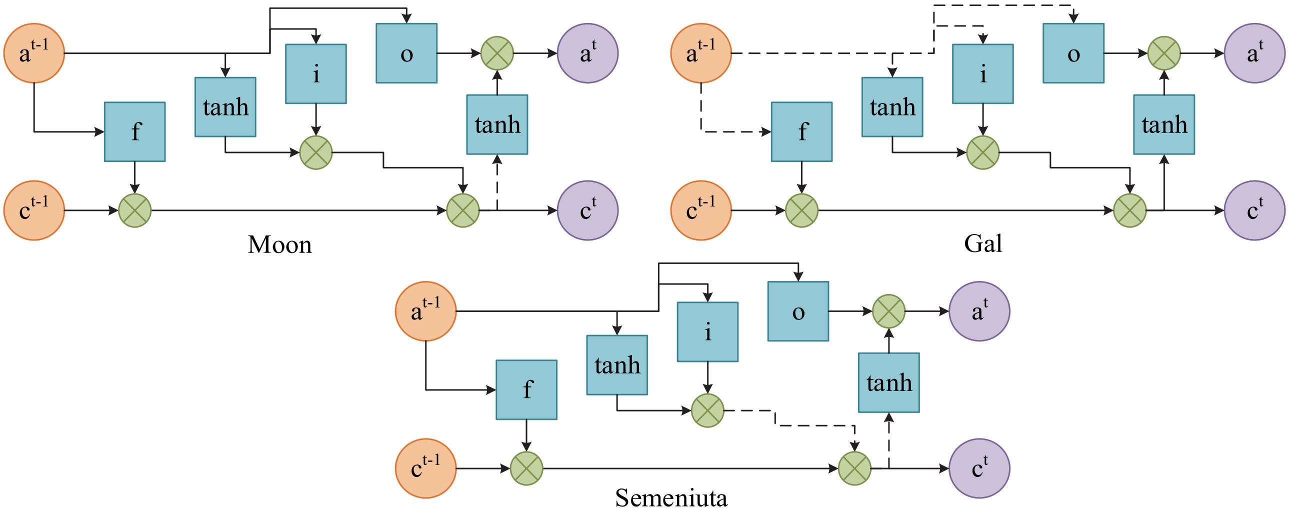


Figure 5 Stochastic inactivation of long and short-term memory networks

In Fig. 5, dropout is applied to the hidden state nodes, and each different unit has a different dropout vector. The essence of training is to repeatedly update the weights for the min loss value. Through the backpropagation algorithm, the derivatives of the loss function, i.e., the gradients, can be obtained. Gradient descent is the process of updating the weights using these derivative values. Common gradient descent algorithms are listed, and the comparison of gradient descent methods is shown in Fig. 6.

In Fig. 6, RMSprop gradient descent further improves the oscillation problem of parameters and the loss value, and speeds up the convergence. Adam gradient descent combines momentum gradient descent and RMSprop gradient descent, making the oscillation of parameters and the loss function more stable, and further accelerating the convergence. In the designed network, softmax regression is used on the second layer output. With using the softmax regression as the activation function of the last layer of the recognition model, each element of the output vector is mapped to a value between 0 and 1, and the index of the maximum element is the classification of the recognized sample.

