*Journal of Network Intelligence*
**Taiwan Ubiquitous Information**
 ISSN 2414-8105
Volume **11**, Number **1**, February **2026**

# Deep Learning-Based 3D Gesture Recognition: Spatial Feature Extraction and Random Forest Integration

Yue-E Yi[1,*]

[1]School of Software, Changsha Social Work College,
Changsha 410004, P. R. China
yiyuee1981@163.com

Peng Hu[2]

[2]School of Systems Engineering, National University of Defense Technology,
Changsha 410003, P. R. China
312240805@qq.com

Todd Stanislav[3]

[3]Faculty Center for Teaching and Learning,
Ferris State University, Michigan 49307, USA
ToddStanislav@ferris.edu

[*]Corresponding author: Yue-E Yi

Received November 06, 2024, revised February 19, 2025, accepted May 17, 2025.

Abstract. *Particularly with developing technologies like virtual reality (VR) and augmented reality (AR), gesture recognition methods are becoming even more crucial in the realm of human-computer interaction. Conventional gesture detection techniques depend on manual feature extraction, hence their sensitivity to environmental changes and generalisation capacity typically limits them in dynamic and complicated surroundings. This work suggests a deep learning-based 3D gesture recognition solution combining spatial feature extraction and random forest integration techniques in order to address these obstacles. Using a deep convolutional nuclear network (CNN), the model first automatically generates high-dimensional spatial features of hand gestures; subsequently, these features are downscaled and ranked in order of importance by a random forest algorithm to choose the most representative features for next recognition activities. We carried out numerous studies including feature extraction accuracy simulation, computation time simulation, feature sensitivity analysis, and cross-valuation experiments to holistically assess the performance of the model. These tests investigated the sensitivity and generalisation capacity of the model to various features in addition to confirming its performance under several gesture kinds and sample sizes. Higher accuracy and improved resilience in 3D gesture recognition tasks are shown by the experimental findings of the strategy suggested in this work. Furthermore, the tests show the model's possibilities in real-time gesture recognition systems. This work not only increases the accuracy and efficiency of gesture recognition but also offers fresh concepts and approaches for next human-computer interaction technologies.*
**Keywords:** three-dimensional gesture recognition; deep learning; spatial feature extraction; random forests.

1. **Introduction.** Especially in applications like VR, AR, and smart interfaces, the value of gesture detection methods in the field of human-computer interaction is growing with technological development [1]. For example, in VR games, players use gestures to control the actions of their characters, while in AR training, gesture recognition is used for real-time feedback and interactive learning. Manual feature extraction—that is, those based on template matching, geometric features, or motion trajectories—is fundamental in traditional gesture recognition techniques [2]. These techniques may be useful in particular situations, but their effectiveness is generally restricted by their sensitivity to environmental changes and their ability to generalise to gesture changes in complex contexts [3], particularly in applications that need real-time responsiveness and high precision [4].

1.1. **Related work.** Since the beginning of the 20th century, gesture recognition technology has undergone a remarkable evolution. From the early days of rule-based methods to modern machine learning-based technologies, each advance has greatly enhanced the naturalness and efficiency of human-computer interaction. Two basic approaches—a vision-based approach and a sensor-based approach—each with special benefits and drawbacks—have been investigated in the study of 3D gesture recognition.

Vision-based techniques extract colour and depth information using cameras to record gestures [5]. Vision-based approaches can offer rich gesture data at a reduced cost with developments in depth sensor technologies [6]. These sensors can provide colour maps and depth maps, which lets algorithms record 3D motion from several angles [7]. Because vision-based techniques can track hand motions and form changes in real time, they are very helpful in dynamic gesture detection. These techniques may thus also provide difficulties for gesture identification, including hand occlusion problems [8], complicated background interference [9], and lighting fluctuations [10]. Furthermore, vision-based approaches usually need considerable processing resources to handle vast volumes of image data, which might restrict their use on limited resources devices.

