processes

MDPI

Article

The Application of the Gesture Analysis Method Based on Hybrid RF and CNN Algorithms in an IoT–VR Human–Computer Interaction System

Xin Li [1] and Shuli He [2,*]

[1] Department of Computer Science, Harbin University, Harbin 150086, China
[2] Basic Teaching and Research Department, East University of Heilongjiang, Harbin 150066, China
* Correspondence: heshuli1@hljeu.edu.cn

Abstract: With the development of the Internet of Things (IoT) and virtual reality (VR) technology, the demand for high-precision gesture intelligent analysis of a human–machine interaction module for IoT–VR systems is increasing. Therefore, random forest (RF) and convolution neural network (CNN) algorithms are used in this study to build an intelligent gesture recognition model. The experiments were conducted to test the application performance of the design model. The test results show that the qualification rate of the analytical model designed in this study is significantly higher than that of the comparative model. When the threshold is determined to be 43.26 mm, the analytical qualification rates of the RF-CNN (the method of combining RF with CNN algorithms), faster regions with CNN features (Faster-RCNN), and RF models are 82.41%, 76.10%, and 59.10%, respectively. The calculation time of the RF–CNN model is between the two comparative models. From the test data, it can be observed that the research results have certain significance for improving the accuracy of gesture machine recognition technology in China's VR Internet of Things (IoT) system.

Keywords: RF; CNN; gesture analysis; VR; IoT; human–computer interaction

Citation: Li, X.; He, S. The Application of the Gesture Analysis Method Based on Hybrid RF and CNN Algorithms in an IoT–VR Human–Computer Interaction System. *Processes* **2023**, *11*, 1348. https://doi.org/10.3390/pr11051348

Academic Editors: Rey-Chue Hwang and Huixin Tian

Received: 17 March 2023
Revised: 19 April 2023
Accepted: 26 April 2023
Published: 27 April 2023

1. Introduction

With the progress of science and technology, various types of computer equipment play an increasingly prominent role in human life. All kinds of intelligent devices are indispensable to human life [1,2]. The Internet of Things (IoT) and virtual reality (VR), as the key technologies leading the future development of science and technology, are increasingly used in different intelligent devices. How to realize an efficient human–computer interaction in IoT–VR devices has become a hot topic. Visual gesture analysis technology has considerable application value in this field [3]. Unlike other methods (such as data and color gloves), visual-based visual gesture analysis technology does not need to install any other devices for the user. Users can use this intelligent device under natural conditions, which does not have a significant impact on other activities [4]. Therefore, vision-based gesture analysis and visualization techniques are broadly used in human–computer interaction modules for education, medical treatment, games, and intelligent devices. However, many problems still exist in relation to the gesture analysis method. For example, most of the analytical models used at present cannot accurately identify different hand-joint points, because the overall difference in human hand skin is minor. Moreover, human hands have many joints. In most cases, it takes about 20 joint points to accurately describe hand movements. The existing gesture recognition models cannot solve this problem well, which is also the motivation for this research. Therefore, the contribution of this study is to attempt to improve the accuracy of human–machine interaction gesture analysis for IoT–VR devices by constructing a hybrid algorithm analysis model. This study can also provide a theoretical reference for the design of IoT–VR devices with good gesture recognition capabilities in the commercial field in the future. In addition, the random forest

algorithm is innovatively selected from numerous machine learning algorithms to extract the details of gesture joints and provide high-quality initial-joint positions. The CNN algorithm, which has a strong ability to extract global nonlinear feature relations, is used to extract the original image data, so as to further improve the overall gesture recognition accuracy of the model.

2. Related Works

Experts in the industry have conducted a considerable amount of research to improve the working performance of gesture analysis and recognition models. Xia et al. [5] proposed a human motion gesture recognition algorithm based on a deep neural network to solve the low-efficiency result in the analysis of an intelligent analytical model of hand gestures in a VR system combined with the Internet of Things (IoT–VR). Test experiments showed that the gesture recognition accuracy of the motion gesture recognition algorithm was significantly higher than that of the other model. Rzecki [6] proposed a new time-series classification algorithm and tested its performance for human gesture recognition. This algorithm provided better classification results than the other machine learning algorithms. For the 22 different gestures of 10 volunteers, the error rate of the new algorithm was 37% to 75% less than that of the other algorithms. Yang et al. [7] observed that the performance of the hand motion recognition model in a VR human dance event was low. Therefore, a new algorithm was proposed for hand motion recognition in dance studies. The results show that the resolution accuracy of the gesture motion technology of the algorithm in multiple VR dance videos was higher than that of the other model. The calculation time was also significantly lower than the comparison algorithm, which had a high application value. Zhang et al. [8] observed that complex human gestures were difficult to recognize and learn by machine models. Therefore, a recognition model combining naive Bayes, decision tree, and support vector machine models was proposed in the research. Based on this model, a hand gesture recognition model that could be deployed on the Android platform was developed. The test results show that the gesture recognition accuracy of the model is high, and the recognition results are less affected by hand occlusion.

Jin et al. [9] observed that hand-tapping and -holding movements were not easily recognized by the visual system. Therefore, an improved hand gesture recognition model was designed based on intelligent hybrid composite finger technology. After testing the model in multiple datasets, the recognition accuracy of human knocking and holding movements of the model was significantly higher than that of traditional machine learning recognition and convolution neural network (CNN) models. Yang [10] observed that the ability of intelligent devices to analyze hand gestures during rainfall was significantly weakened. Therefore, an analytical model for hand gestures during rainfall was proposed, which combined a Kalman filter and state transfer function. The performance test results showed that the hand gesture resolution accuracy of the model during rainfall was much higher than that of the traditional analytical model. The analytical results were also less affected by ambient light.

Li et al. [11] observed that gesture recognition was of great significance for human–computer intelligent interactions between the IoT and VR devices. Therefore, a hand gesture recognition model based on an improved spatial fuzzy-matching algorithm was proposed. The model had low requirements for the size of the training dataset. The error of gesture recognition was 15.63% lower than that of the recognition model based on the support vector machine algorithm. Fioranelli et al. [12] proposed a gesture analysis algorithm based on short-time Fourier transform technology and a two-way recurrent neural network. The effectiveness of the algorithm was tested using three common gesture parsing data. It was proved that the accuracy of the gesture analysis of this model was 11.6~24.0% higher than that of the machine learning model. Velliangiri et al. [13] designed an automatic detection model that combined CNN and crow search algorithms to solve false-frame recognition in video images. The test results showed that the method had a

false-frame-recognition accuracy of 97.21% on the experimental dataset, which was higher than the other existing methods. The literature review is summarized in Table 1.

Table 1. Summary table for literature review.

Reference Number	Author	Title	Contribution
[5]	Xia Z, Xing J, Wang C, Li X	Gesture Recognition Algorithm of Human Motion Target Based on Deep Neural Network	A high-precision motion gesture recognition algorithm is designed
[6]	Rzecki K	Classification Algorithm for Person Identification and Gesture Recognition Based on Hand Gestures with Small Training Sets	A new time-series classification algorithm is proposed, which has a stronger gesture classification ability
[7]	Yang W, Wang J, Shi J	Video Quality Evaluation toward Complicated Sport Activities for Clustering Analysis	A new algorithm is proposed for hand motion recognition during a dance
[8]	Zhang H, Xu W, Chen C, Bai L, Zhang Y	Your Knock Is My Command: Binary Hand Gesture Recognition on Smartphone with Accelerometer	A mobile gesture recognition model combining naive Bayes, decision tree, and support vector machine models is proposed
[9]	Jin H, Dong E, Xu M, Yang J	A Smart and Hybrid Composite Finger with Biomimetic Tapping Motion for Soft Prosthetic Hand	An improved hand gesture recognition model based on intelligent hybrid composite finger technology is proposed
[10]	Yang Z	Unscented Kalman Filter (UKF)-Based Algorithm for Regional Frequency Analysis of Extreme Rainfall Events in a Nonstationary Environment	A hand gesture analysis model for rainfall events is designed by combining a Kalman filter and state transition function
[11]	Li H, Wu L, Wang H, Han C, Quan W, Zhao J	Hand Gesture Recognition Enhancement Based on Spatial Fuzzy Matching in Leap Motion	A hand gesture recognition model based on improved a spatial fuzzy-matching algorithm is proposed
[12]	Fioranelli F, Guendel R G, Yarovoy A	Phase-Based Classification for Arm Gesture and Gross-Motor Activities Using Histogram of Oriented Gradients	A gesture parsing algorithm based on short-time Fourier transform technology and a bidirectional recurrent neural network is established
[13]	Velliangiri S, Premalatha J	A Novel Forgery Detection in Image Frames of the Videos Using Enhanced Convolutional Neural Network for Face Images	A video false-frame detection model combining CNN and crow search algorithms is designed

In summary, various, improved hand gesture analysis and recognition models were designed. However, most of these models were only applied to general-life scenarios. The research results that can be applied to VR and IoT scenarios are quite rare. VR and IoT devices will be widely used in the future. The gesture analysis and recognition models must be able to support this application environment. Therefore, our research attempted to design a gesture analysis model that could be applied to the gesture human–computer interaction module of VR combined with IoT intelligent devices.