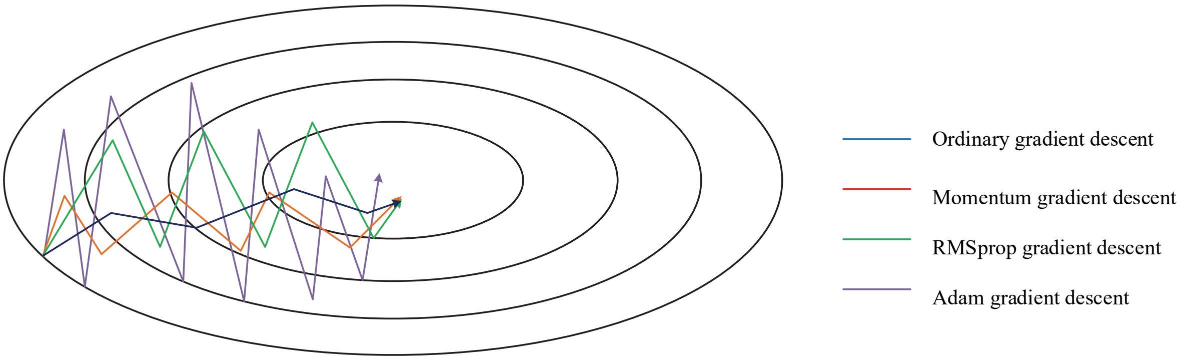


Figure 6 Comparison of gradient descent

3.3 Model Validation

The first two sections have studied the setting of the model and the specific structure and function of the model. To verify the effectiveness of the model, this section designed an experiment of dynamic gesture library to analyze the model. Leap Motion sensors, VS 2019 programming software, a laptop and two cloud servers were used in the experiments. The laptop was configured with a 3.2GHz i5 processor and 8 GB of RAM, one of the computational cloud servers had the same configuration as the laptop, while the other standard cloud server was configured with a 3.2GHz i5 processor and 4 GB of random access memory. The study used the Leap Motion sensor as a gesture data acquisition tool, which was able to track hand movements and gestures in real time. The sensor was installed in front of the user and ensured that the user's hand movement was within the tracking range of the sensor. Collected data included hand shape, speed, acceleration and other characteristic information. The data collected by the sensor was saved in a local text file to provide the original data source for subsequent processing and analysis. It should check the data set for invalid data and outliers and remove or correct them to ensure the quality of the data set. Filtering algorithms were applied to reduce noise in data, such as Gaussian filtering or median filtering, to improve the accuracy and reliability of data. Key gesture features such as palm coordinates, palm speed, fingertip speed, finger length, width, and knuckle Angle were extracted from the raw data. For each feature vector, a standardized method was adopted to make the data have zero mean and unit variance, thus eliminating the influence of different dimensions.In the process of collecting and using gesture data, informed consent was obtained from participants, who were informed about the purpose of the study, the process, the potential risks, and how their data are used and stored. The study anonymized the data to protect participants' privacy. It is necessary to avoid disclosing personally identifiable information in any public report or release. It is necessary to take appropriate technical and administrative measures to protect data from unauthorized access, disclosure or misuse. The use of data should be limited to the purpose of the study and should not be used for other purposes unrelated to the study. The processed dynamic gesture library is divided into training set and test set. The training set is used to train the recognition model and the test set is used to test the performance of the model. The division is done in such a way that the data distribution of the training set and test set is consistent to improve the accuracy and reliability of the experiments. The two-layer Bi-LSTM network model is trained using the training set [24]. By adjusting the parameters and structure of the model, it enables the model to learn and recognize dynamic gestures better. During the training process, optimization algorithms such as gradient descent can be used to minimize the loss function of the model and improve the accuracy of the model. The trained recognition model is tested using a test set and the recognition results of the model on the test set are recorded. The recognition results include information such as the category and confidence level of the gesture. Based on the recognition results, evaluation metrics such as precision, recall, and F1 score of the model are calculated to assess the performance of the model. The experimental flow is shown in Fig. 7.

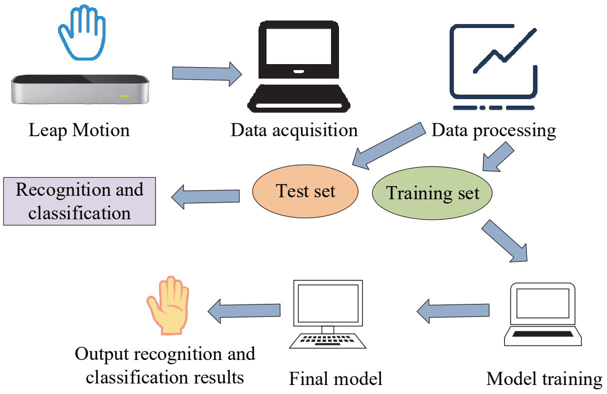


Figure 7 Experimental process

To evaluate the designed recognition model more comprehensively, three important evaluation metrics are introduced: precision rate, recall rate and F1 score. These three metrics can reflect the recognition effect of the model more precisely. Specifically, precision rate is used to measure the accuracy of the model in the recognition process, as shown in Eq. (16).

