Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 5256940, 16 pages
https://doi.org/10.1155/2021/5256940

Research Article

Agricultural Machinery Virtual Assembly System Using Dynamic Gesture Recognitive Interaction Based on a CNN and LSTM Network

Po Zhang, Junqiang Lin, Jianhua He, Xiuchan Rong, Chengen Li, and Zeqin Zeng

College of Engineering, South China Agricultural University, Guangzhou 51000, China

Correspondence should be addressed to Po Zhang; 1101700433@qq.com

Received 22 July 2021; Revised 28 September 2021; Accepted 27 October 2021; Published 25 November 2021

Academic Editor: A. M. Bastos Pereira

The agricultural machinery experiment is restricted by the crop production season. Missing the crop growth cycle will extend the machine development period. The use of virtual reality technology to complete preassembly and preliminary experiments can reduce the loss caused by this problem. To improve the intelligence and stability of virtual assembly, this paper proposed a more stable dynamic gesture cognition framework: the TCP/IP protocol constituted the network communication terminal, the leap motion-based vision system constituted the gesture data collection terminal, and the CNN-LSTM network constituted the dynamic gesture recognition classification terminal. The dynamic gesture recognition framework and the harvester virtual assembly platform formed a virtual assembly system to achieve gesture interaction. Through experimental analysis, the improved CNN-LSTM network had a small volume and could quickly establish a stable and accurate gesture recognition model with an average accuracy of 98.0% (±0.894). The assembly efficiency of the virtual assembly system with the framework was improved by approximately 15%. The results showed that the accuracy and stability of this model met the requirements, the corresponding assembly parts were robust in the virtual simulation environment of the whole machine, and the harvesting behaviour in the virtual reality scene was close to the real scene. The virtual assembly system under this framework provided technical support for unmanned farms and virtual experiments on agricultural machinery.

1. Introduction

The agricultural machinery experiment is severely restricted by the production season. Missing the proper crop growth period led to an extension of the development cycle [1]. It was difficult to complete all of the experiments in one cycle. To solve such problems, some researchers chose agricultural machinery cross-regional experiments [2], but such experimental methods were only suitable for countries with abundant climate resources. With the development of virtual reality technology, some researchers have begun to use virtual reality technology to complete simulation experiments [3–5]. The application of virtual reality technology in the field of agricultural machinery was mainly concentrated in the design and manufacture of agricultural machinery products, and there were also successful cases in virtual experiments of agricultural machinery products [6–10]. However, the models in these schemes were all standard assemblies generated in 3D software. Skipping the steps of virtual equipment would cause the assembly dimension chain to be in a state of minimum or zero error. A series of experiments conducted in this state was not typical.

Accurate dynamic gesture recognition was a prerequisite for virtual assembly systems. Deep learning played an important role in gesture recognition technology due to its unique advantages. Lu et al. completed a one-time learning gesture recognition algorithm from the first perspective [11, 12]. Ameur et al. developed an intelligent training assistance system based on gesture recognition to automatically evaluate employee performance [13], and Bangaru et al. proposed a time

series classification algorithm for small training sets [14]. Fajardo et al. [15] proposed using time-spectrum discrete analysis to obtain manual features and a CNN to extract depth features to classify single-channel device recording signals. In other related areas of identification, attempts were made to improve accuracy or system robustness by combining traditional algorithms with new neural networks [16–20]. To adapt to the characteristics of a few samples and low storage space available for deep learning in the agricultural machinery industry, as well as the time and space dependence of dynamic gesture recognition, this paper combines an LSTM and a CNN to construct a CNN-LSTM network.

Based on the above understanding, the construction of an agricultural machinery virtual assembly system using dynamic gesture recognitive interaction based on a CNN-LSTM network was conducive to the error analysis of virtual simulation experiments and had practical importance for the design and simulation of agricultural machinery. Therefore, this paper proposed a more stable dynamic gesture cognition framework: the TCP/IP protocol constituted the network communication terminal, the leap motion-based vision system constituted the gesture data collection terminal, and the CNN-LSTM network constituted the dynamic gesture recognition classification terminal. The dynamic gesture recognition framework and the harvester virtual assembly platform formed a virtual assembly system to complete gesture interaction. Simultaneously, a simulation experiment scene was constructed, and comparative experiments were conducted with harvesting behaviour and binocular vision navigation, which confirmed the superiority of virtual simulation.

2. Virtual Assembly System Architecture and Principles

We used the harvester of an enterprise as an example to construct a virtual assembly system: use Unity3D to build a virtual assembly platform, build a dynamic gesture cognitive interaction system based on the CNN-LSTM algorithm, and use leap motion to obtain real-world hand command information and position information. The complete system structure is shown in Figure 1. It consisted of a virtual assembly platform and a dynamic visual recognition system. The communication thread of the entire platform used the TCP/IP protocol to ensure the stability of the interaction.

The main functional modules of the virtual assembly platform were composed of four parts. Part 1, vision controller: as a control interface, it could control the input and output of the viewing angle and other module parameters. Part 2, model library: we stored models of harvesting machinery, gestures, and parts. Part 3, assembly object controller: we controlled the behaviour of the parts operated by gestures based on mechanics and a dynamics engine. Part 4, virtual gesture controller: we perceived the gesture information and worked with the equipment object controller to complete the virtual assembly.

The main principle of the dynamic gesture recognition visual system was to use leap motion to perceive the posture and depth data of external gestures as input. After data input, we first used a CNN to extract feature vectors, constructed the feature vectors in a time series sequence, and used them as an LSTM network for input data. Then, we used an LSTM network for processing gesture classification to obtain gesture classification information. Then, the depth data was processed and turned into a mapping of the corresponding gesture position on the virtual assembly side. The above two kinds of information cooperated with the object controller and gesture controller of the virtual assembly platform to complete the virtual assembly process.

3. Construction of Dynamic Gesture Recognition System

3.1. Gesture Library Definition. The construction of the gesture library was the basis for gesture recognition and instruction judgement. The establishment of the gesture library included gesture data and gesture definition. The specific construction content is shown in Tables 1 and 2.

(1) Gesture Data. There were five kinds of data types in gesture data:

The data frame was the protocol data unit of the data link layer, which consisted of three parts: the frame header, a data part, and the end of the frame. Among them, the frame head and end contained some necessary control information, such as synchronisation information, address information, and error control information. We mainly recorded new dynamic data of gestures.

InteractionHand: This mainly recorded the number of hands participating in the interaction and shielded the data from the interfering hands. Its state was switched by the mode switch in Table 2.

InteractionHand Palm: This was the estimated value of palm width and length. Here, we used the anchor-based algorithm to introduce a number of prior values and approximate guesses.

InteractionHand IndexTip: We recorded the three states of the IndexTip: (Extended) straight, (Not Extended) bending, and (Either) merger.

InteractionHand GrabAngle: We recorded the degree of bending of all fingers. This was calculated by observing the direction of the four fingers and the folder between the directions of the fingers, as shown in Algorithm (3). When calculating the length, the thumb was not considered. The open palm was 0 arc of an open hand. When this position was a tight fist, it would reach the π arc.

(2) Gesture Definition. Gesture definition rule algorithm:

(1) Rotation:

$$\begin{cases} P_{LI}(x_{LI}, y_{LI}, z_{LI}), \\ P_{LT}(x_{LT}, y_{LT}, z_{LT}), \\ 0 \le \mathrm{Mag}_{P_{LI}-P_{LT}} \le m, \end{cases} \tag{1}$$

where P_{LI} is the left index finger coordinates, m; P_{LT} is the left thumb coordinates, m; and $\mathrm{Mag}_{P_{LI}-P_{LT}}$ is the Euclidean metric, m.