3. Construction of Hand Gesture Analysis Model Based on Hybrid RF and Improved CNN Algorithms

3.1. Design of Gesture Analysis Model Based on Hybrid RF and CNN Algorithms

One of the core elements in the human–computer interaction of the Internet of Things is to interpret human-motion elements. On this basis, a virtual space model was designed. The human hand has many joints, making it significantly more difficult to understand than other parts of the body. Therefore, RF and CNN models were selected to build models that could improve the accuracy of gesture-recognition technology in the human–computer interaction of the IoT. CNNs have powerful feature-extraction capabilities, which are widely used in gesture analysis tasks. This is also the main reason why the CNN algorithm was chosen for the gesture feature extraction in this study. At present, in the intelligent analysis of hand gestures, most of the regression results are directly output by the CNN network. Figure 1 presents the gesture parsing process. The input of the model is the hand image and the output is the joint node coordinate map. The disadvantage of this method is that a lot of image details are lost. It is difficult to improve the accuracy of the results further [14–17].

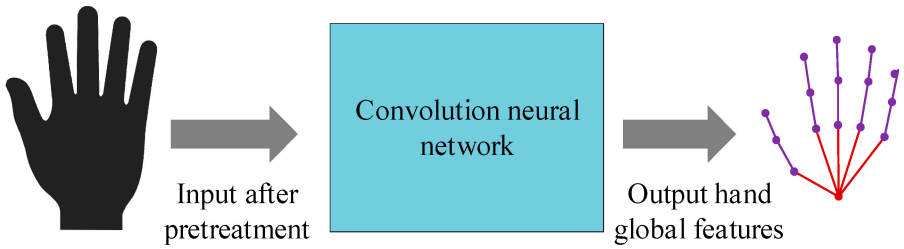

Figure 1. Calculation flow of hand gesture analysis model based on the CNN.

Other scholars have attempted to provide an intelligent analysis of hand gestures through RF algorithms. Compared to other machine learning algorithms, this algorithm has a rapid training speed and low understanding difficulty. For example, support vector machines have a high computational complexity and slow computational speed. Figure 2 presents the analysis process of the algorithm. In this model, the input is the hand image. The output is the joint-point coordinate map. The algorithm extracts the feature of the joint from the input image and outputs the offset corresponding to the joint point. The offset is used by the initial joint to update the position of the subsequent moment [18].

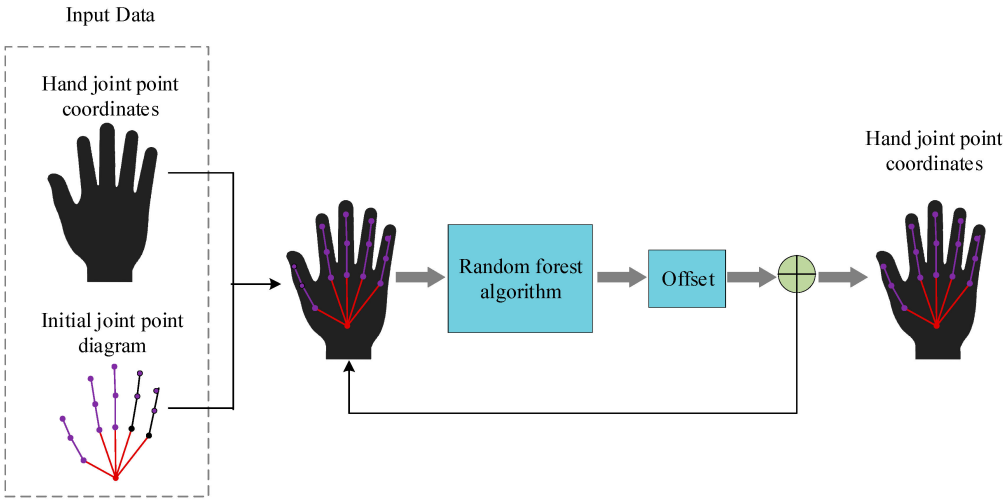

Figure 2. Calculation flow of gesture analysis model based on RF.

The disadvantage of this model is that the initialization effect of joint points is poor. The misplacement of prediction points easily occurs. To sum up, the hand joint can be regarded as a hinge structure with certain integrity. It is suitable to use the CNN to extract global features; however, the CNN's ability to capture details is weak [19–22]. This shortcoming can be solved by the RF algorithm. At the same time, the CNN algorithm can also solve the poor initial joint-point effect of the RF algorithm and provide high-quality initial-joint position images. Therefore, a hybrid gesture analysis model integrating RF and CNN algorithms was proposed. The calculation flow of this algorithm is presented in Figure 3. For this model, the hand image captured from the global joint feature information in the CNN was first input. Then, the information and original image were entered into the RF algorithm to extract the offset of the joint-point coordinates. Finally, the offset of the joint point was used to update the gesture position to complete the gesture resolution [23–25].

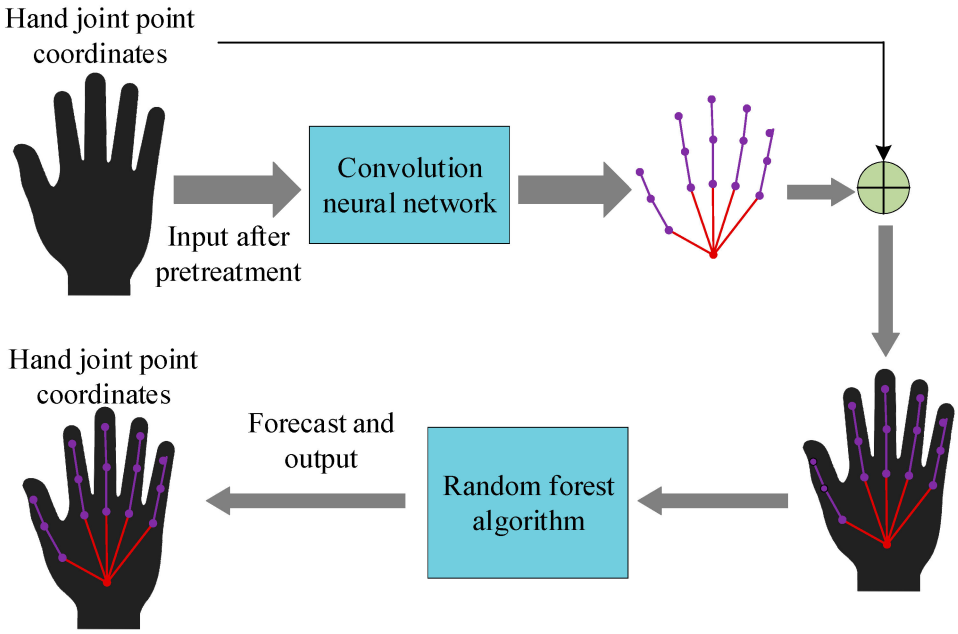

Figure 3. The calculation process of the hybrid gesture analysis model based on RF and CNN algorithms.

The CNN part of the RF and CNN hybrid gesture analysis model is usually facilitated by a 2D neural network structure. However, this traditional neural network structure is not directly applicable to 3D regression tasks. Therefore, the neural network module needed to be improved. The specific design of this part is presented in the following section. The rest of the gesture recognition model is mainly explained here. First, the correction process of the RF algorithm in the model was designed. In essence, the RF algorithm repeatedly trains multiple interrelated decision tree models through the input data. The final calculation results are generated from the calculation results of these models in a certain grass-roots way. Since this algorithm is relatively common, it is not repeated here. After the CNN algorithm output the initial results of the gesture, the initial joint position and depth map were used to extract the features. Then, the results were input into the RF algorithm.