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

In Eq. (16), *TP* denotes the number of samples indicating that the true label belongs to the positive class and the predicted value is positive. *FP* denotes the number of samples showing that the true label belongs to the negative class but the predicted value is positive. Recall, on the other hand, reflects the degree of coverage of the model in all relevant samples, as shown in Eq. (17).

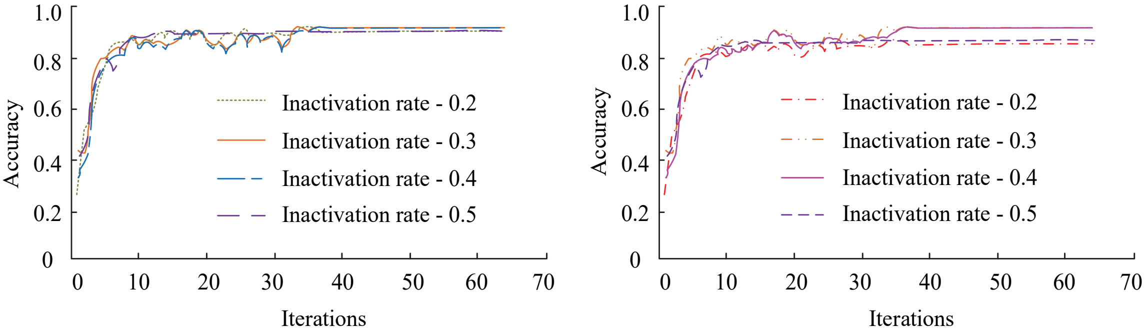
$$Recall = \frac{TP}{TP + FN} \tag{17}$$

In Eq. (17), *FN* denotes the number of samples whose true labels belong to the positive class but whose predicted values are negative. The F1 score, on the other hand, is the reconciled average of precision and recall, which can comprehensively reflect the overall performance of the model, as shown in Eq. (18).

$$F1\text{-}score = \frac{precision \cdot recall}{2(precision + recall)} \tag{18}$$

4 RESULT AND DISCUSSION

The study used Leap Motion sensors to capture dynamic gestures and saved them in a local txt text file as a means of constructing an ASL library and a Craft Gesture library. Subsequently, these dynamic gesture libraries were divided into a training set and a test set. Next, experiments were conducted on these two sets, using the training set to train the previously proposed two-layer Bi-LSTM model to gradually approximate the ideal model. Afterwards, the trained recognition model was tested using the test set and its accuracy on the test set was recorded as the final recognition result. The experimental results were analyzed in detail as a way to demonstrate the effectiveness of the designed recognition model. When training the model on the dynamic gesture library, the weight matrix was initialized with a Gaussian distribution, and the bias matrix was initialized to all zeros. The learning rate was fixed at 0.001, and Adam optimization with mini-batch gradient descent was used [25]. The study set the initial learning rate to 0.001, if the learning rate was too high, it might cause shock or miss the optimal solution during training. If the learning rate was too low, it might lead to a slow training process or even a local minimum. The Dropout rate was set at 0.4. The Dropout rate determined how many neurons were discarded in each iteration. The choice of Dropout rate depended on the complexity of the model and the training data. With the batch size set to 64, a smaller batch size provided a more stable gradient estimate, while a larger batch size sped up training. The study used an Adam optimizer with adaptive learning rate, which could automatically adjust the learning rate during training. For the ASL library and the Craft Gesture library, the recognition model was run for approximately 70 epochs by changing the inactivation rate, as shown in Fig. 8.

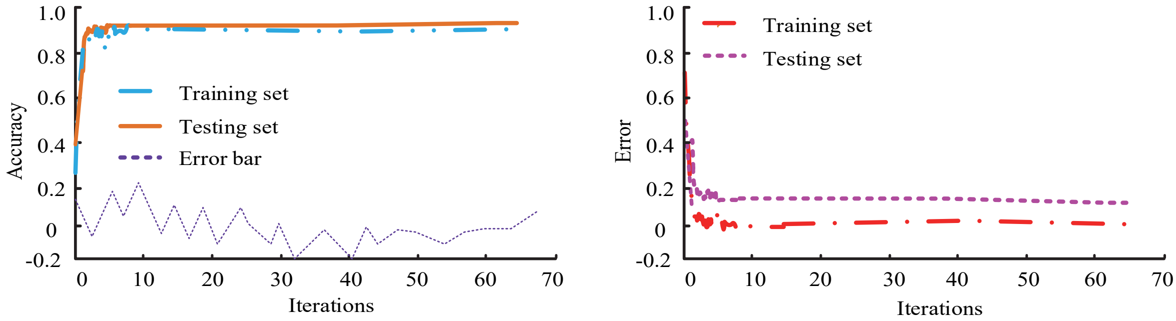


(a) American sign language library

(b) Craft sign language library

Figure 8 Comparison of different inactivation rates on classification accuracy of recognition models