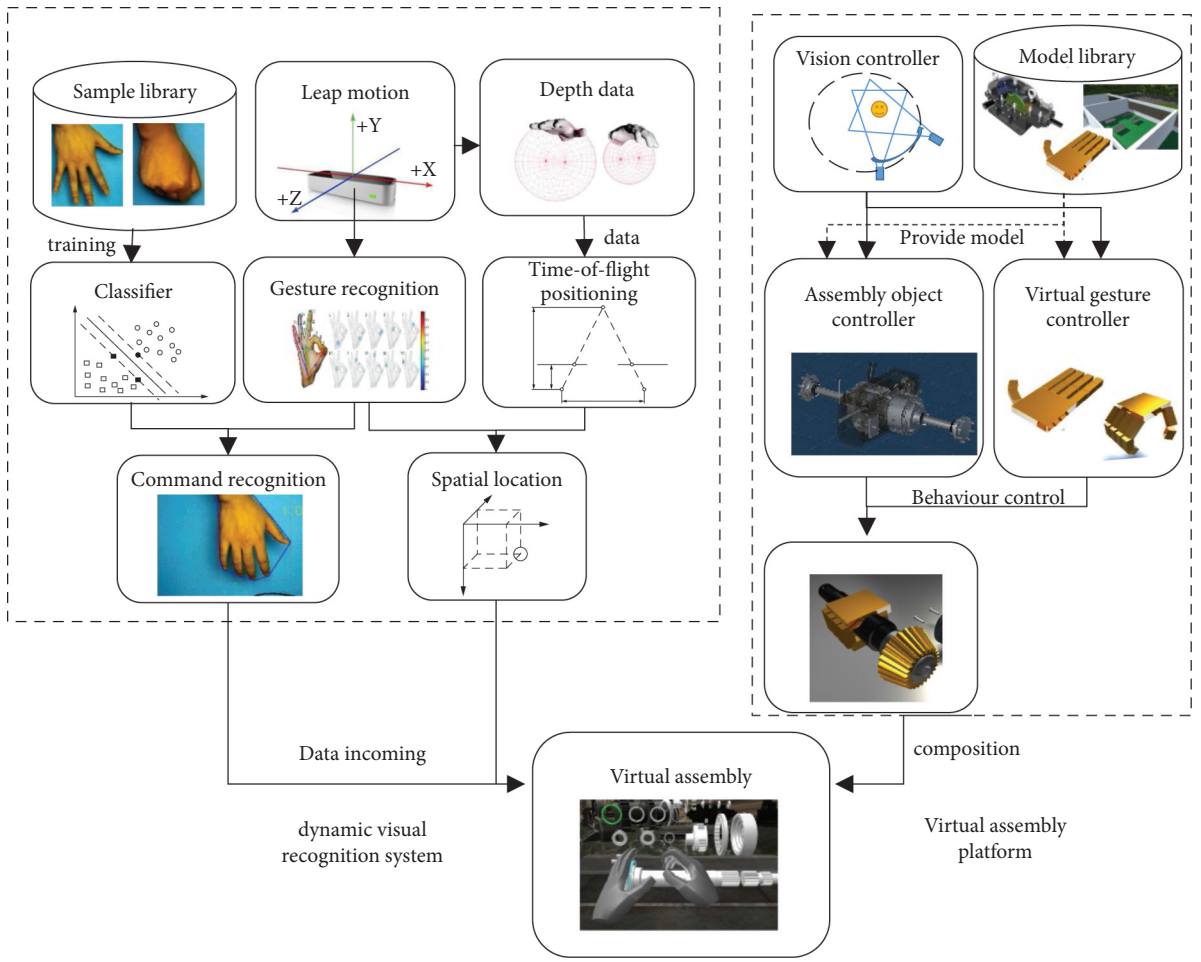


Figure 1: Architecture diagram of the virtual assembly system.

Table 1: Gesture data.

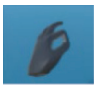
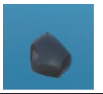
Data type
Frame
InteractionHand
InteractionHand Palm
InteractionHand IndexTip
InteractionHand GrabAngle

(2) Translation:

$$\begin{cases} P_{RV}\left(x_{RV}, y_{RV}, z_{RV}\right), \\ K, \\ T_0 = \begin{bmatrix} 1 & 0 & 0 & x_{RV} * K \\ 0 & 1 & 0 & y_{RV} * K \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \end{cases} \tag{2}$$

where P_{RV} is the right hand speed, m/frame; K is the translation amplitude parameter; and T_0 is the translation matrix, mm.

Table 2: Gesture definition.

Gesture type	Command signal	Gesture behaviour
One-hand	Rotation	
	Translation	
	Reset	
	Mode switch	
Hands	Perspective away	
	Perspective close	

(3) Mode switch:

$$\begin{cases} N_{RP}(x_{RP}, y_{RP}, z_{RP}), \\ y_{RP} \le 0, \\ G_{RP} \ge \pi, \end{cases} \tag{3}$$

where N_{RP} is the right palm vector, m, and G_{RP} is the angle between the index of the right hand and the palm vector, m.

(4) Hands gesture:

$$\begin{cases} P_{LI}(x_{LI}, y_{LI}, z_{LI}), \\ P_{LT}(x_{LT}, y_{LT}, z_{LT}), \\ 0 \le \mathrm{Mag}_{P_{LI}-P_{LT}} \le m, \end{cases} \quad \begin{cases} P_{RI}(x_{RI}, y_{RI}, z_{RI}), \\ P_{RT}(x_{RT}, y_{RT}, z_{RT}), \\ 0 \le \mathrm{Mag}_{P_{RI}-P_{RT}} \le m, \end{cases} \tag{4}$$

where P_{LI} is the left index finger coordinates, m; P_{LT} is the left thumb coordinates, m; P_{RI} is the right index finger coordinates, m; P_{RT} is the right thumb coordinates, m; and $\mathrm{Mag}_{P_{LI}-P_{LT}}$ and $\mathrm{Mag}_{P_{RI}-P_{RT}}$ are the Euclidean metric, m.

3.2. Dynamic Gesture Recognition System. Convolutional neural networks (CNNs) have characteristics such as a large amount of training model data and high memory usage of the training set. Long short-term memory (LSTM) network models could fully reflect the long-term historical process in the input time series data and could not mine the effective information and potential relationships between discontinuous data. Therefore, in this section, a CNN-LSTM network was built to achieve dynamic gesture recognition, a CNN spatial dimension feature learning, and an LSTM that was responsible for time dimension feature learning.

The construction of the dynamic gesture recognition visual system was mainly divided into two steps: defining gesture commands and constructing a CNN-LSTM classifier. The process of recognising gestures is shown in Figure 2. It captures the behaviours corresponding to gesture commands in the actual environment and takes position, rotation, and index as feature values. After the data were converted, they were input into the trained classifier. The actual gestures and the defined gestures are obtained by the classifier at the same time. A defined classification probability between gestures, the largest classification probability gesture, was selected to interpret the actual gesture commands, and the actual gesture command recognition was completed. Because the recognition algorithm took less than 36 ms, it ensured the fluency of recognition and could be used for real-time and dynamic gesture recognition.

3.2.1. Gesture Command Definition. After analysis [21, 22], gesture behaviours were mostly expressed by index finger depiction, such as writing and drawing. Capturing index finger behaviour required collecting the following feature values: three components of position, three components of rotation angle, and three components of an index finger, as shown in Figure 3.

To strengthen the classification effect and to enhance the characteristics of gesture commands, due to the obvious feature changes between Arabic numerals, the form of depicting Arabic numerals by the index finger was selected as the gesture command. The expression of the gesture command was stipulated as follows: within the specified unit time, the right index finger kept repeating a certain number, and the system recorded the characteristic value of the right index finger. The three-dimensional visualisation of the index finger feature values of some samples is shown in Figure 4.

To facilitate data analysis, the visualisation of gesture commands was needed. The captured index finger depicted the characteristic values of numbers 0–9, as shown in a line graph in Figure 5, where from left to right the gesture labels of numbers 0–9 could be seen. From top to bottom, there were three components: position, rotation angle, and index.

Through Figure 5, obvious differences between the gesture commands, indicating that the gesture commands used the index finger to draw Arabic numerals, had a strong degree of discrimination, strengthened the classification effect, and were beneficial to building a classifier with better performance and higher robustness.

3.2.2. Construction of CNN-LSTM Classifier. Dynamic gesture recognition is a classification task based on time series. There are traditional methods such as HMM and DTW for time series processing methods, but the traditional methods are more complicated and often require professional feature engineering processing work. In contrast, the deep learning method could avoid tedious feature engineering. Considering that the differences between digital gestures often existed in multiple local features, we could use a CNN to extract time series features and then use the extracted features as input to an LSTM to classify gestures and form a convolutional timing network (CNN-LSTM). The network structure is shown in Figure 6. The network mainly included 2 convolutional layers, 1 pooling layer, and 1 LSTM layer.

Training in the CNN-LSTM network: After the picture with the hand motion information was processed by the input layer, the pixel information with the picture was formed and input into the convolutional layer, and the different features of the input image were extracted through the convolutional layer. For such input data, the first layer of the convolutional layer may have extracted only some low-level features, such as edges, lines, and corners, and more layers of the network iteratively extracted more complex features from the low-level features. Therefore, two convolutional layers were used for feature extraction here.

For the data volume of the image after the convolutional layer was directly used for network training and prediction, the calculation amount was too large, and the phenomenon of overfitting was prone to occur. To reduce the amount of

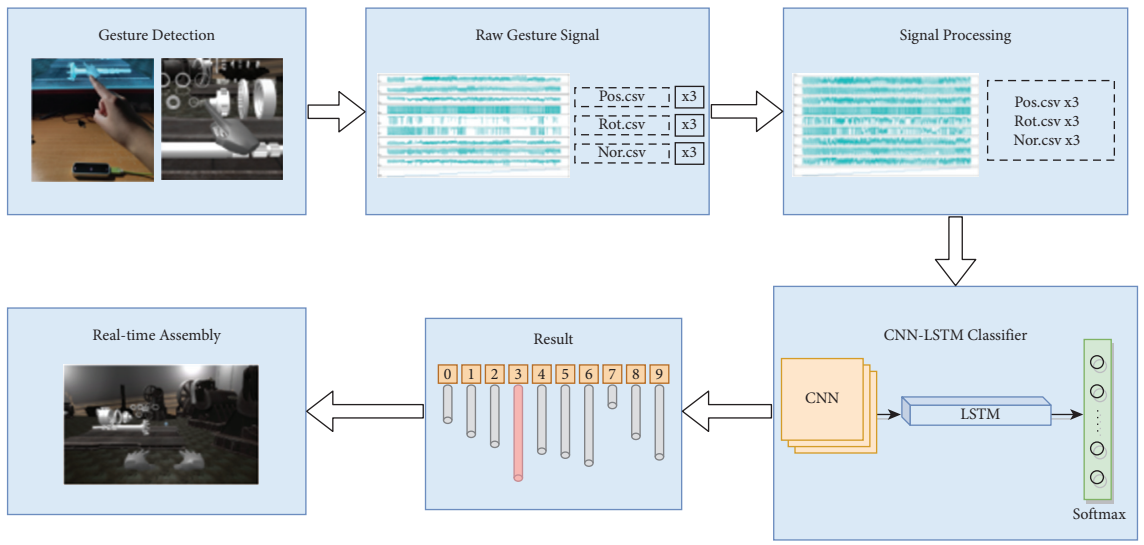


Figure 2: Flow chart of the dynamic gesture recognition vision system.