Sensor-based techniques directly capture hand motion, position, and velocity using a range of sensors and instruments unlike vision-based methods [11]. These comprise optical tracking systems, data gloves and inertial measuring units (IMUs). These techniques have the benefit of being independent of background complexity and lighting conditions, therefore enabling their application in a greater spectrum of surroundings. Important for applications requiring fine-grained gesture control (e.g., medical surgical simulation) [12], sensor-based approaches also enable extremely accurate hand tracking data [13]. These methods have the drawback, too, in that they could force the user to wear extra gear, therefore influencing the comfort and naturalness of the contact. Furthermore, the deployment and maintenance of the sensors can raise the whole cost of the solution [14].

Deep learning methods—particularly the use of CNNs and Recurrent Neural Networks (RNNs)—show significant promise for gesture detection as they evolve [15]. Deep learning technology can significantly reduce the need for artificial feature extraction and improve the accuracy of dynamic gesture detection by automatically learning complex feature representations from large amounts of data. By automatically learning sophisticated feature representations from vast volumes of data, these models help to lower the demand for human feature extraction. Deep learning approaches are especially useful in dynamic gesture detection activities since they shine in managing high-dimensional data and extracting spatial-temporal information [16]. Nonetheless, the training of these models usually depends on a lot of annotated data, which could not be feasible in useful applications. Furthermore, deep learning models' training and inference process can take a lengthy time, which could restrict uses requiring real-time feedback [17].

In essence, both vision-based and sensor-based methods have difficulties even if they have made considerable advancement in 3D gesture detection. While sensor-based solutions must strike a compromise between comfort and cost, vision-based methods must solve light and background interference problems. Though effective, deep learning techniques demand vast volumes of data and computational tools. To thus reach more accurate, strong and natural gesture recognition systems, researchers are searching for fresh approaches to solve these obstacles.

1.2. **Contribution.** This study proposes a deep learning based approach to recognise 3D hand gestures through spatial feature extraction and random forest integration. The innovation of this article is as follows:

(1) **Innovative Fusion of Deep Learning and Random Forest:** Combining the powerful decision-making power of random forest with the effective feature extraction capacity of deep learning, particularly CNN, is one of the main novel ideas of this work. Apart from increasing the accuracy of gesture recognition, this cross-domain technological fusion strengthens the model's adaptability and robustness to intricate gesture modifications. CNN's high-dimensional spatial features automatically extracted enable Random Forest to do more accurate feature screening and classification.
(2) **Breakthrough in 3D spatial feature extraction technology:** In the field of 3D gesture recognition, the traditional 2D feature extraction method can no longer meet the demand. This study achieves a more comprehensive and in-depth understanding of gesture data by developing an advanced 3D spatial feature extraction technique. The technique is able to extract key spatial information from the 3D coordinates of the gesture, including the position of joints, movement trajectories and rotation angles, providing a richer and more accurate feature representation for gesture recognition.
(3) **Optimisation of real-time gesture recognition performance:** In response to the demands of real-time application scenarios, the model in this study is optimised to ensure that the computation time is reduced while maintaining high accuracy. Apart from increasing the responsiveness of the model, this optimisation guarantees stability and practicality in dynamic and changing real-world surroundings. Virtual reality, augmented reality, and other fields including quick gesture interaction depend on this breakthrough.

## 2. Theoretical analysis.

2.1. **Deep Learning: CNN.** As a major subfield of machine learning, deep learning has advanced significantly recently in numerous disciplines like computer vision and natural language processing. Its basic concept is to use multilevel nonlinear transformations to automatically extract features from vast data. Particularly with high-dimensional data—that is, pictures and three-dimensional models—deep learning demonstrates its great capacity [18].

The complexity and high-dimensional aspects of 3D gesture data make conventional manual feature extraction difficult in gesture recognition research. Because of its outstanding feature learning ability, CNNs have grown in favour as 3D gesture recognition tool of choice [19]. CNNs are fundamentally composed of several layers—including fully connected, convolutional, and pooling layers.