In Fig. 8, for the ASL library, an appropriate inactivation rate was around 0.3. For the Craft Gesture library, an appropriate inactivation rate was around 0.3 or 0.5. The ASL library was divided as 7:3 ratio for train and testing set. The recognition model was then trained on the training set for 650 epochs, and the effectiveness of the model was validated using the test set. The final results of the model testing in the ASL library are shown in Fig. 9.

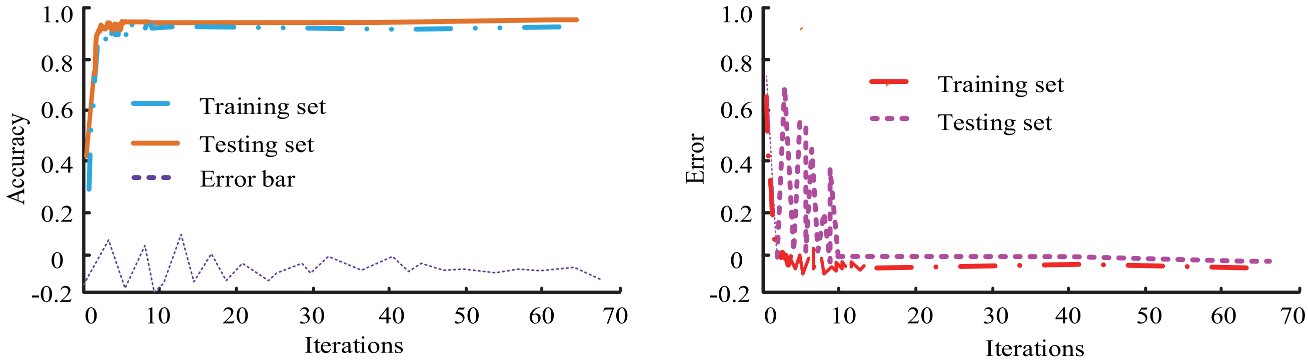


(a) Accuracy iteration chart

(b) Error iteration chart

Figure 9 Results of model testing in an ASL library

In Fig. 9, the training recognition accuracy reached 99%, and the test value was 97%. The training error converged to within 0.2, and the test error converged to within 0.01. The same experiments and evaluations were conducted on the Craft Gesture library, which was also categorized for training and test. 650 epochs was the train numbers, and the effectiveness of the recognition model was validated using the test set. The final results of the model testing in the Craft Gesture library are shown in Fig. 10.



(a) Accuracy iteration chart

(b) Error iteration chart

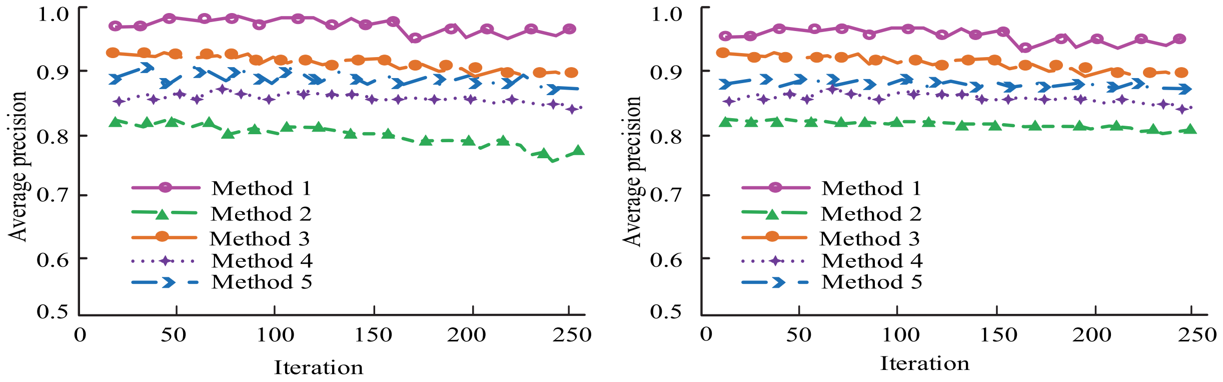
Figure10 Results of model testing in the Craft sign language library