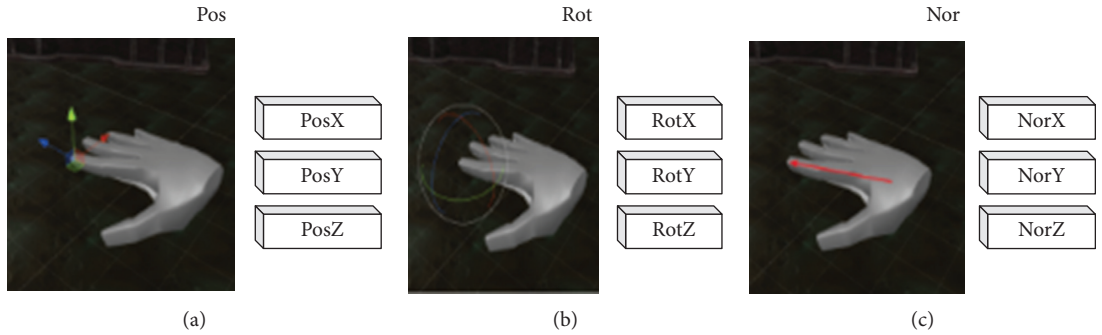


Figure 3: The characteristic values needed to capture the index finger. (a) Position feature. (b) Rotation angle feature. (c) Index pointing characteristic.

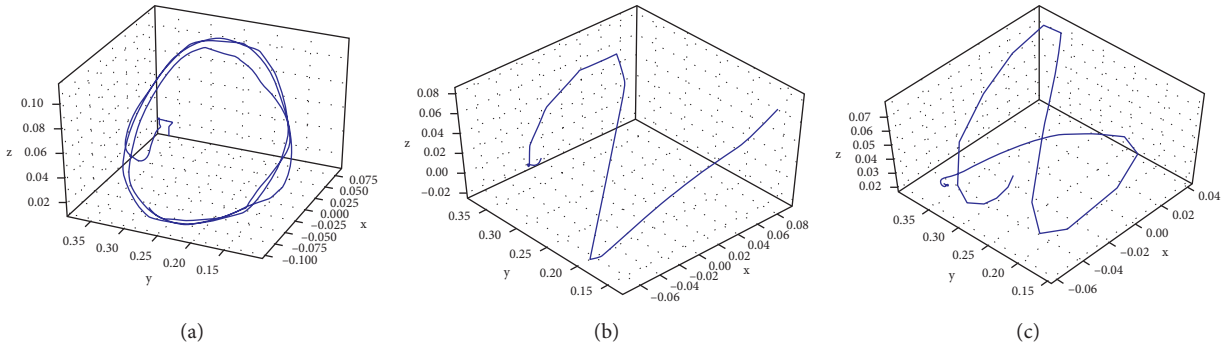


Figure 4: Three-dimensional visualisation of index finger eigenvalues.

calculation and suppress the overfitting phenomenon, the pooling process was performed, and the downsampling operation was performed while retaining the important original data on the feature map, thereby reducing the size of the feature map.

After the pooling process, an LSTM was used to classify the recognised gesture features. The neuron structure diagram of an LSTM is shown in Figure 7.

The figure refers to the implementation details of the hidden layer. The input and output of this layer were x_t and h_t, respectively, and the memory unit was c_t.

The input gate was used to control the value of the current input data x_t flowing into the memory unit, which could be expressed as

$$i_t = \delta\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right). \tag{5}$$

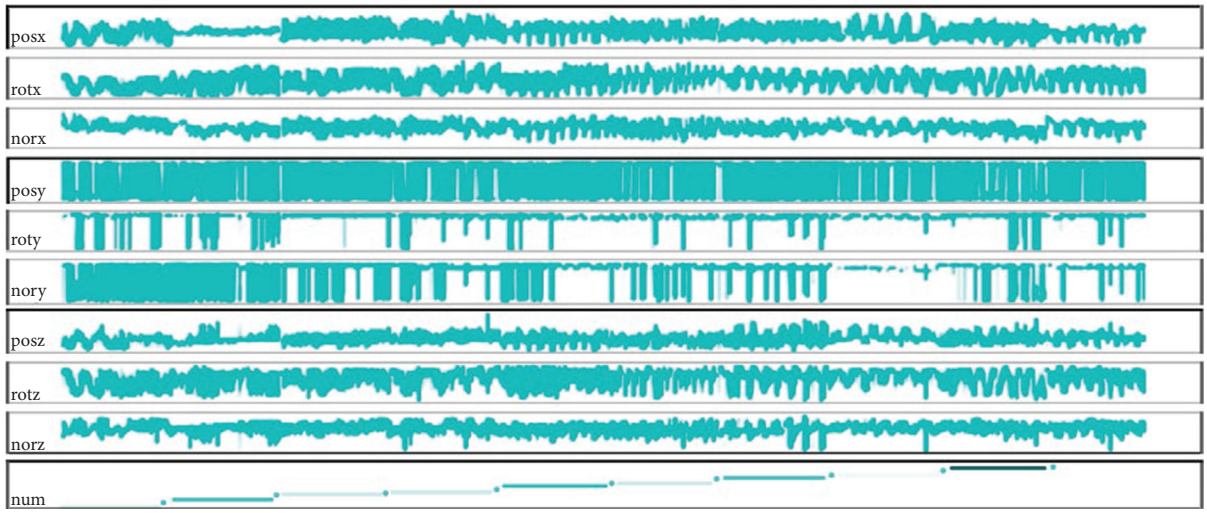


Figure 5: Comparison of eigenvalues drawn by the index finger from 0 to 9.

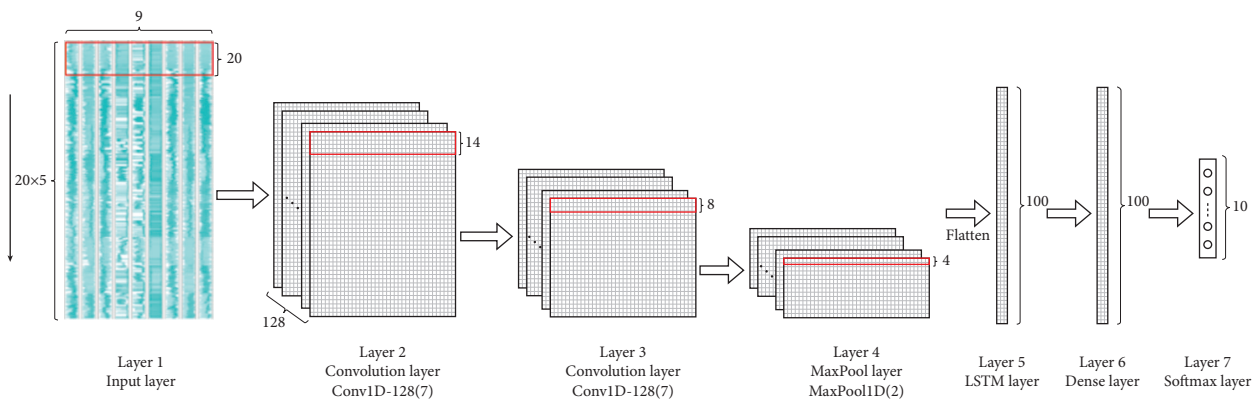


Figure 6: Schematic diagram of CNN-LSTM network structure.

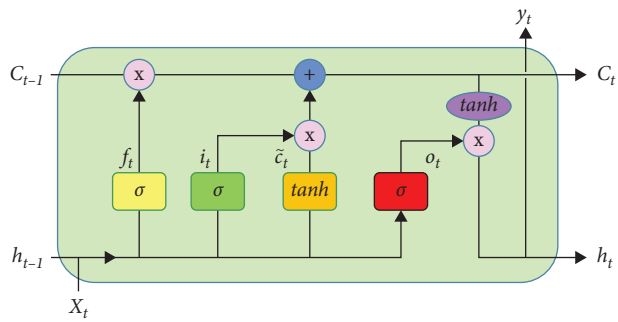


Figure 7: Neuron structure diagram of an LSTM.