The random difference feature was selected as the form of feature extraction. However, in the previous hand gesture analysis algorithms, the deviation between the initial and actual positions of the joint could not be accurately estimated. Due to the high degree of similarity between the joints, the features containing global information were extracted. Therefore, differential features with global sampling were obtained. The random sampling method was selected to reduce the high number of calculations obtained by the global sampling method. Because the CNN algorithm was used to obtain the initial joint position this time, the global regression problem was mapped to a local regression problem. There was no need to extract the global features.

The dense difference feature was selected for the calculation. This method takes the joint point as the center and selects a sampling area with a side length of w. In this area, samples are obtained in a fixed step to obtain a lattice-style sampling result. The dense difference can be regarded as a special case of random difference. Compared to the latter method, the former has a stronger ability to describe the texture features of the gesture depth map.

In the correction stage, the RF algorithm still needs the offset ΔJ_i data to describe how the current joint point moves to the real position. The index is calculated according to Formula (1):

$$\Delta J_i = J_i^* - J_i^0 \tag{1}$$

In Formula (1), J_i^* and J_i^0 represent the actual coordinate and predicted positions of the i-th joint point output by the CNN network. $i = 1, 2, \ldots, C$. C represents the total number

of joints in the hand image. The data output by the RF algorithm during the prediction process is also offset. For the i-th joint point, after obtaining the offset Δ_i, position J_i^0 of the joint point in the previous phase can be updated. Then, joint position J_i^p can be updated by Formula (2):

$$J_i^p = J_i^0 + \Delta_i \tag{2}$$

The correction phase in the gesture analysis model is completed.

3.2. Design of Improved 3D CNN for Gesture Analysis

The traditional 2D CNN cannot be directly applied to the 3D joint-point estimation. The 2D neural network sets the values of depth maps to grayscale maps, while the convolutional kernel only calculates features in the $x - y$ plane. The information in the z direction cannot be used. When using a 3D CNN, the depth map is viewed as a point that extracts the features in a 3D space to solve the problem of 3D positioning. The study attempted to design an improved 3D CNN dedicated to the intelligent analysis of hand gestures in IoT–VR human–computer interaction. Figure 4 presents the calculation process of the CNN algorithm. The algorithm presented in Figure 4 first maps the input depth map to a point cloud. The distance function with projection was used to describe the point cloud data, and the processed point cloud data was entered into the 3D CNN. Finally, the final 3D coordinates of the joint points were output.

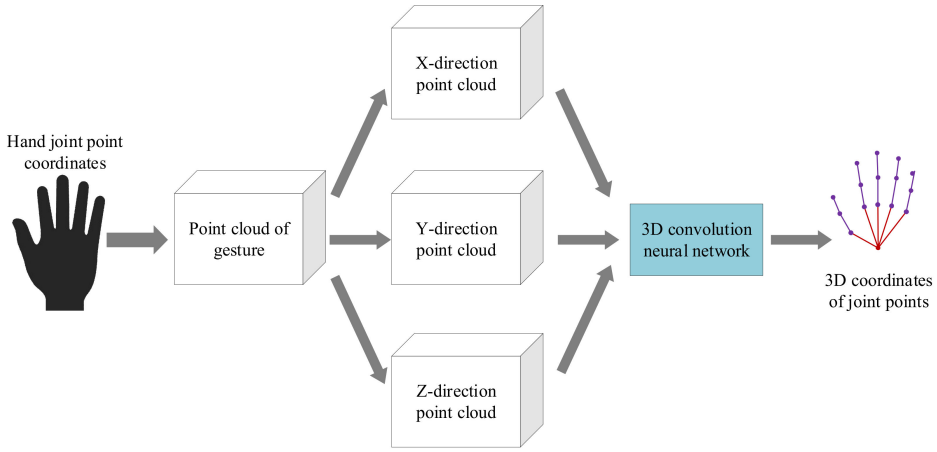

Figure 4. Calculation flow of 3D CNN.

The data obtained by a depth camera can be regarded as 2.5D point cloud. The depth camera cannot perceive the back data, while the front scanning result is in the form of a point cloud. Although 2.5D data can best describe the original information, it cannot be directly input into a 3D convolutional neural network. Therefore, the greatest role of the 3D description is to map a 2.5D point cloud into a form that can be processed in a 3D neural network and describe the hand features in a 3D space. However, it is very time-consuming to accurately calculate the value of a truncated signed distance function (TSDF) with a projection. The TSDF with directional projection was used to replace the TSDF. The design process is analyzed in detail below. To calculate a directional projection TSDF with a $M \times M \times M$ dimension, the length, width, and width of the axis aligned bounding box (AABB) of the point cloud were aligned with the x, y, and z directions of the point cloud. The center of the AABB coincided with the center of gravity of the point cloud. The side length of AABB is l, which can be calculated according to Formula (3):

$$l = \max\{l_x, l_y, l_z\}/M \tag{3}$$

In Formula (3), l_x, l_y, and l_z are the side lengths of the AABB in the three point cloud directions. l_x, l_y, l_z is the resolution of the cube. The TSDF divides AABB into several grids. There is a value in each grid. The value is truncated to be between -1 and 1, and

the TSDF is mapped according to the calculation direction presented in Figure 5. The red line in Figure 5 represents the object to be described. The data in the mesh represent the distance from the mesh to the object surface. Each grid x in the space has an intersection point p connected to the camera. The explanation for the character variables in Figure 5 is presented in the lower part of Figure 5.

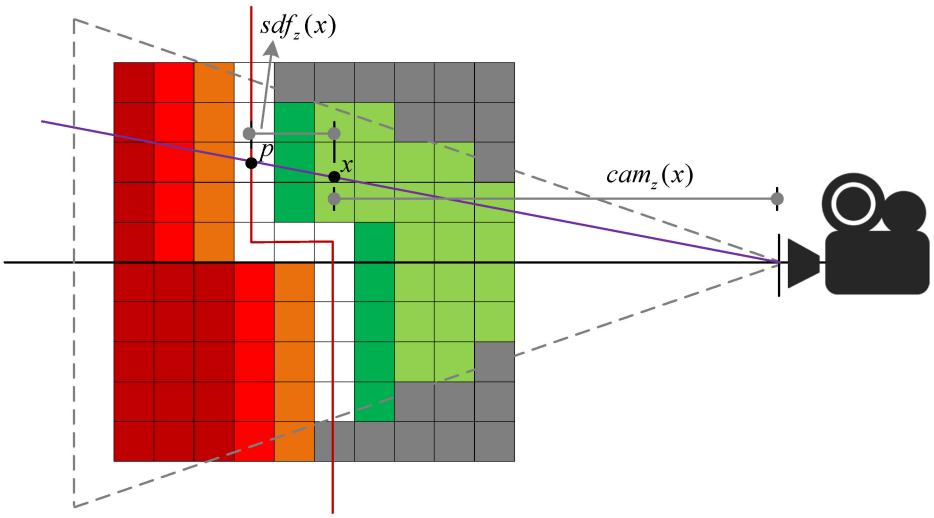

Figure 5. Example of calculating TSDF in the z direction. The gray dotted line is the camera's line of sight. The space in the figure is divided into four sections, namely, a white surface, a green visible space, a red object occlusion, and a gray invisible space.

The distance of x and p in the z direction is $sdf_z(x)$; the calculation method is presented in Formula (4):

$$sdf_z(x) = cam_z(x) - depth(pic(x)) \tag{4}$$

In Formula (4), $pic(x)$ represents the projection point of x on the image. $pic(x)$ is the depth value corresponding to the intersection of the projection point in the projection direction and the plane. The reason why the TSDF value needs to be truncated is that, if it is too far away, this causes the relationship between the value and object surface to not be obvious, thus affecting the description of the object. The TSDF value $TSDF_z(x)$ in the z direction can be calculated according to Formula (5):

$$TSDF_z(x) = \max\left(-1, \min\left(1, \frac{sdf_z(x)}{t}\right)\right) \tag{5}$$

$TSDF_x(x)$ and $TSDF_y(x)$ can be obtained in the same way.

Figure 6 presents the structure of the 3D CNN. Different-colored matrices represent different processing sets. The horizontal arrow represents the data flow. The arrow direction was used to describe the direction of the data flow. The vertical arrow represents the format of the data flow. The matrix splicing presented in Figure 6 concatenates multiple matrices along the channel dimension. Figure 6 describes the calculation process and structure of the neural network in detail.