In Fig. 10, the training recognition accuracy reached 97%, and the testing recognition accuracy reached 96%. The training error converged to within 0.2, and the that of testing converged to within 0.1. However, there were occasional mutations, which occurred randomly and at a low rate, so the model's performance was still considered good. To more accurately evaluate the designed recognition model, indicators were hired for model's performance assessing. Finally, these data were averaged. The indicators for the ASL library and the index results of craft language are displayed in Tab. 1.

In Tab. 1, the average precision, recall, and F1 score for the ASL library were 97, 97.11, and 98.22, respectively, indicating that the model performed well when applied to the ASL library. The average precision, recall, and F1 score for the Craft Gesture library were 96.78, 97.33, and 97.67, indicating that the model performed well when applied to the Craft Gesture library. To scientifically demonstrate the DGR performance (Method 1), it was compared with other models: recursive-based gesture recognition model (Method 2), 3D accelerometer-based gesture recognition model (Method 3), 3D convolutional neural network-based gesture recognition model (Method 4), and flexible neural tree and sEMG-based gesture recognition model (Method 5). The average accuracy comparison is shown in Fig. 11.

Table 1 The index results of the American and craft manual language database

American hand language library				Craft language database			
Gesture categories	Precision	Recall	F1 Score	Gesture categories	Precision	Recall	F1 Score
Left	100	100	100	Left	100	100	100
Right	90	92	98	Right	95	94	99
Forward	95	96	96	Forward	90	94	94
Backward	100	100	100	Backward	100	100	100
Please transfer	98	92	94	Please transfer	96	97	96
Received	92	98	98	Received	94	95	92
Reject	100	100	100	Reject	100	100	100
Goodbye	100	100	100	Goodbye	100	100	100
Learning	98	96	98	Learning	96	96	98
Average score	97	97.11	98.22	Average score	96.78	97.33	97.67



(a) American sign language library

(b) Craft Sign language library

Figure 11 Comparison of average accuracy of different models

In Fig. 11, the average accuracy of Model 2, Model 3, Model 4, and Model 5 was 82%,91%, 89%, and 88%, respectively, while Model 1 had an average accuracy of 98%, which was higher than the average accuracy of the other four models. Therefore, the proposed DGR model has the best recognition performance. The study showed the performance comparison of different models, including confidence intervals, to study the improvement of model performance, as shown in Tab. 2.

Table 2 Confidence interval comparison of models

Model type	Accuracy rate / %	Standard deviation	95% confidence interval	p
Method 2	82	2.5	82.13 - 87.87	<0.001
Method 3	91	1.8	90.38 - 93.62	<0.001
Method 1	98	1.2	95.58 - 98.42	<0.001

In Tab. 2, 30 independent experiments were conducted for each model, and the corresponding accuracy and standard deviation were obtained. From the perspective of confidence interval, the accuracy of Method 1 was significantly higher than that of Method 3 and Method 12, and the confidence interval was very narrow, indicating that the results were very stable.Finally, the GDR model (Model 1) based on Leap Motion and Bi-LSTM was applied to the real-world scene and compared with the latest GDR model. Comparison models included Transformer model based on self-attention mechanism (Model 2), GDR model based on graph convolutional network (Model 3), and GDR model based on long short-term memory network (Model 4). The comparison indexes included: computing efficiency, robustness, scalability, real-time, user experience, user acceptance, and anti-interference ability of the model. All indicators were normalized. The final results are shown in Tab. 3.