The forgetting gate controlled the retention and forgetting of information. The influence of the information in the memory unit c_{t-1} at the previous moment on the current memory unit c_t avoided the disappearance and explosion of the gradient caused when the gradient propagates back over time, which could be expressed as

$$f_t = \delta\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right),$$
$$c_t = f_t\theta c_{t-1} + i_t\theta \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right). \quad (6)$$

The influence of the output gate control memory unit c_t on the current output value h_t, also referring to which part would be output, could be expressed as

$$o_t = \delta\left(W_{xo}x_t + W_{ho}h_{t-1} + b_o\right),$$
$$h_t = o_t\theta \tanh\left(c_t\right). \quad (7)$$

4. Virtual Assembly Platform

The virtual assembly platform contained four core modules, which were developed based on the Unity3D engine, namely, 1. viewing the angle controller, 2. the model library, 3. the assembly object controller, and 4. the virtual gesture controller, as shown in Figure 8. For specific functions, see Section 1.

After the model library was constructed, the bridge and the gearbox were the target objects, which were modelled by SolidWorks, formatted by PiXYZ Studio and imported into Unity3D. This section specifically introduced the perspective controller adapted to the virtual assembly platform and the virtual assembly object controller (virtual hand could be regarded as a special virtual assembly object).

5. Perspective Controller

In this system, in addition to the common first-person arbitrary roaming perspective, to ensure that the actual interaction of the assembly system was smooth, a synchronised perspective based on the synchronisation point of both hands and a perspective change guided by a special gesture operation should also be included.

The unified formula for synchronous perspective and perspective transformation was

$$\begin{cases} T_0 = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ T_1 = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ \left(x_0 = 0, y_0 = 0.408, z_0 = 0.457\right), \\ \left(\varphi_{x_0} = 11^\circ, \varphi_{y_0} = 0^\circ, \varphi_{z_0} = 0^\circ\right), \end{cases} \quad (8)$$

where x_0, y_0, z_0 are the camera local coordinates (relative to leap motion); m, φ_{x_0}, φ_{y_0}, φ_{z_0} are the camera partial angles (relative to leap motion); T_0 is the camera translation matrix, mm; T_1 is the camera rotation matrix around the y axis; and T_2 is the camera rotation matrix around the x axis.

The perspective controller activated the corresponding rotation matrix after sensing the gesture command and used the rotation matrix to complete the transformation of the camera, thereby bringing about the transformation effect of the perspective. Considering the actual assembly operation, the first was used as the trigger gesture of the viewing angle movement command, and the position change of the hands was used as the operation map. The effect of the change is shown in Figure 9.

5.1. Virtual Assembly and Virtual Hand Controller. This part mainly completed the logical control of virtual assembly, and this part was composed of assembly logic and interference detection of trigger logic.

Interference detection was a computer application technology that detected whether there was interference or penetration between objects in a virtual environment. It was mainly determined by the positional relationship of the triangles that constituted the model. The position of the triangle overlaps represented interference. Start logic was detected. Accurate triangle patch detection would have improved the accuracy of interference detection, but it would have also increased the large amount of calculation. Therefore, this paper used hierarchical transmission to construct six different levels of bounding boxes to achieve interference detection, as shown in Figure 10.

For models that do not have key attributes, we used a, b, and c, three bounding boxes to fit the shape according to the specific shape; used mesh rows to fit the entire model for key parts, such as gearboxes; and included models that emphasized mechanical properties, such as wheels. Extremum slip, extremum value, asymptote slip, and asymptote value wheel collider were used for interference detection to reduce the amount of calculation while maintaining accuracy. The fitting effect of the assembly process is shown in Figure 11.

After analysing the virtual assembly requirements, the interaction rules between the virtual hand and the target part are shown in Figure 12. The virtual assembly platform obtained hand position information and hand command information. According to the hand position information, the virtual hand was generated at the corresponding position in the virtual environment. The virtual hand was controlled to shoot rays according to the hand command information. If the command was recognised as an open hand, the virtual hand did not emit or stop the ray. If the command was recognised as a fist, the virtual hand generated a ray; when the ray intersected the corresponding part, it could be based on the hand. The part position information changed the position of the corresponding part to realise the function of grabbing and moving the part; after grabbing the part, when the command was recognised as the open hand, the virtual hand stopped generating rays, disconnected from the part, and judged whether the part had reached the installation. If it reached the installation position, the part moved to the

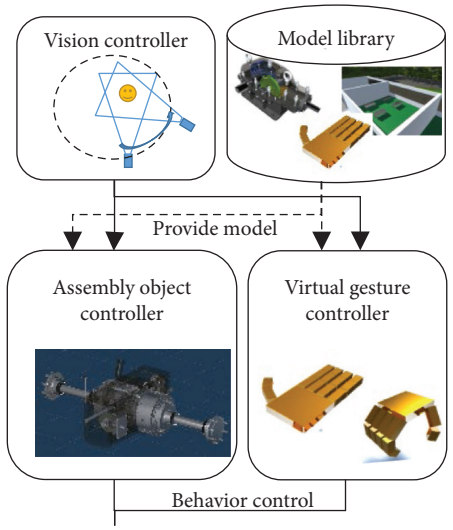


Figure 8: The four core modules of the virtual assembly platform.

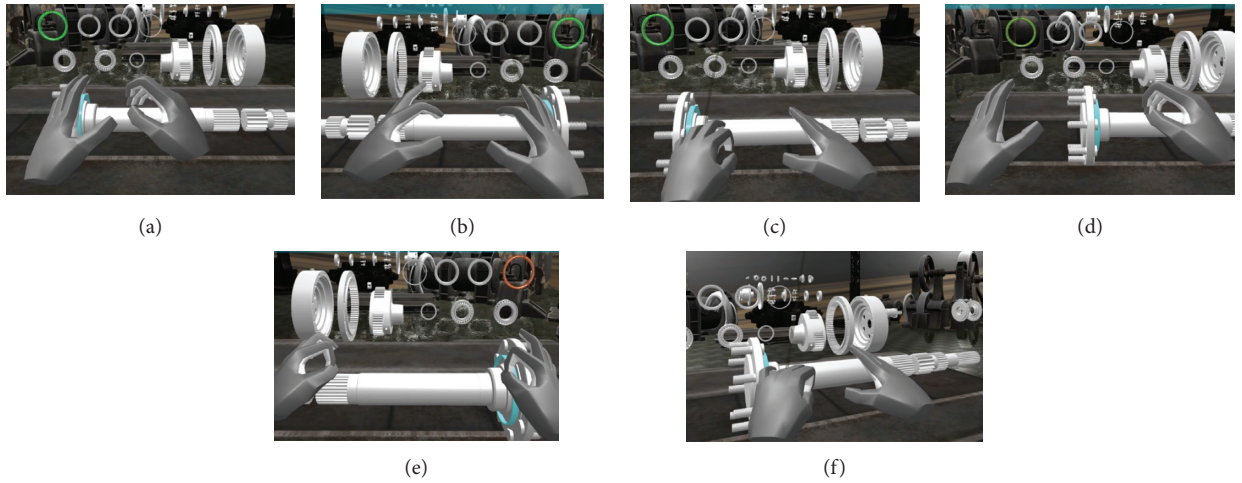


Figure 9: Perspective change guided by gestures. (a) Original perspective 1. (b) Original perspective 2. (c) Original perspective 3. (d) Move left 1. (e) Move back 2. (f) Rotating angle of view.

installation position; when the part coincided with the installation position, we judged whether the part is installed correctly and gave a prompt.

6. Experiment and Analysis

6.1. CNN-LSTM Gesture Recognition Algorithm Performance Analysis

6.1.1. CNN and LSTM Network Performance Analysis. To analyse the performance of the deep learning method, the machine learning algorithm was used for gesture recognition. Among the three algorithms of RF, SVM, and KNN, the accuracy of random forest was the highest, reaching 78%, which was used as the accuracy baseline. We optimised the key parameter filter and kernel function of the initial CNN network and determined the optimal number of filters as 64, the size of the optimal kernel function was 10, and its test accuracy was 96.6% (±1.200). The time was 1 minute and 15 seconds.

When the LSTM parameter "maximum epochs" was set to 200, the accuracy could reach 99%, and the time used was 4 minutes and 47 seconds; when the number of maximum epochs was 100, the test accuracy was 92.2% (±3.763), and the time used was 2 minutes and 32 seconds. Although LSTM had a high accuracy, it took a long time and was difficult to use for network training on large datasets.