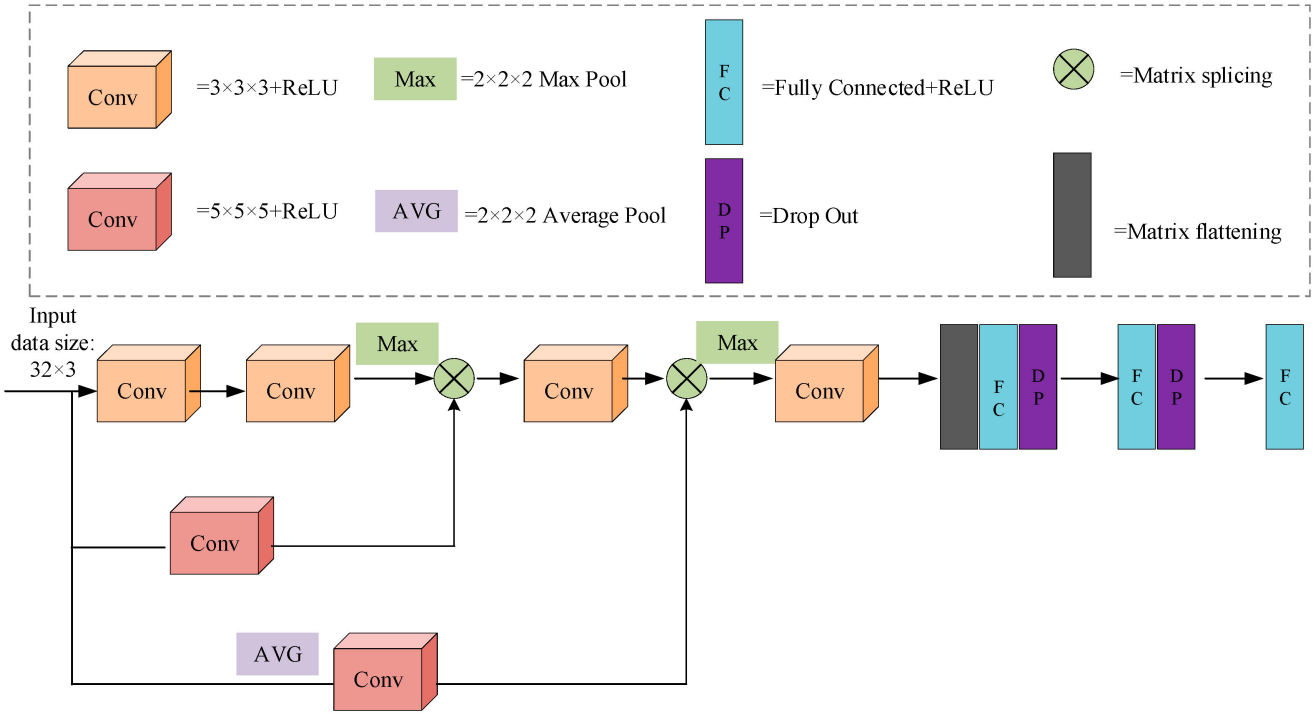

Figure 6. Structure diagram of improved 3D CNN.

The loss function of the 3D neural network is presented below. The training set is (X_n, Φ_n, Ψ_n). X_n, Φ_n, and Ψ_n represent the depth map, joint coordinates, and gesture types of the depth map in the camera coordinate system. $Yn = 1, 2, \ldots, N$. The depth map X_n can be mapped to V_n by zonal projection. Then, Φ_n is converted from the camera coordinate system to the cube coordinate system to obtain Y_n. The calculation method is presented in Formula (6):

$$Y_n = T_{cam}^{vol}(Y^*)/M + 0.5 \tag{6}$$

In Formula (6), $T_{cam}^{vol}(\cdot)$ is the conversion matrix of the camera coordinate system to the directional projection coordinate system. The loss function *loss* in the form of two-norm was selected to construct the CNN. The calculation method is presented in Formula (7):

$$loss = \sum_{n=1}^{N} \|Y_n - \Gamma(X_n)\|^2 \tag{7}$$

In Formula (7), $\Gamma(X_n)$ represents the prediction result of the neural network. To prevent the over-fitting phenomenon of the neural network, the random discard module was incorporated. Each neuron p has the potential to stop working. Before this step is performed, neuron i satisfies Formulas (8) and (9):

$$z_i^{(l+1)} = w_i^{(l+1)} y^l + b_i^{(l+1)} \tag{8}$$

$$y_i^{(l+1)} = f\left(z_i^{(l+1)}\right) \tag{9}$$

$z_i^{(l+1)}$ is the total input corresponding to the i-th neuron in layer $z_i^{(l+1)}$. $w_i^{(l+1)}$ is the neuron weight coefficient. y^l is the output of the l-th layer connected to the corresponding neuron. $b_i^{(l+1)}$ is the bias coefficient of the corresponding neuron. $y_i^{(l+1)}$ is the corresponding prediction result of the neuron. $f(\cdot)$ represents the activation function. When the book-

discarding operation is performed, according to the discarding probability p, the output calculation process of the neurons is presented in Formulas (10) and (11):

$$z_i^{(l+1)} = w_i^{(l+1)} \tilde{y}^{(l)} + b_i^{(l+1)} \tag{10}$$

$$y_i^{(l+1)} = f\left(z_i^{(l+1)}\right) \tag{11}$$

$\tilde{y}^{(l)}$ is the neuron following random discarding in the l-th layer. To prevent the neural network from over-fitting, the regularization term is added. The commonly used regularization methods are one-norm regularization L_{re1} and two-norm regularization. The calculation method is presented in Formulas (12) and (13), respectively:

$$L_{re1} = L + \lambda \|\theta\| \tag{12}$$

$$L_{re2} = L + \lambda \|\theta\|^2 \tag{13}$$

In Formulas (12) and (13), θ is the parameter to be optimized. λ is the weight attenuation coefficient. λ is the loss function. The two-norm regularization term can suppress the size of the parameters while the known parameters are not zero and can also prevent the model from being too sparse. Therefore, the two-norm regularization term was selected to design the neural network. Finally, the training method of the 3D CNN was designed. The objective function is λ. It is determined that the optimal network parameter is the existence of θ^* satisfaction in Formula (14):

$$\theta^* = \arg\min_{\theta} L \tag{14}$$

The gradient descent method was used to optimize the network. In the calculation process, the derivative of λ was first calculated to obtain the gradient. Then, the gradient was used to update the model until the model completed the convergence. The calculation process is presented in Formula (15):

$$\theta_{j+1} = \theta_j + \frac{lr \cdot \partial L}{\partial \theta_j} \tag{15}$$

In Formula (15), θ_j is the model parameter when the iteration reaches the j-th time. $\frac{\partial L}{\partial \theta_j}$ is the calculated gradient. $\frac{\partial L}{\partial \theta_j}$ is the learning rate of the neural network. The IoT–VR human–computer interaction gesture recognition model based on RF and improved CNN algorithms was established. The calculation process is presented in Figure 7.

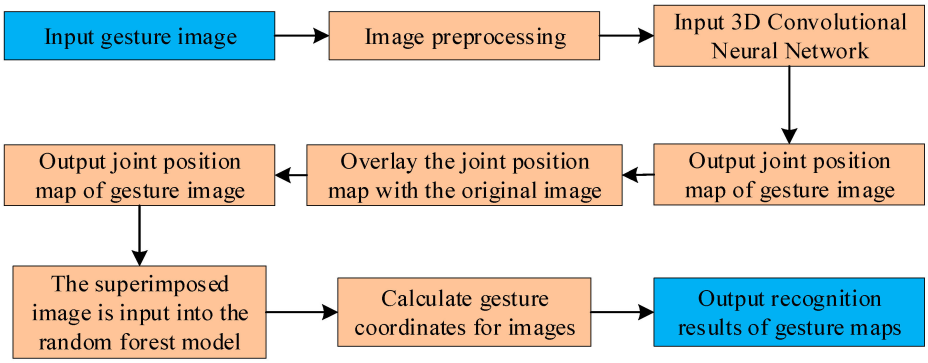

Figure 7. Computation flow of human–computer interaction gesture recognition model of IoT–VR based on RF-CNN.

4. Experiment on the Application Effect of the Gesture Analysis Model Using Mixed RF and Improved CNN Algorithms

4.1. Verification of Parameter Selection for the Gesture Analysis Model

The purpose of this section was to determine the appropriate operating parameters for the gesture analysis model, including loss probability, learning rate, and weight attenuation coefficient. To verify the performance of the gesture analysis model designed in this study, an experiment was designed. The dataset in the experiment was obtained from the backend user database of domestic IoT–VR applications. These data were recorded when users played tactile games using VR devices, with the user's permission. This dataset contained 79,450 gesture depth maps from 10 different users. Each user's data map can be divided into 8 categories based on their different gestures, abbreviated as C1~C8. The joints in the depth maps were annotated using semi-manual methods. These data were used to train and test the recognition effects of each gesture recognition model.