Table 3 Performance comparison of different dynamic gesture recognition models

Models	Model 1	Model 2	Model 3	Model 4
Computational efficiency	0.95	0.85	0.78	0.70
Robustness	0.96	0.82	0.80	0.75
Extensibility	0.92	0.88	0.84	0.76
Real-time performance	0.97	0.80	0.79	0.72
User experience sense	0.98	0.87	0.82	0.74
User acceptance	0.95	0.83	0.80	0.73
Anti-interference capability	0.96	0.84	0.81	0.71

In Tab. 3, Model 1performed well on all indicators, especially in the sense of user experience and real-time performance, showing that the model has significant advantages in providing a smooth and intuitive user experience. Model 2 performed well in terms of computational efficiency and robustness, but was slightly lower than Model 1 in terms of real-time and scalability. Model 3 performed moderately in terms of robustness and anti-interference ability, but fell slightly short in terms of computational efficiency and user experience.

Based on the above analysis of the experimental results, the DGR model exhibited high recognition accuracy on both the training and test sets, and the error was effectively converged. In the American hand language database, the recognition accuracy of the model training set reached 99%, and the test set reached 97%. On the process gesture library, the recognition accuracy of model training set was 97%, and that of test set was 96%. The error on the training set converged to less than 0.2, and the error on the test set converged to less than 0.01, which indicated that the model had stable performance on both training and test data. The average precision rate, recall rate and F1 score of American hand language database were 97, 97.11 and 98.22, respectively. The average precision rate, recall rate and F1 score of craft language database were 96.78, 97.33 and 97.67, respectively. The model showed good performance on two different gesture libraries, which indicated that the model had good generalization ability. The model can capture and recognize gestures in real time, which is crucial for GDR. There are some errors in the model on the test set, which means there are problems in the recognition of some specific gestures. Some gestures may be visually or dynamically very similar, making the model difficult to distinguish. The data sets used in experiments may not fully represent the diversity of gestures in the real world, which affects the model's ability to generalize in the real world. The model has yet to be optimized in processing the continuity and timing information of dynamic gestures to adapt to the needs of faster and more complex gesture recognition. The GDR model based on Leap Motion sensor and Bi-LSTM has important practical significance in the practical application of digital terminal [26]. GDR provides a more natural and intuitive way of interaction, enabling users to communicate with digital terminals through gestures, improving the convenience and intuitiveness of operations. For people with disabilities, GDR provides an alternative way to interact with traditional input devices, thereby improving the accessibility of digital terminals. In the field of education, GDR can be integrated into educational software and applications to provide students with an interactive learning experience and enhance learning motivation. In intelligent environments such as smart homes and smart offices, GDR can be used as a means to control home or office equipment and improve the intelligence level of the environment. In environments where both hands are required to operate, gesture recognition can be used as a non-contact interaction mode to reduce physical contact with the device and improve the safety of operation.

5 CONCLUSION

HCI technology plays a crucial role as a bridge between humans and digital media terminals. A DGR multimedia terminal system based on Leap Motion sensor and Bi-LSTM was designed in this study. This system has excellent multimedia terminal operation capabilities and provides a more convenient and user-friendly way of interaction. The study successfully achieved high-precision gesture capture by Leap Motion sensor, which provided a reliable data base for subsequent gesture recognition. Secondly, the Bi-LSTM network showed a strong ability in processing temporal data, with high accuracy in recognition and classification of gestures and good generalization performance. The model performed well on several evaluation metrics and provided a new solution in the field of DGR. The GDR system based on Leap Motion sensor and Bi-LSTM greatly enriched the HCI mode by providing an efficient and accurate gesture recognition method, so that users can communicate with digital terminals more naturally and intuitively. This kind of interaction is expected to improve user experience, increase user satisfaction, and provide new ideas for designing interactive interfaces that are more in line with user needs. The research results can be directly applied to multimedia equipment, smart home control system, virtual reality and augmented reality and other fields to promote the innovation and function of terminal design. Although the system proposed in the study performed well in experiments, there are still some limitations. For example, the current model has a delay in processing gestures in quick succession, which can be a problem in application scenarios that require quick responses. In addition, the ability of the model to recognize atypical gestures has not been fully verified, which limits the application of the model to a wider user group. Current data sets do not fully cover all possible gesture variations and user diversity. Future research can focus on the optimization of the algorithm to improve the recognition accuracy and response speed of the model to complex dynamic gestures. In addition, the study can explore the robustness of the model to different lighting conditions, background noise, and other environmental disturbances. The cross-cultural adaptability of the model can also be studied, considering that the user's gesture habits may be different in different cultural backgrounds, the research can focus on how to make the system better adapt to and recognize gestures in different cultures.