6.1.2. CNN-LSTM Performance Analysis. In separate experiments on the CNN and LSTM networks, all sample points of a single sample were exposed to the network, and one-time weight training was performed. The learned features were based on global considerations. However, real gestures were often based on differences between multiple local features and other gestures, which ultimately formed a

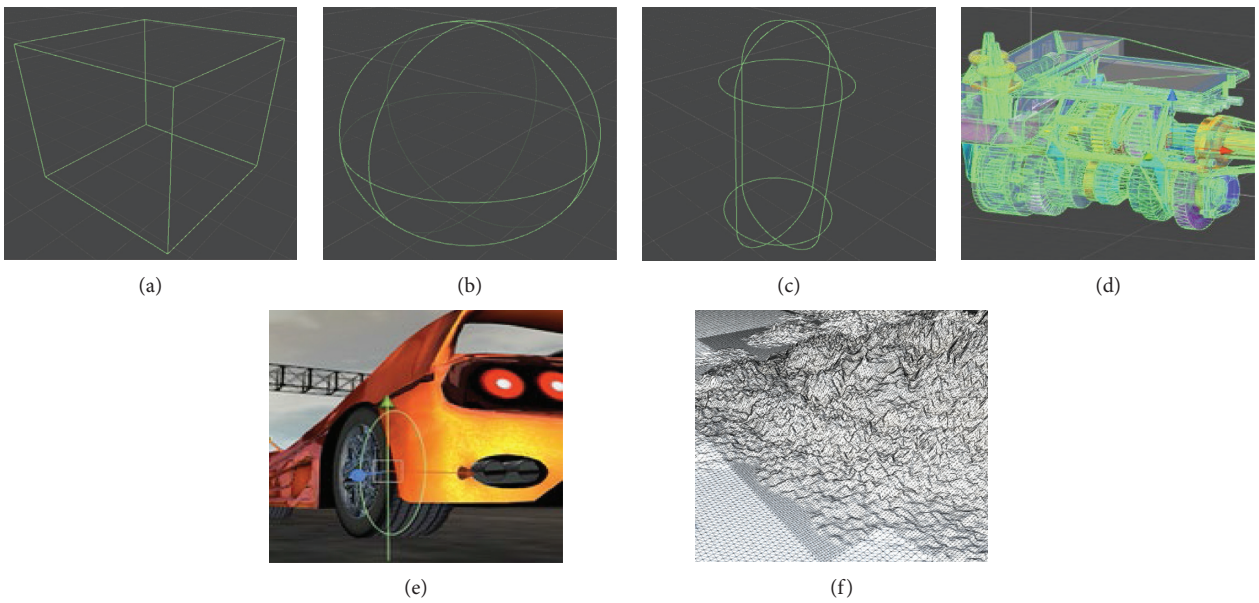


Figure 10: Bounding boxes for different objects. (a) Box collider. (b) Sphere collider. (c) Capsule collider. (d) Mesh collider. (e) Wheel collider. (f) Terrain collider.

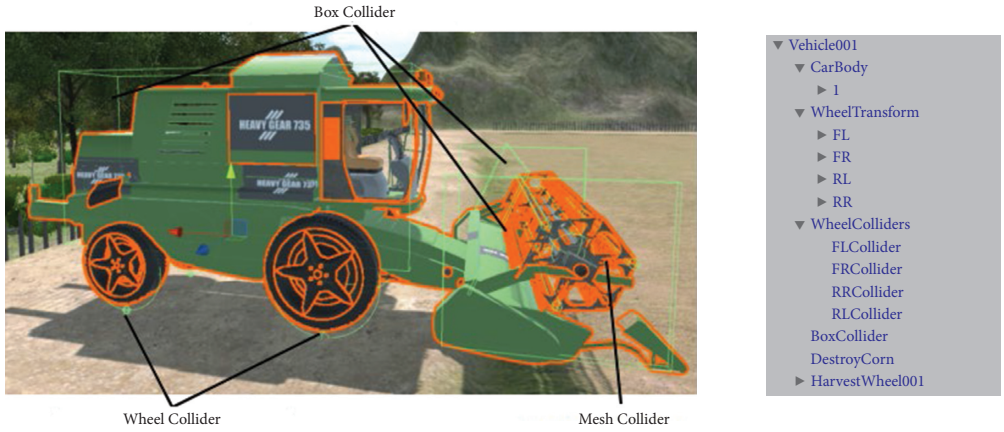


Figure 11: Bounding boxes in the assembly of the whole machine.

fundamental difference. The CNN-LSTM network could learn based on time series; we segmented the sample points and finally applied local features to single-sample gesture recognition. To obtain a good CNN-LSTM network and optimise the structural parameters of the network, three experiments were designed: the best time sequence test, the F/K parameter selection test, and the best network performance test.

(1) The best time sequence test. To select the initial parameters of the CNN-LSTM network, we performed multisequence segment detection experiments and selected the best sequence segment according to the accuracy range of different sequence segments.

The data structure of the CNN-LSTM network could be expressed as $m * t * b * n$, where m is the number of samples, t is the number of time steps, b is the length of the time single step, and n is the type of eigenvalue. The best sequence segment test mainly involved the parameters filter, kernel, b, and t. T is a single-sample sequence segment with 1000 sample points, and the sequence segment was divided into 6 types, as shown in Table 3. The results of the test are shown in Figure 13.

From the analysis of Figure 13, we could see that from the perspective of amplitude change, the best time sequence segment was $t = 5$, $b = 20$, and its accuracy distribution was very tight.

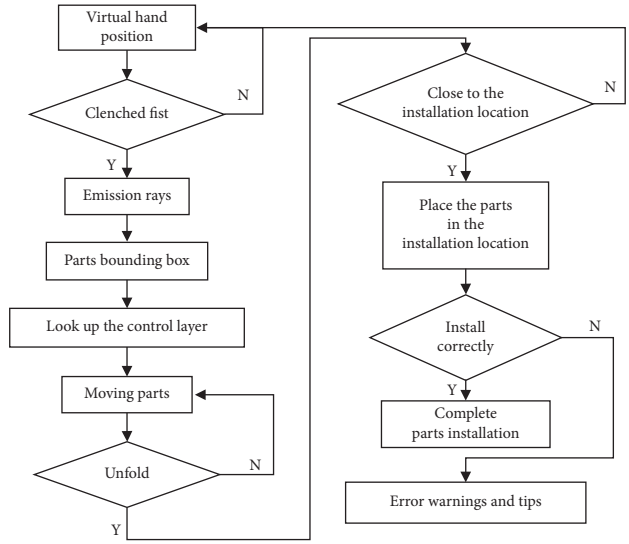


Figure 12: Interactive rules of virtual assembly.

Table 3: The segmentation status table of the time series.

Num	Timing period	
	t	b
1	1	100
2	2	50
3	4	25
4	5	20
5	10	10
6	20	5

(2) F/K parameter selection test. By observing the K/F-Acc table of the best time sequence ($t = 5$, $b = 20$; see Table 4), we selected the best filter and kernel through comparison to determine the F/K parameters.

Through the horizontal and vertical comparison of Table 4, when filter = 64, 128, 256 and kernel = 3, 5, 7, the results were similar. Considering the fluctuation of accuracy and numerical value, kernel = 7 was the most stable because the larger the filter was, the greater the number of parameters. Thus, filter = 128 was chosen as a compromise.

(3) Best network performance test. After selecting other parameters, optimised network training was performed. The network structure parameters were shown in Table 5. The model accuracy and standard deviation were obtained from the experiment, and the training time was recorded. The result is shown in Figure 14.

According to the analysis in Figure 14, the test accuracy was 98.0% (±0.894), and the time used was 3 minutes and 06 seconds. The accuracy of the model was higher than that of the CNN network, the number of parameters was approximately 7 times that of the LSTM, and the time was only approximately 30 s longer than that. This network could integrate the advantages of the two types of networks and had a compromise between accuracy and time consumption. The overall performance was better than the CNN and the LSTM single networks.

The optimised network training process is shown in Figure 15. The accuracy was approximately 95%, the loss was approximately 0.25, and the trained model size was only 6.32.

6.1.3. Performance Test Analysis of the Gesture Recognition Algorithm. We explored whether the real-time detection accuracy of the gestures met the standard and conducted real-time detection experiments to record the classification of the gestures. We generated a confusion matrix based on the result of the gesture classification and observed the real-time classification of each gesture, as shown in Figure 16.

From the analysis of Figure 16, it could be seen that, except for gestures 9 and 7, the classification accuracy of the other gestures was relatively high, which met the accuracy requirements of the gesture recognition model and could be applied to real-time gesture classification.

At the same time, through this experiment, we proved that when data was collected, there was approximately 0.5 s of file input and output processing. Thus, there was an intermittent reduction in frame rate, which was reduced to approximately 20 FPS. When the data collection was completed, the frame rate immediately returned to normal. The time consumption of each frame was maintained at