The joints in the depth map were marked in a semi-artificial way. In the study, the classic RF and Faster-RCNN algorithms were selected to build a comparison model. Based on personal neural network training experience, the parameters in the gesture interpretation model were set as follows. The model test and training sets were randomly selected according to the 3:7 ratio, and the batch size was set to 32. The Initial values of the learning rate, dropping probability, and weight attenuation coefficient of the model needed to be obtained by analyzing the test run results. Different discard probabilities were used and the average error change curve of the joint points in the training process is presented in Figure 8. Because numerous drop probability schemes exist, they are presented in two subgraphs.

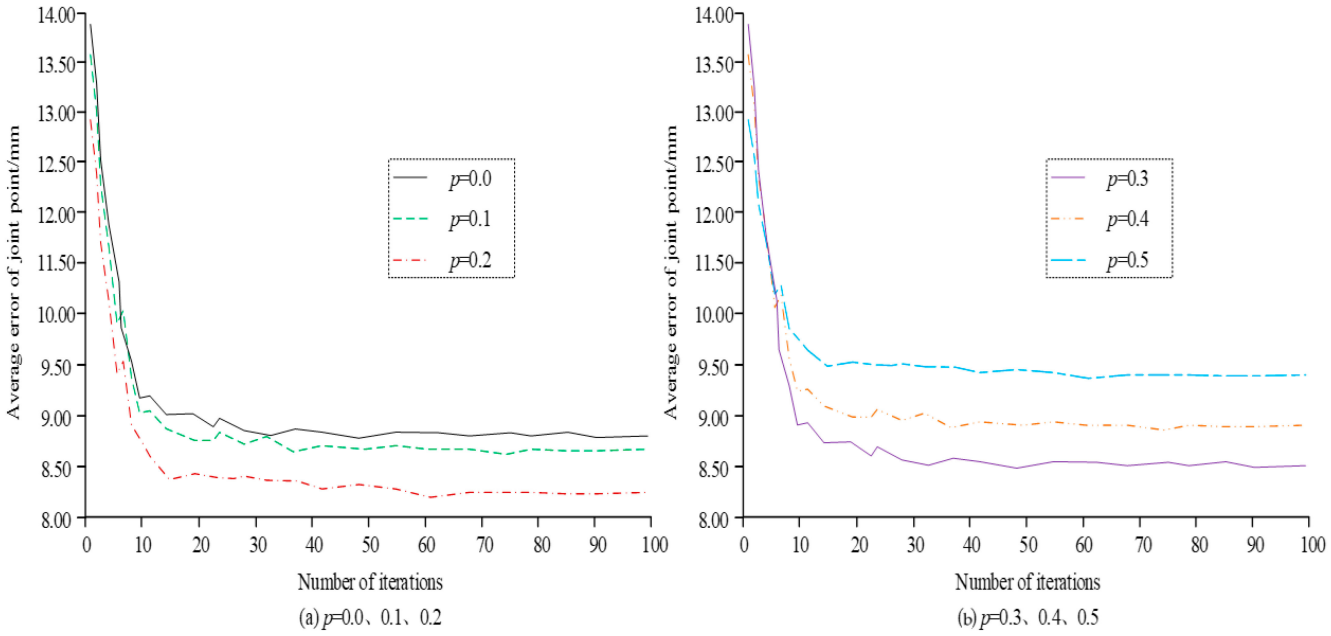

Figure 8. Average error change curve of joint points under different discard probability schemes.

The horizontal axis represents the number of iterations and the vertical axis represents the average errors of the joint points in the training set. Different line-types represent different drop probability schemes. With the increase in the number of iterations, the average errors of the model joint points under each discarding probability scheme present an overall downward trend. However, when the number of iterations is high, the average errors of joint points of each model gradually converge. This shows that the training process of the model is good and avoids the local optimal solution. From the perspective of the loss probability, with the increase in the loss probability value, the average error values of the joint points after model training present a trend of decreasing first and then increasing.

When the number of iterations is 100, the average error values of the corresponding joint points with loss probabilities of 0.0, 0.1, 0.2, 0.3, 0.4, and 0.5 are 8.86, 8.77, 8.34, 8.49, 8.95, and 9.57, respectively. It shows that 0.2 is the most appropriate loss probability.

Different initial learning rate parameters were set to conduct the training experiments, and the results are presented in Figure 9. The results of the horizontal and vertical axes presented in Figure 8 are consistent. From the change curve of the average joint error of each scheme, it can be observed that the curve does not show a significant rebound effect, indicating that the model did not fall into the local optimal solution. From the initial learning rate, it can be observed that, when the initial learning rate is less than 0.005, the lower the initial learning rate, the smaller the average error value of the joint point of the model. However, when the initial learning rate is 0.003, the average error value of the joint points is higher than that of the scheme with the parameter of 0.005. Therefore, it was appropriate to determine the initial learning rate of 0.005 in the model.

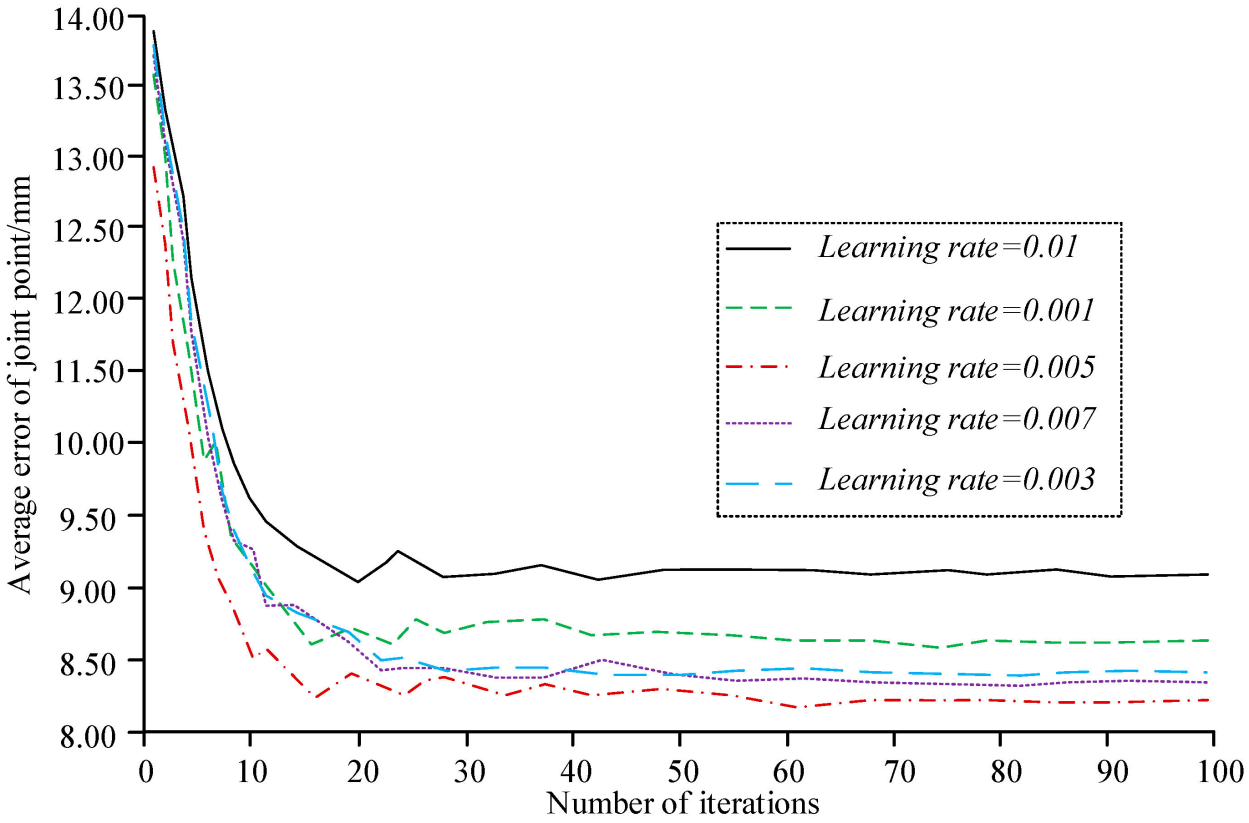

Figure 9. Training effects of RF-CNN algorithm under different initial learning rate parameters.