6 REFERENCES

[1] Diederich, S., Brendel, A. B., Morana, S., & Kolbe, L. (2022). On the Design of and Interaction with Conversational Agents: An Organizing and Assessing Review of Human-Computer Interaction Research. *Journal of the Association for Information Systems*, *23*(1), 96-138. https://doi.org/10.17705/1jais.00724

[2] Liang, W. (2023). Retraction Note: Scene art design based on human-computer interaction and multimedia information system: an interactive perspective. *Multimedia Tools and Applications*, *82*(10), 15919-15919. https://doi.org/10.1007/s11042-018-7070-6

[3] Zhao, L. (2023). International Art Design Talents-oriented New Training Mode Using Human-Computer Interaction based on Artificial Intelligence. *International Journal of Humanoid Robotics*, *20*(04), 2250012-2250012. https://doi.org/10.1142/S0219843622500128

[4] Moencks, M., Roth, E., Bohne, T., & Kristensson, P. O. (2022). Human-Computer Interaction in Industry: A Systematic Review on the Applicability and Value-addedof Operator Assistance Systems. *Foundations and trends in human-computer interaction*, *16*(2/3), 1-154. https://doi.org/10.1561/1100000088

[5] Li, H., Lin, Z., An, Z., ZuoS., Zhu, W., Zhang, Z., Mu, Y., Cao, L., & García, J. D. P. (2022). Automatic electrocardiogram detection and classification using bidirectional longshort-term memory network improved by Bayesian optimization. *Biomedical signal processing and control*, 73(Mar.), 103424.1-103424.8. https://doi.org/10.1016/j.bspc.2021.103424

[6] Ge, Z., Wu, Z., Han, X., & Zhao, P. (2023). Gesture Recognition and Master - Slave Control of a Manipulator Based on sEMG and Convolutional Neural Network-Gated Recurrent Unit. Journal of Engineering and Science in Medical Diagnostics and Therapy, *6*(2), 021004-021013. https://doi.org/10.1115/1.4056325

[7] Alashhab, S., Gallego, A. J., & Lozano, M. A. (2022). Efficient gesture recognition for the assistance of visually impaired people using multi-head neural networks. *Engineering Applications of Artificial Intelligence*, *114*(1), 1-21. https://doi.org/10.1016/j.engappai.2022.105188

[8] Wu, Z., Ma, N., Gao, Y., Li, J., Xu, X., Yao, Y., & Chen, L. (2022). Attention Mechanism Based on Improved Spatial-Temporal Convolutional Neural Networks for Traffic Police Gesture Recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, *36*(8), 1-19. https://doi.org/10.1142/S0218001422560018

[9] Zhang, J., Wang, F., & Lan, F. (2021). Dynamic Hand Gesture Recognition Based on Short-Term Sampling Neural Networks. *IEEE/CAA Journal of Automatica Sinica*, *8*(1), 114-124. https://doi.org/10.1109/JAS.2020.1003465

[10] Kumar, A. K., Kumar, A. K., & Guo, S. (2021). Two view points based real-time recognition for handgestures. *IET Image Processing*, *14*(5), 1456-1458. https://doi.org/10.1049/iet-ipr.2019.1458

[11] Salvador, R. A. A., &Naval, P. C. (2022). Towards a Feasible Hand Gesture Recognition System as Sterile Non-contact Interface in the Operating Room with 3D Convolutional Neural Network. *Informatica: An International Journal of Computing and Informatics*, *46*(1), 1-12. https://doi.org/10.31449/inf.v46i1.3442

[12] Liu, C. C., Hu, W. L., Liang, C. K., & Hsu, C. C. (2022). Wireless Control of Six-axis Robot Arms byLeap Motion Sensor. *Sensors and materials: An International Journal on Sensor Technology*, *34*(10), 3765-3779. https://doi.org/10.18494/SAM4080

[13] Ovur, S. E., Su, H., Qi, W., Momi, E. D., & Ferrigno, G. (2021). Novel Adaptive Sensor Fusion Methodology for Hand Pose Estimation With Multileap Motion. *IEEE Transactions on Instrumentation and Measurement*, *70*(99), 1-8. https://doi.org/10.1109/TIM.2021.3063752