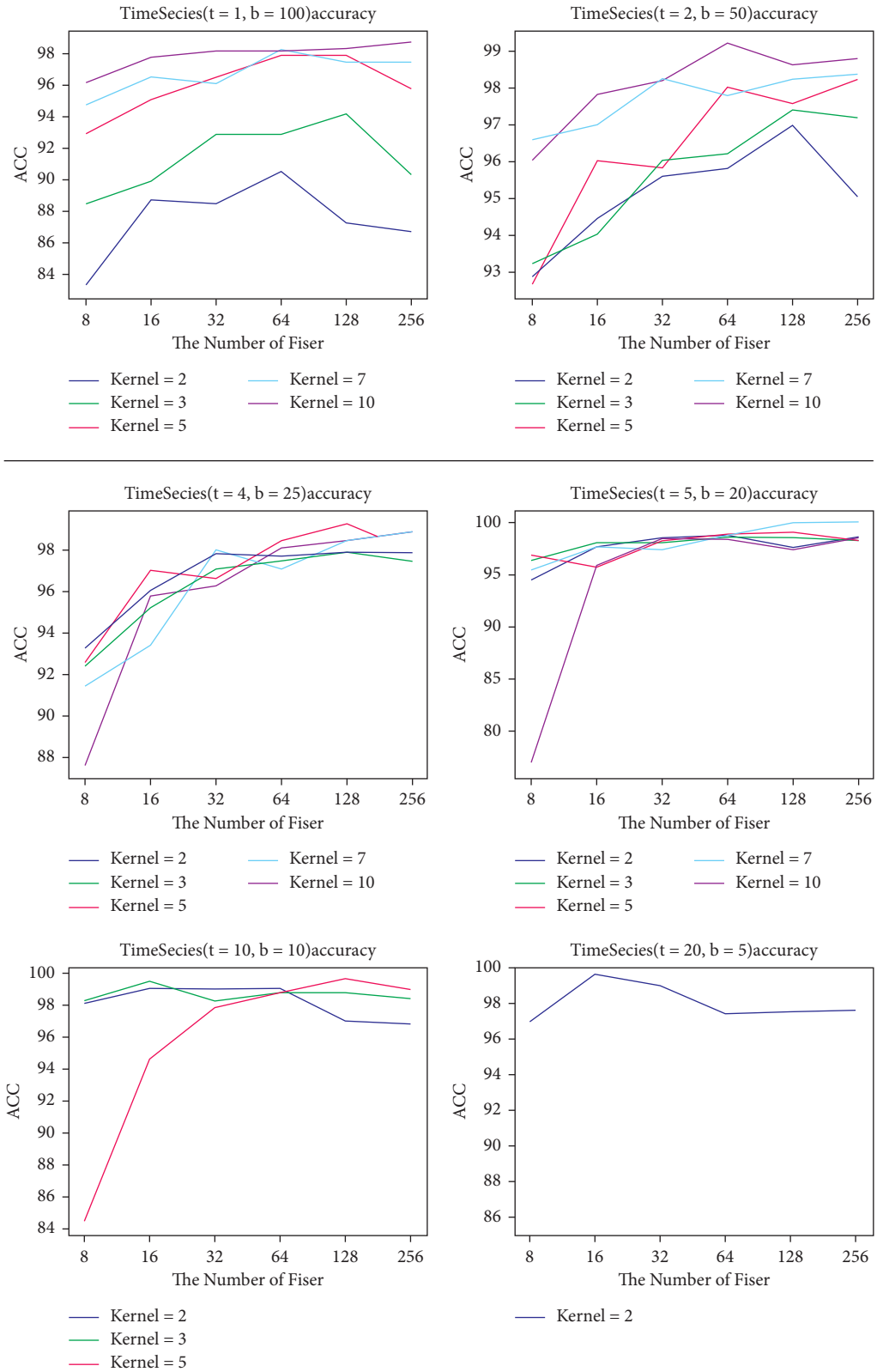


Figure 13: The best time sequence segment test data line graph.

approximately 2 ms, during which there was no communication blocking phenomenon. This result showed that gesture recognition was stable.

6.2. Whole Machine Analysis of the Assembly Effect of the Virtual Assembly System. The assembly time and records are shown in Table 6. In general, the optimised system increased

Table 4: Time series K/F-Acc table.

Kernel	Filter					
	8	16	32	64	128	256
2	94.4 (±3.200)	97.6 (±0.490)	98.4 (±1.020)	98.6 (±1.020)	97.6 (±1.744)	98.6 (±0.800)
3	96.4 (±1.356)	98.0 (±1.265)	98.0 (±1.095)	98.6 (±1.020)	98.6 (±1.497)	98.2 (±0.980)
5	96.8 (±2.040)	95.8 (±3.059)	98.0 (±1.673)	98.8 (±1.166)	99.0 (±0.894)	98.4 (±0.800)
7	95.4 (±2.417)	97.4 (±1.625)	97.4 (±0.490)	98.4 (±0.800)	99.8 (±0.400)	100.0 (±0.000)
10	76.6 (±8.913)	95.8 (±3.709)	98.4 (±1.200)	98.4 (±1.497)	97.4 (±2.417)	98.4 (±1.200)

Table 5: CNN-LSTM network structure parameters.

CNN-LSTM configuration	
6 weight layers	
Input (1000∗5∗20∗9 TimeSeries)	
Conv1D-128 (7)	
Conv1D-128 (7)	
MaxPool1D (2)	
Flatten	
LSTM (100)	
FC-100	
Softmax (10)	
Parameter	Value
Maximum epochs	100
Batch_size	32
Optimiser	Adam
Metrics	Acc & loss
Min–max normalisation	Yes
The number of parameters	380K
Method	Epochs = 5, mean/std

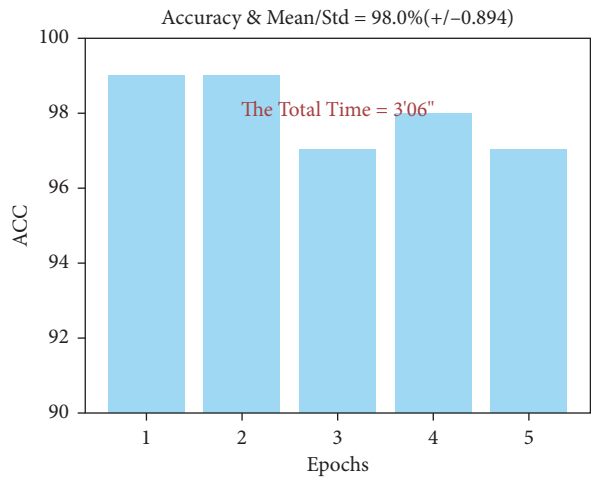


Figure 14: Results.

the efficiency by approximately 15% on the basis of the original system. According to the actual assembly experiment, it was clear that the optimised system used the network model to perform gesture detection, quickly reacted, and recognised the gestures again, and the recognition feedback speed was very fast. Even in the case of gesture recognition errors, it immediately re-recognised the gestures. This was not available in the original system. Since the original system selected parts through the indicator rod, when the part was not selected due to unintentional hand shaking, it was necessary to reset the timing to select the part, which led to a waste of time and a lower assembly efficiency. Therefore, the current system quickly responded to gestures and performed efficient assembly with the original gesture library.

In the average time-consuming index per frame, the two systems were not much different; both met the assembly requirements and performed the assembly operations smoothly.

6.3. Overall Performance Test of the Agricultural Machinery Virtual Assembly System to Test Agricultural Climbing. The behavioural expression of virtual objects was composed of behavioural models and cognitive models. The behaviour model was the direct response of the virtual object to external changes, and the simulated behaviour conformed to the basic motion and behavioural rules in the real world. The cognitive model was a description of the virtual character's analysis of the acquired information and the execution of the decision-making process. To verify the practicability of the virtual assembly system, the assembled corn harvester was imported into the virtual simulation experiment system, and the corn harvesting experiment was carried out to observe the experimental effect of the whole machine. The specific experimental steps are shown in Figure 17.

With the gearbox as the only power output, the parts completed a series of actions under the action mechanics and finally completed the harvesting behaviour. This virtual simulation experiment could only be completed if the virtual assembly model met the requirements. At the same time, the complete simulation behaviour was also of great importance to the actual harvest. The experimental results are as follows.

The observation and analysis of the operation behaviour are shown in Figure 18. The operation of the crop object and the harvesting platform in the simulation conformed to the harvesting operation process: First, after starting the harvesting platform, the reel rolled, restricted, and guided the movement of the crop when it touched the crop. As shown in Figure 18(a), when the cutting blade of the header touched the crop object, the root of the crop object separated from the main body. The crop body was entangled by the reel and guided into the header, and then the auger pushed the harvest port of the crop object main body, as shown in Figure 18(b). Finally, the crops sent to the harvesting port were eliminated, and the harvest data was counted at the same time, as shown in Figure 18(c). The normal harvest state is shown in Figure 18(d), and the crops that had been sent to the header were transported to the harvest. At the

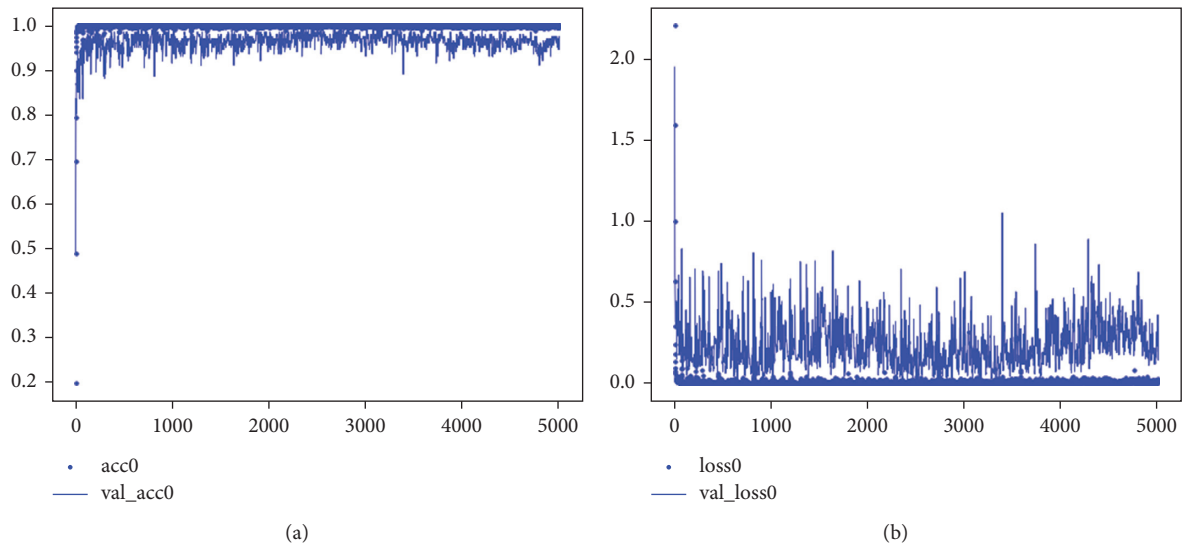


Figure 15: Loss change diagram of the network training process. (a) Training and validation accuracy. (b) Training and validation loss.