The statistical results of the model training under different weight attenuation coefficients are presented in Figure 10. Different weight attenuation coefficients correspond to different model training effects. In 100 iterations, the average error values of the joint points corresponding to the weight attenuation coefficients of 0.0, 0.0005, 0.0001, and 0.0001 were 9.24, 8.72, 8.41, and 8.28, respectively. Therefore, it was appropriate to determine the weight attenuation coefficient as 0.0001.

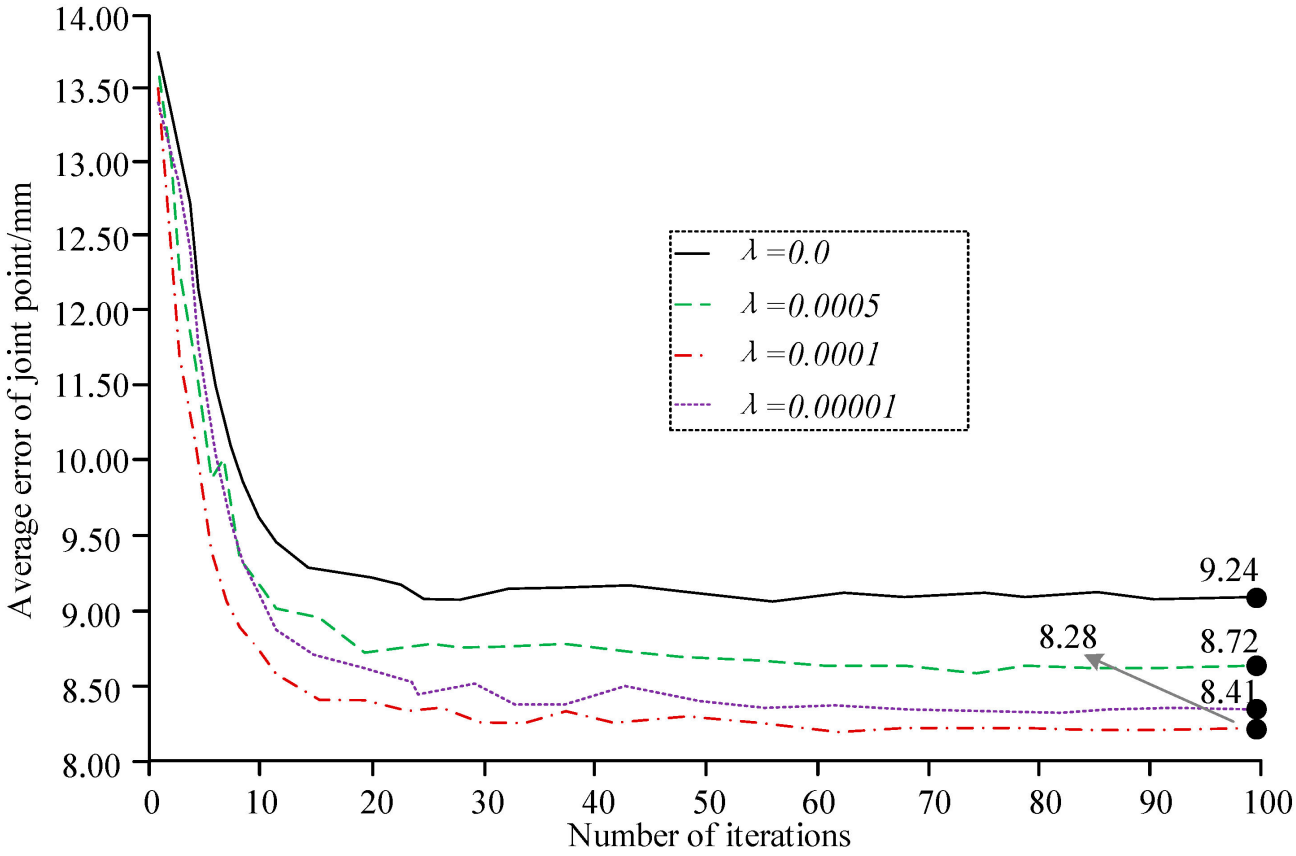

Figure 10. Comparison of training effects under different model weight attenuation coefficients.

4.2. Analysis of Performance Test Results of the Gesture Analysis Model

The purpose of this section is to analyze the performance of the gesture analysis model designed in this study during the training and testing stages. The indicators involved include qualified threshold, gesture type, average joint error, analysis time, and experimental parameters not involved. After training the model, according to the optimal parameter combination, Figure 11 compares the qualified rate of gesture resolution of different models under different thresholds. When the threshold in Figure 10 reached 100 mm, the resolution qualification rate of the RF-CNN model reached 100%. This was because the sample with the highest resolution error corresponded to an error value lower than the judgment threshold of 100 mm, which is classified as a resolution qualified sample. Therefore, in this case, all samples were classified as samples with qualified resolutions. The horizontal axis represents different qualification rate judgment thresholds. The vertical axis represents the qualification rate under the corresponding experimental scheme. Different styles of icons represent different analytical models. "RF-CNN" represents the gesture analysis model designed in this study that combines RF and improved CNN algorithms. The other two are analytical models of contrast gestures. The gray dotted line is the guideline for the viewing data. Each data point in the figure represents the pass rate value of a certain gesture analysis model under the corresponding threshold parameter. For example, when the threshold value was determined to be 43.26 mm, the error between the output joint position of the recognition model and label position was less than 43.26 mm, which is considered to be qualified.

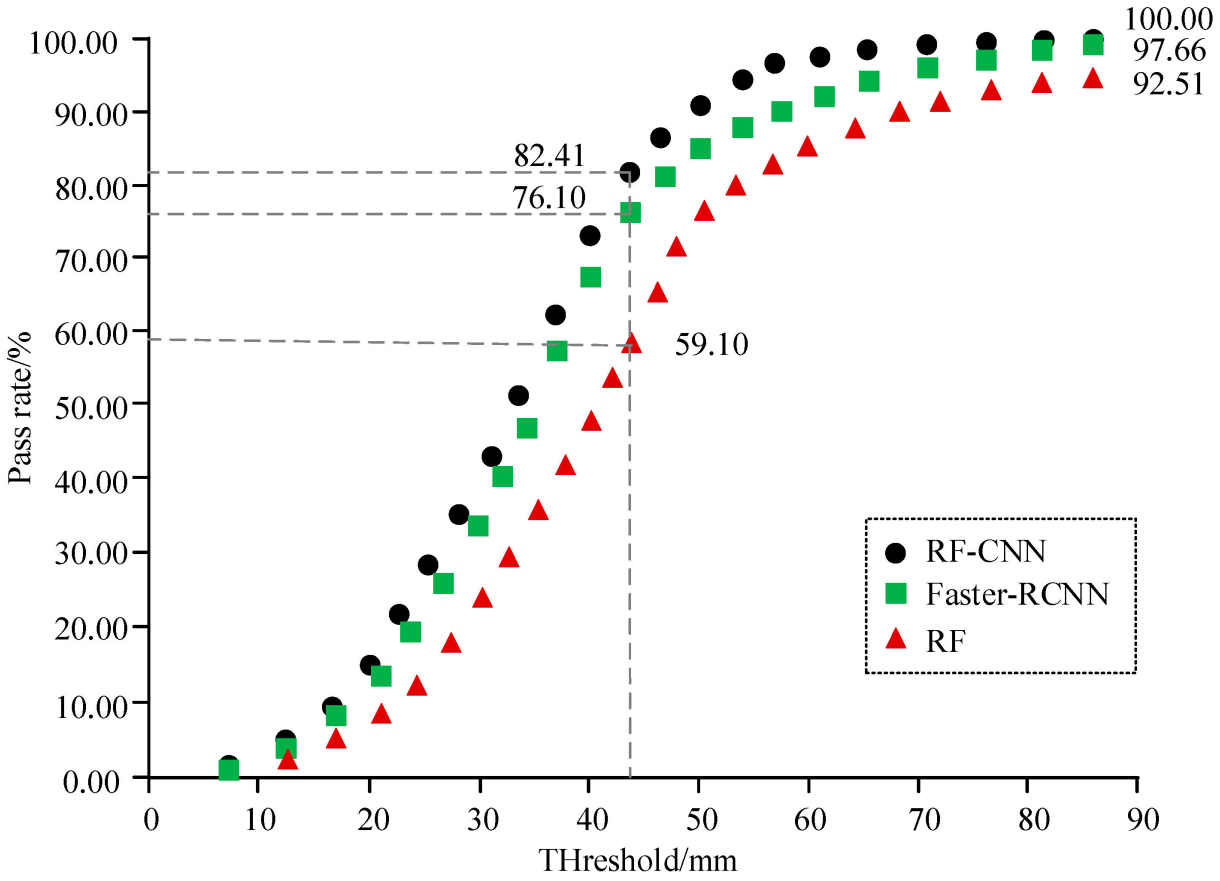

Figure 11. Pass rate of gesture analysis results of each model under different thresholds.