[14] Xia, C., Saito, A., & Sugiura, Y. (2022). Using the virtual data-driven measurement to support the prototyping of hand gesture recognition interface with distance sensor. *Sensors and Actuators A: Physical*, *338*(1), 113463-11374. https://doi.org/10.1016/j.sna.2022.113463

[15] Han, T., Ma, R., & Zheng, J. (2021). Combination bidirection allong short-term memory and capsule network for rotating machinery fault diagnosis. *Measurement*, *176*(1), 1-15. https://doi.org/10.1016/J.MEASUREMENT.2021.109208

[16] Bai, L. H. & Xu, H. (2021). Accurate estimation of tidal level using bidirectional longshort-term memory recurrent neural network. *Ocean engineering*, 235(Sep.1), 108765.1-108765.15. https://doi.org/10.1016/j.oceaneng.2021.108765

[17] Du, W., Jiang, G., Xu, W., & Ma, J. (2023). Sequential patent trading recommendation using knowledge-aware attentional bidirectional longshort-term memory network (KBiLSTM). *Journal of Information Science*, *49*(3), 814-830. https://doi.org/10.1177/01655515211023937

[18] O'Brien, H. L., Roll, I., Kampen, A., & Davoudi, N. (2022). Rethinking (Dis)engagement in human-computer interaction. *Computers in human behavior*, *128*(Mar.), 107109.1-107109.11. https://doi.org/10.1016/j.chb.2021.107109

[19] Kriglstein, S., Martin-Niedecken, A. L., Spjut, J., Damen, N. B., Tuerkay, S., & Drachen, A. (2022). Esports Meets Human-Computer Interaction. *Interactions*, *29*(3), 42-47. https://doi.org/10.1145/3524855

[20] Zhang, G., Wang, C., Long, J., Liu, Q., Wei, J., & Duan, L. (2021). Inertial Sensor-Based Motion Analysis System of Bridge-Style Movement For Rehabilitation Treatments. *Journal of mechanics in medicine and biology*, *21*(8), 2150066-2150066. https://doi.org/10.1142/S0219519421500652

[21] Ding, I. J. & Zheng, N. W. (2022). RGB-D Depth-sensor-based Hand Gesture Recognition Using Deep Learning of Depth Images with Shadow Effect Removal for Smart Gesture Communication. *Sensors and materials: An International Journal on Sensor Technology*, *34*(1 Pt.2), 203-216. https://doi.org/10.18494/SAM3557

[22] Xue, X., Jiang, C., Zhang, J., & Hu, C. (2021). Biomedical Ontology Matching Through Attention-Based Bidirectional Long Short-Term Memory Network. *Journal of Database Management: An Official Publication of the International Data Management Institute of the Information Resources Management Association*, *32*(4), 14-27.

https://doi.org/10.4018/JDM.2021100102

[23] Yang, M. (2022). Research on vehicle automatic driving target perception technology based on improved MSRPN algorithm. *Journal of Computational and Cognitive Engineering*, *1*(3), 147-151. https://doi.org/10.47852/bonviewJCCE20514

[24] Arslan, H., Işik, Y. E., Görmez, Y., & Temiz, M. (2024). Machine Learning and Text Mining based Real-Time Semi-Autonomous Staff Assignment System. *Computer Science and Information Systems*, *21*(1), 75-94. https://doi.org/10.2298/CSIS220922065A

[25] Yen, N. Y., Jeong, H., Madani, K., & Massetto, F. I. (2021). Guest Editorial: Emerging Services in the Next-Generation Web: Human Meets Artificial Intelligence. *Computer Science and Information Systems*, *18*(2), 1-4. https://doi.org/10.2298/CSIS210200iY

[26] Yao, B., Liu, S., & Wang, L. (2023). Using Machine Learning Approach to Construct the People Flow Tracking System for Smart Cities. *Computer Science and Information Systems*, *20*(2), 679-700. https://doi.org/10.2298/CSIS220813014Y

Contact information:

Jian DONG
Xi'an University of Science and Technology,
Art College, Xi'an University of Science and Technology, Xi'an, Shaanxi,
710054, China
E-mail: 18092533246@163.com