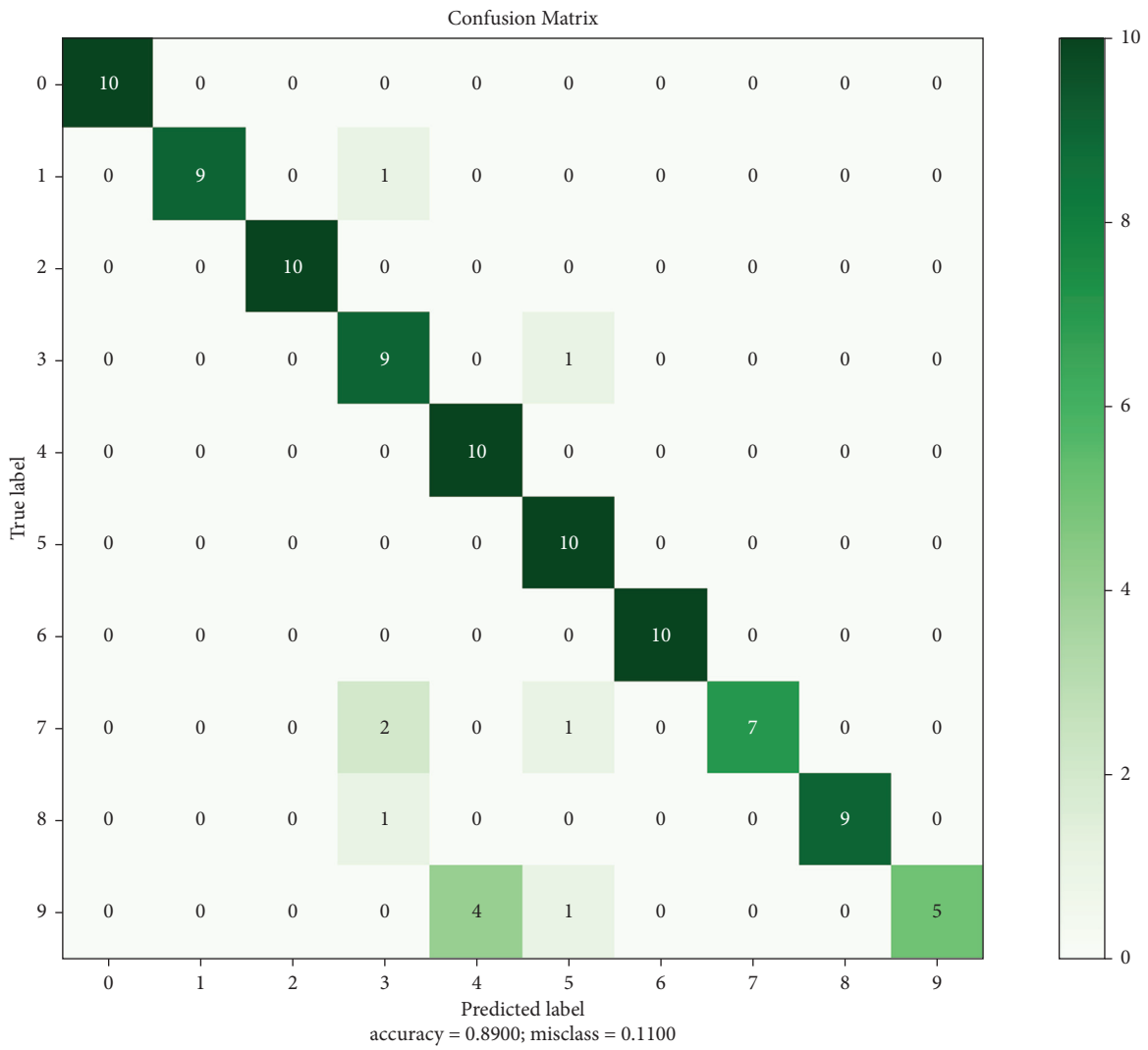


Figure 16: Real-time classification of gestures.

Table 6: Assembly time-consuming statistics.

Serial number	Time-consuming assembly (min)		Average time per frame (ms)		Efficiency improvement (%)
	Original system	Existing system	Original system	Existing system	
1	2′10″	1′49″	5.3	5.4	16.1
2	2′05	1′52″	4.9	5.5	10.4
3	2′08′	1′48″	5.1	5.2	15.6
4	2′14″	1′55″	5.0	5.5	14.2
5	2′12″	1′52″	5.3	5.1	15.1

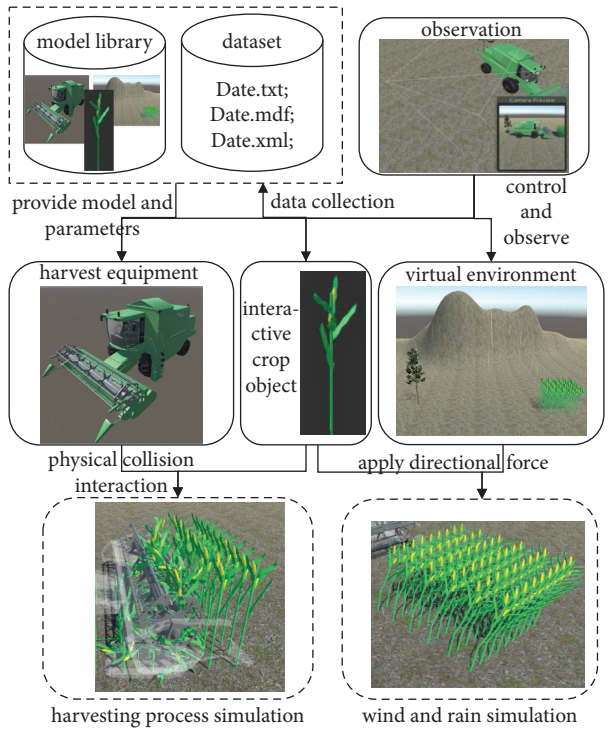


Figure 17: The idea of the whole machine harvesting experiment.

same time, the harvesting platform and reel continued to send new crops to the harvesting platform. If the harvesting equipment travelled at a too high speed, the crops would be thrown away, as shown in Figure 18(e). During the harvest, there was fruit leakage. The missing section is shown in Figure 18(f). The observation results showed that the operation simulation platform met the requirements of operation simulation and could be used for the evaluation test of the operation behaviour and performance of harvesting equipment. Therefore, the virtual assembly platform was of great importance to complete the virtual simulation.

6.4. Harvest Crop Loss Experiment at Different Driving Speeds. The harvest performance test was carried out, the fruit harvest data were counted, and the harvest operation simulation tests of the travel speed of the harvesting equipment and the under-yield rate were carried out under different crop distribution densities. The test results are shown in Figure 19. Figure 19 shows that under the same crop distribution density, the overall yield showed a downward trend. As the travelling speed increased, the under-yield rate decreased rapidly. When the speed exceeded 0.39 m/s, the downward trend of the under-yield rate became slower. However, as the distribution density of the crops increased, the downward trend of the corresponding under-yield rate also became slower. The results of the simulation experiment were consistent with the conclusions of the relevant documents of harvesting operations [23, 24]. Using the data to perform variance analysis, the result showed that both the travel speed and the crop density had a substantial impact on the harvesting performance of the harvesting equipment ($P < 0.01$) [25].