The analytical qualification rates of the RF-CNN, Faster-RCNN, and RF models were 82.41%, 76.10% and 59.10%, respectively. Whether it was the RF-CNN analytical model or the Faster-RCNN and RF comparison models, there was a monotonically increasing relationship between the qualified rate of interactive gesture resolutions and the threshold of the IoT–VR system. Specifically, when the threshold was approximately less than 40 mm, the analytical qualification rate of each analytical model increased with the increase in the qualified threshold. The growth rate continued to accelerate. However, when the threshold was greater than 40 mm, the analytical qualification rate of each analytical model still increased with the increase in the qualification threshold; however, the growth rate gradually decreased. Moreover, the growth rate of the qualification rate of the analytical model designed in the study decreased to 0 at the earliest stage. From the absolute value of the qualification rate, the analytical model designed in this study was always the best. Considering the practical application requirements, setting a threshold of 70 mm was reasonable. During this time, the analytical qualification rates of the RF-CNN, Faster RCNN, and RF models were 98.03%, 94.26%, and 91.40%, respectively. Subsequent research was conducted under this condition.

The results of the analytical model for various gestures are shown in Figure 12. The horizontal axis in Figure 12 was used to present different gesture categories and the average items in all gesture categories. The vertical axis was used to present the average errors of joint points in each category. Different icons represent different gesture parsing models. To improve the reliability of the gesture analysis results, various experimental schemes in this experiment were repeated 10 times. The calculation results are displayed by the mean values and standard deviations of the errors. The RF-CNN model presented in Figure 11 is a comparative model used to verify the rationality of selecting the CNN as the algorithm for extracting gesture features, where "BP" represents the back-propagation neural network. In Figure 12, the average joint error of each recognition model is significantly lower in the

C2 and C3 gesture types. These two types of gestures are more complex and have the most occluded parts, making parsing the most difficult method. However, the average joint error of the RF-CNN analytical model designed in this study was higher than that of the RF analytical model for various gestures. The average joint error of the RF-CNN model in most categories was higher than that of the Faster-RCNN analytical model. However, from the perspective of various overall projects, the average joint error of the RF-CNN model was greater than that of the other models. Moreover, the average error of the overall joint points of the RF-CNN model was also lower than the algorithm designed in this study, proving the rationality of selecting the CNN algorithm to extract the gesture features.

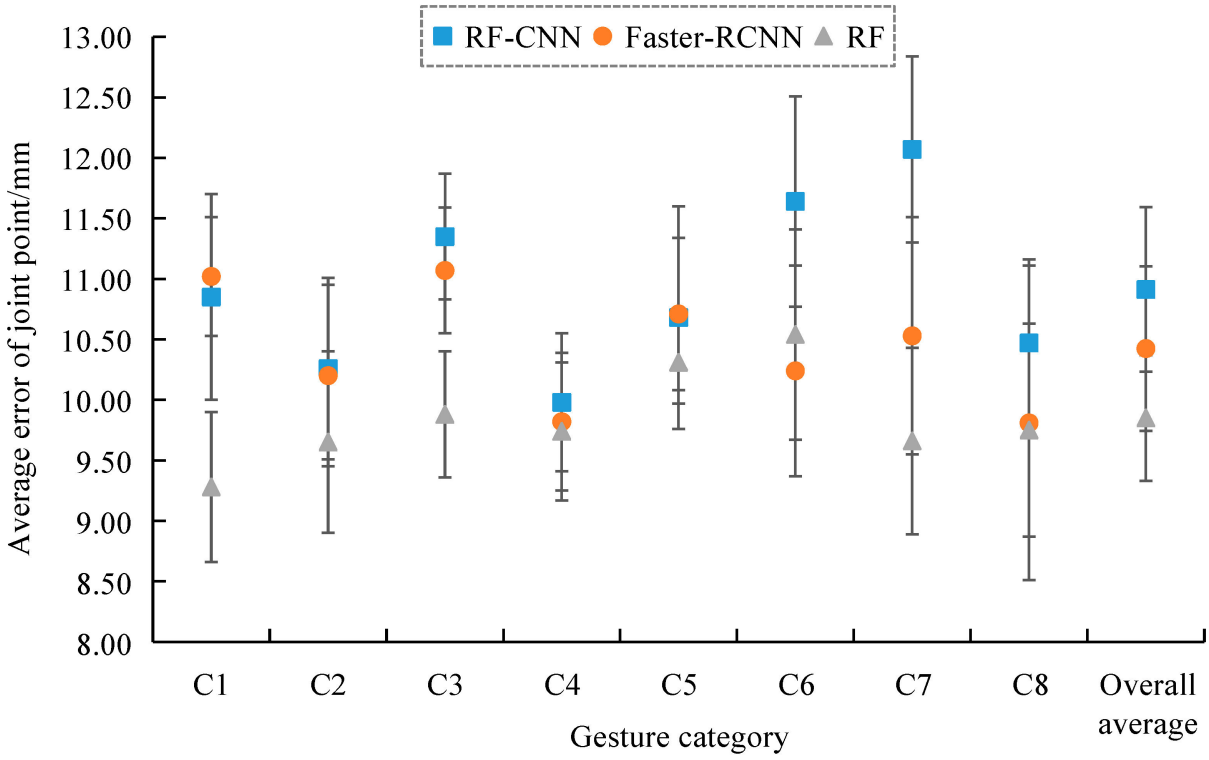

Figure 12. Resolution quality of analytical models for various gestures.

Table 2 presents the analytical efficiency results obtained for each model. The average and maximum analysis times obtained for the RF-CNN analytical model are between the Faster-RCNN model and RF. The relationship between the number of gesture images to be parsed and the parsing time was not a simple linear correlation, but a fuzzy positive correlation. The parsing time of a single image decreased as the number of parsed gesture images increased. The Faster-RCNN analytical model presented the lowest computational efficiency; however, the model was stable, because the maximum calculation time presented the smallest increase compared to the average analytical time under the same conditions. When all the gesture images to be parsed belonged to the test set (i.e., 23,835), the average and maximum parsing times of the RF-CNN, Faster-RCNN, and RF models were 24,770, 51,232, 21,765, 30,385, 60,255, and 64,552 ms, respectively. The SVM presented in Table 2 represents the support vector machine algorithm. The calculation time of the SVM is much higher than that of the RF-CNN algorithm designed in this study, which verifies the rationality of the algorithm design.

Table 2. Comparison of the analytical efficiency results of each model.

Number of Gesture Images to Be Parsed	Average Parsing Time/ms				Maximum Parsing Time/ms		
	RF-CNN	Faster-RCNN	RF	SVM-CNN	RF-CNN	Faster-RCNN	RF
10	16	46	13	89	21	61	15
100	125	296	107	765	175	376	256
1000	1180	2674	983	7420	1852	3109	2199
10,000	11,963	25,514	10,668	73,664	17,740	38,510	26,416
23,835	24,770	51,232	21,765	145,210	30,385	60,255	64,552

To further validate the effectiveness of the model, the designed recognition and comparison algorithms were deployed in C language to a VR hardware system based on the IoT. A total of 100 volunteers were invited to participate in testing each model. After the test was completed, the data were collected and analyzed using Python programming language. Table 3 presents the volunteer group that received the highest satisfaction results with the RF-CNN algorithm model designed in this study. A total of 58 people chose the "satisfactory" evaluation level. The average joint error of this model was 11.88, which is significantly higher than the other comparative models.

Table 3. Comparison of test results based on hardware devices.

Algorithm	Distribution of Satisfaction Level Personnel				Average Error of Joint Points
	Dissatisfied	Relatively Dissatisfied	Neutral	Satisfied	
RF-CNN	2	5	35	58	11.88
Faster-RCNN	7	15	44	34	11.37
RF	11	25	38	26	10.28
SVM-CNN	14	27	40	19	10.54
RF-BP	9	22	37	32	11.09

Finally, the response time of each algorithm model was analyzed. Due to the concise data obtained, statistical charts were not provided in this study. When the number of calculation samples was 23,835, the response times of the gesture recognition models based on the RF-CNN, Faster RCNN, RF, and SVM-CNN algorithms were 26, 58, 37, and 104 ms, respectively. The RF-CNN algorithm designed in this study had the shortest response time.

5. Conclusions

Aiming at solving the problem of the insufficient accuracy of gesture resolution in an IoT–VR system, an intelligent gesture resolution model integrating RF and improved CNN algorithms was designed. To test the application performance of the model, a performance test experiment was designed. The experimental results show that the dropping probability is 0.2, the initial learning rate is 0.005, and the weight attenuation coefficient is 0.0001 for the model designed in the study. The analytical error of the average joint point on the training set was the lowest. The designed model was trained with the optimal parameters. The analytical effect on the test set was compared to that of the contrast model. Statistically, the performance of the design analytical model was always the best from the absolute value of the qualification rate. When the threshold was 43.26 mm, the analytical qualification rates of the RF-CNN, Faster-RCNN, and RF models were 82.41%, 76.10%, and 59.10%, respectively. The average errors of the joint points of each recognition model in C2 and C3 gesture types were significantly lower. However, the average joint errors of the RF-CNN analytical model designed in the study were higher than that of the RF analytical model for all kinds of gestures. From the perspective of various overall projects, the average joint errors of RF-CNN were higher than that of the other models. Moreover, the average and maximum analysis times of the RF-CNN analytical model were between the Faster-RCNN and RF

models. However, due to the research limitations, a commercial-level IoT–VR system intelligent gesture analysis model was not designed. The performance in real application scenarios was not verified. The abovementioned two points should be considered for future research directions.

Author Contributions: Conceptualization, X.L.; formal analysis, X.L. and S.H.; investigation, X.L. and S.H.; methodology, X.L.; writing—original draft, X.L.; writing—review and editing, X.L. and S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, Z.; Yu, Z.; Lou, X.; Guo, B.; Chen, L. Gesture-Radar: A Dual Doppler Radar Based System for Robust Recognition and Quantitative Profiling of Human Gestures. *IEEE Trans. Hum.-Mach. Syst.* **2020**, *51*, 32–43. [CrossRef]
2. Bianco, S.; Napoletano, P.; Raimondi, A.; Rima, M. U-WeAr: User Recognition on Wearable Devices through Arm Gesture. *IEEE Trans. Hum.-Mach. Syst.* **2022**, *52*, 713–724. [CrossRef]
3. Long, Y.; Huang, S.; Peng, L.; Wang, S.; Zhao, W. A Novel Compensation Method of Probe Gesture for Magnetic Flux Leakage Testing. *IEEE Sens. J.* **2021**, *21*, 10854–10863. [CrossRef]
4. Sun, Y.; Fei, T.; Li, X.; Warnecke, A.; Warsitz, E.; Pohl, N. Real-Time Radar-Based Gesture Detection and Recognition Built in an Edge-Computing Platform. *IEEE Sens. J.* **2020**, *20*, 10706–10716. [CrossRef]
5. Xia, Z.; Xing, J.; Wang, C.; Li, X. Gesture Recognition Algorithm of Human Motion Target Based on Deep Neural Network. *Mob. Inf. Syst.* **2021**, *2021*, 5–12. [CrossRef]
6. Rzecki, K. Classification Algorithm for Person Identification and Gesture Recognition Based on Hand Gestures with Small Training Sets. *Sensors* **2020**, *20*, 7279. [CrossRef]
7. Yang, W.; Wang, J.; Shi, J. Video Quality Evaluation toward Complicated Sport Activities for Clustering Analysis. *Future Gener. Comput. Syst.* **2021**, *119*, 43–49. [CrossRef]
8. Zhang, H.; Xu, W.; Chen, C.; Bai, L.; Zhang, Y. Your Knock Is My Command: Binary Hand Gesture Recognition on Smartphone with Accelerometer. *Mob. Inf. Syst.* **2020**, *2020*, 8864627.1–8864627.16. [CrossRef]
9. Jin, H.; Dong, E.; Xu, M.; Yang, J. A Smart and Hybrid Composite Finger with Biomimetic Tapping Motion for Soft Prosthetic Hand. *J. Bionic Eng.* **2020**, *17*, 484–500. [CrossRef]
10. Yang, Z. The Unscented Kalman Filter (UKF)-Based Algorithm for Regional Frequency Analysis of Extreme Rainfall Events in a Nonstationary Environment. *J. Hydrol.* **2021**, *593*, 21–37. [CrossRef]
11. Li, H.; Wu, L.; Wang, H.; Han, C.; Quan, W.; Zhao, J. Hand Gesture Recognition Enhancement Based on Spatial Fuzzy Matching in Leap Motion. *IEEE Trans. Ind. Inform.* **2020**, *16*, 1885–1894. [CrossRef]
12. Fioranelli, F.; Guendel, R.G.; Yarovoy, A. Phase-Based Classification for Arm Gesture and Gross-Motor Activities Using Histogram of Oriented Gradients. *IEEE Sens. J.* **2021**, *21*, 7918–7927.
13. Velliangiri, S.; Premalatha, J. A Novel Forgery Detection in Image Frames of the Videos Using Enhanced Convolutional Neural Network in Face Images. *Comput. Model. Eng. Sci.* **2020**, *125*, 625–645. [CrossRef]
14. Zhu, H.; Xue, M.; Wang, Y.; Yuan, G.; Li, X. Fast Visual Tracking with Siamese Oriented Region Proposal Network. *IEEE Signal Process. Lett.* **2022**, *29*, 1437–1441. [CrossRef]
15. Li, Q.; Lin, H.; Tan, X.; Du, S. H∞ Consensus for Multiagent-Based Supply Chain Systems under Switching Topology and Uncertain Demands. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 4905–4918. [CrossRef]
16. Li, J.; Liu, X.; Wang, Z.; Zhang, T.; Qiu, S.; Zhao, H.; Zhou, X.; Cai, H.; Ni, R.; Cangelosi, A. Real-Time Hand Gesture Tracking for Human–Computer Interface Based on Multi-Sensor Data Fusion. *IEEE Sens. J.* **2021**, *21*, 26642–26654. [CrossRef]
17. Wang, F.; Wang, H.; Zhou, X.; Fu, R. A Driving Fatigue Feature Detection Method Based on Multifractal Theory. *IEEE Sens. J.* **2022**, *22*, 19046–19059. [CrossRef]
18. Gao, P.; Zhao, D.; Chen, X. Multi-Dimensional Data Modelling of Video Image Action Recognition and Motion Capture in Deep Learning Framework. *IET Image Process.* **2020**, *14*, 1257–1264. [CrossRef]
19. Kong, H.; Lu, L.; Yu, J.; Chen, Y.; Tang, F. Continuous Authentication through Finger Gesture Interaction for Smart Homes Using WiFi. *IEEE Trans. Mob. Comput.* **2021**, *20*, 3148–3162. [CrossRef]
20. Huang, C.; Jiang, F.; Huang, Q.; Wang, X.; Han, Z.; Huang, W. Dual-Graph Attention Convolution Network for 3-D Point Cloud Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–13. [CrossRef]
21. Shen, X.; Jiang, H.; Liu, D.; Yang, K.; Deng, F.; Lui, J.C.S.; Liu, J.; Dustdar, S.; Luo, J. PupilRec: Leveraging Pupil Morphology for Recommending on Smartphones. *IEEE Internet Things J.* **2022**, *9*, 15538–15553. [CrossRef]

22. Han, Z.; Lu, Z.; Wen, X.; Zhao, J.; Guo, L.; Liu, Y. In-Air Handwriting by Passive Gesture Tracking Using Commodity WiFi. *IEEE Commun. Lett.* **2020**, *24*, 2652–2656. [CrossRef]
23. Zhou, W.; Guo, Q.; Lei, J.; Yu, L.; Hwang, J. IRFR-Net: Interactive Recursive Feature-Reshaping Network for Detecting Salient Objects in RGB-D Images. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–13. [CrossRef]
24. Cao, B.; Sun, Z.; Zhang, J.; Gu, Y. Resource Allocation in 5G IoV Architecture Based on SDN and Fog-Cloud Computing. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 3832–3840. [CrossRef]
25. Zhang, B.; Zhou, X.; Liu, Y.; Bin, Y.; Yang, Z. Combining Application of Wavelet Analysis and Genetic Algorithm in Wind Tunnel Simulation of Unidirectional Natural Wind Field Near a Sand Ground Surface. *Rev. Sci. Instrum.* **2021**, *92*, 15123.1–15123.9.