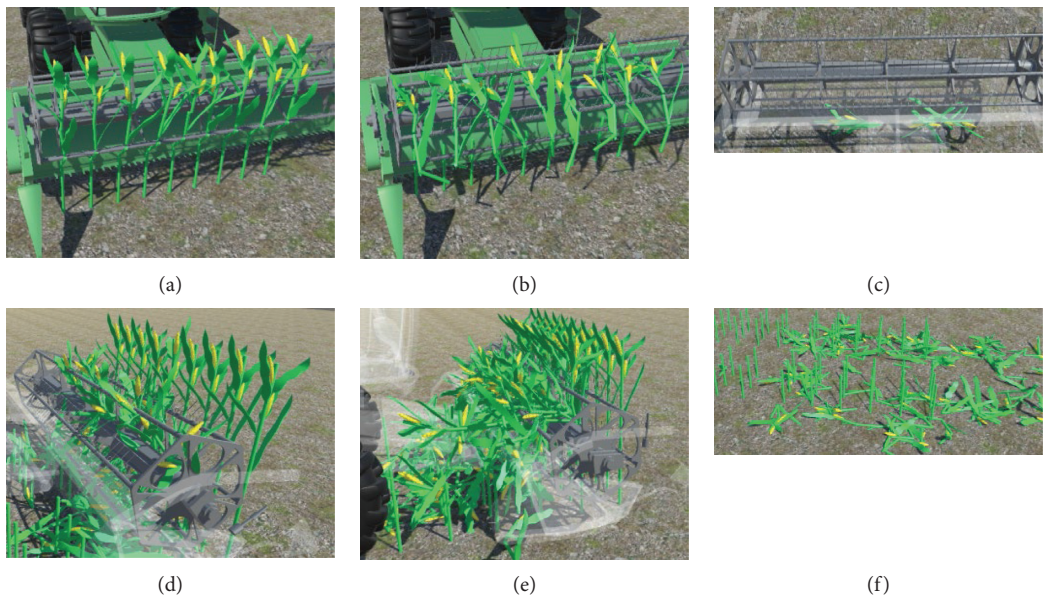


Figure 18: The idea of the whole machine harvesting experiment. (a) Restricted crops. (b) After cutting the crop. (c) Eliminated crops. (d) Normal harvest. (e) Crop drawn in and thrown out. (f) Missing knots.

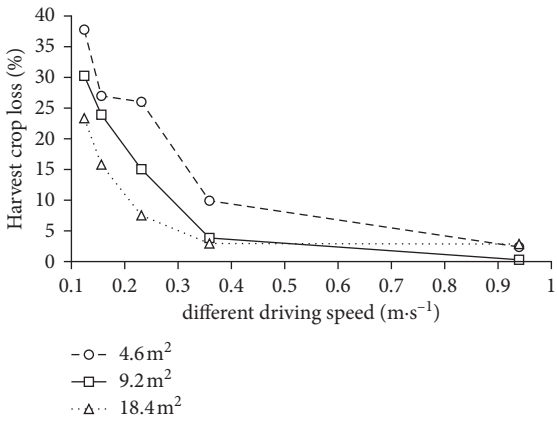


Figure 19: Harvest crop loss at different driving speeds.

7. Conclusions

(1) Through the machine learning algorithm experiment, an accuracy baseline of 78% was selected. In the experiment using the deep learning algorithm, the accuracy of all of the networks was greater than the accuracy baseline, which proved that the deep learning algorithm had advantages in establishing a gesture recognition model.

(2) After experimentation, the CNN-LSTM proposed in this paper could combine the advantages of the CNN and LSTM to quickly establish a stable and accurate gesture recognition model, which realised the initial extraction of time series feature segments through the CNN, reduced the burden on the LSTM network, and improved the time series of the LSTM classification efficiency.

(3) After testing, the accuracy and stability of the model met the requirements.

(4) After the whole machine harvesting experiment, the virtual assembly system model could be used by the virtual simulation system. This system was very important for the agricultural machinery assembly and simulation experiments.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2017YFD0700100) and International Cooperation in Science and Technology by Department of Science and Technology of Guangdong Province (2019A050510035).

Supplementary Materials

All the supplementary materials have been provided, including software systems, high-definition images, codes, and all other files. (*Supplementary Materials*)

References

[1] Z. Zhai, Y. Du, Z. Zhu, J. Lang, and E. Mao, "Three-dimensional reconstruction method of farmland scene based on rank transformation," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 31, no. 20, pp. 157–164, 2015.

[2] F. Wang, *Research on Test Method of Self-Propelled Agricultural Machinery Based on Virtual Reality*, China Agricultural University, Beijing, China, 2014.

[3] M. Karkee, B. L. Steward, A. G. Kelkar, and Z. T. Kemp, "Modeling and real-time simulation architectures for virtual prototyping of off-road vehicles," *Virtual Reality*, vol. 15, no. 1, pp. 83–96, 2011.

[4] K. M. Sagayam, A. J. Timothy, C. C. Ho, L. E. Henesey, and R. Bestak, "Augmented reality-based solar system for e-magazine with 3-D audio effect," *International Journal of Simulation and Process Modelling*, vol. 15, no. 6, pp. 524–534, 2021.

[5] S. Kaliyaperumal, M. H. Abd Wahab, K. M. Sagayam, R. Ambar, and H. Mhd Poad, "Impact of pairing an augmented reality demonstration with online video lectures… does it improve students' performance?… does it improve students' performance?" *Asian Journal of University Education*, vol. 16, no. 4, pp. 91–98, 2021.

[6] S. Inoue, T. Ojika, M. Harayama, T. Kobayashi, and T. Imai, "Cooperated operation of plural hand-robots for automatic harvest system," *Mathematics and Computers in Simulation*, vol. 41, no. 3-4, pp. 357–365, 1996.

[7] K. Melemez, G. Di Gironimo, G. Esposito, and A. Lanzotti, "Concept design in virtual reality of a forestry trailer using a QFD-TRIZ based approach," *Turkish Journal of Agriculture and Forestry*, vol. 37, no. 6, pp. 789–801, 2013.

[8] Y. Zang, Z. Zhu, Z. Song, W. Meng, H. Bo, and M. Enrong, "Establishment of virtual experiment system platform for agricultural equipment," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 41, no. 9, pp. 70–74, 2010.

[9] Y. Yuan, X. Zhang, C. Wu, J. Li, and L. Bo, "Interaction control system of agricultural machinery virtual test," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 42, no. 8, pp. 149–153, 2011.

[10] L. Luo, X. Zou, T. Cheng, Z. Yang, C. Zhang, and Y. Mo, "Design of virtual test system based on hardware-in-loop for picking robot vision localization and behavior control," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 33, no. 4, pp. 39–46, 2017.

[11] Z. Lu, S. Qin, L. Li, and D. Zhang, "Classification and recognition of first-view gesture expression in intelligent human-computer interaction," *Automation Equipment*, vol. 47, no. 6, pp. 1284–1301, 2021.

[12] Y. Miao, J. Li, and S. Sun, "Dynamic gesture recognition combining global gesture motion and local finger motion," *Journal of Computer-Aided Design & Computer Graphics*, vol. 32, no. 9, pp. 1492–1501, 2020.

[13] S. Ameur, A. Ben Khalifa, and M. S. Bouhlel, "A novel hybrid bidirectional unidirectional LSTM network for dynamic hand gesture recognition with leap motion," *Entertainment Computing*, vol. 35, Article ID 100373, 2020.

[14] S. S. Bangaru, C. Wang, X. Zhou, H. W. Jeon, and Y. Li, "Gesture recognition-based smart training assistant system for construction worker earplug-wearing training," *Journal of Construction Engineering and Management*, vol. 146, no. 12, Article ID 04020144, 2020.

[15] J. M. Fajardo, O. Gomez, and F. Prieto, "EMG hand gesture classification using handcrafted and deep features," *Biomedical Signal Processing and Control*, vol. 63, Article ID 102210, 2021.

[16] P. M. Ashok Kumar, J. B. Maddala, and K. Martin Sagayam, "Enhanced facial emotion recognition by optimal descriptor selection with neural network," *IETE Journal of Research*, pp. 1–20, 2021.

[17] K. M. Sagayam, A. D. Andrushia, A. Ghosh, O. Deperlioglu, and A. A. Elngar, "Recognition of hand gesture image using deep convolutional neural network," *International Journal of Image and Graphics*, Article ID 2140008, 2021.

[18] M. Chen, Y. Tang, X. Zou, Z. Huang, H. Zhou, and S. Chen, "3D global mapping of large-scale unstructured orchard integrating eye-in-hand stereo vision and SLAM," *Computers and Electronics in Agriculture*, vol. 187, Article ID 106237, 2021.

[19] H. Wang, L. Dong, H. Zhou et al., "YOLOv3-litchi detection method of densely distributed litchi in large vision scenes," *Mathematical Problems in Engineering*, vol. 2021, Article ID 8883015, 11 pages, 2021.

[20] F. Wu, J. Duan, S. Chen, Y. Ye, P. Ai, and Z. Yang, "Multi-target recognition of bananas and automatic positioning for the inflorescence axis cutting point," *Frontiers of Plant Science*, vol. 11, p. 510, 2021.

[21] G. Ottoboni, R. Nicoletti, and A. Tessari, "The effect of sport practice on enhanced cognitive processing of bodily indices: a study on volleyball players and their ability to predict hand gestures," *International Journal of Environmental Research and Public Health*, vol. 18, no. 10, p. 5384, 2021.

[22] I. Vilà-Giménez and P. Prieto, "The value of non-referential gestures: a systematic review of their cognitive and linguistic effects in children's language development," *Children*, vol. 8, no. 2, p. 148, 2021.

[23] J. He, J. Tong, and C. Hu, "Influence of snapping roll type an d harvesting speed on 4YW-Q corn harvester," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 37, no. 3, p. 46, 2006.

[24] J. Chen, X. Ning, C. Li, G. Yang, P. Wu, and S. Chen, "The model of the forward speed of the combined harvester refers to the fuzzy adaptive control system," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 45, no. 10, p. 87, 2014.

[25] Y. Chen, Z. Zeng, W. Wang, X. Zou, and P. Zhang, "Virtual environment construction and simulation platform of harvesting machinery," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 50, no. 7, p. 159, 2019.