Usually, the 3D gesture data comes as a point cloud or voxel grid. Using voxel data, for instance, the input data can be shown as $X \in \mathbb{R}^{W \times H \times D}$, where $W, H$ and $D$ respectively

indicate width, height, and depth. A 3D convolution operation extracts features in the convolution layer; the mathematical structure of the convolution operation is thus:

$$Y(i,j,k) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{p=0}^{P-1} X(i+m, j+n, k+p) \cdot W(m,n,p) + b \tag{1}$$

When $Y$ is the output feature map, $W$ is the convolution kernel and $b$ is the bias term. CNN can extract local gesture features including edges and forms by means of the convolution technique.

Usually including a pooling layer, the model helps to lower the computational complexity and prevent overfitting by means of maximum pooling's action:

$$Y(i,j,k) = \max_{m,n} X(i \cdot s + m, j \cdot s + n, k) \tag{2}$$

When $s$ is the pooling step size, the pooling procedure lowers the dimensionality of the feature map while also helping to maintain salient features. Following several layers of convolution and pooling, the feature map spreads out and feeds into the fully linked layer. The fully connected layer has a formula:

$$Y_{output} = softmax(W \cdot Z + b) \tag{3}$$

while $Z$ is the spread feature vector and $W$ is the weight matrix. The softmax function lets one convert the output to the probability distribution of every category.

Usually, the performance of the model is assessed during model training using the cross-entropy loss function—expressed as:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{4}$$

where $y_i$ is the model's predicted probability and $\hat{y}_i$ is the actual label. Optimisation techniques—such as stochastic gradient descent SGD—adjust the network parameters to raise the model's classification accuracy.

By means of the multilayer structure, the CNN can efficiently capture the intricate aspects of gestures, so offering strong feature support for the random forest integration later on and finally obtaining correct gesture detection.

2.2. **3D feature extraction.** One important stage in 3D gesture recognition is spatial feature extraction, which directly influences recognition efficiency and accuracy. Common used feature extraction techniques like point cloud processing and voxel mesh contrasted to conventional 2D image processing methods include the higher complexity of 3D data [20].

Represented as a set of individual points in 3D space, a point cloud consists of perhaps additional properties (such as colour or normal vector) together with 3D coordinates $(x, y, z)$. Commonly utilised techniques including normal features, curvature features, and global descriptors help one extract effective features from a point cloud. Calculating the normal direction of every point helps one to reflect the surface shape; the formula for computing the normal is as follows:

$$N(p) = \frac{1}{k} \sum_{i=1}^{k} (p - p_i) \tag{5}$$

$N(p)$ is the normal to point $p$; $p_i$ is the points next to $p$; $k$ is the count of points in the neighbourhood. By means of the computation of the major curvature, which is given by:

the curvature feature offers information about the local curvature of the surface:

$$K_1, K_2 = eigenvalues(H(p)) \tag{6}$$

where $H(p)$ is the Hessian matrix at point $p$ and its eigenvalues $K_1$ and $K_2$ represent the principal curvatures.

Global shape features are rather well captured by global descriptors as FPFH (Fast Point Feature Histograms) [21]. The computation of FPFH consists in the building of local geometric characteristics, Eq:

$$FPFH(p) = \sum_{i=1}^{k} Histogram(N(p), N(p_i), d(p, p_i)) \tag{7}$$

where $d(p, p_i)$ is the distance from point $p$ to its neighbourhood point $p_i$.

Divining the 3D space into homogeneous cubic voxels, each with information about points in a certain area, the voxel mesh approach. One can articulate this process as:

$$V(i, j, k) = \frac{1}{N} \sum_{p \in P} \delta Voxel_{(i,j,k)}(p) \tag{8}$$

where $V(i, j, k)$ denotes the number of points within the voxel at position $(i, j, k)$ and $Voxel_{(i,j,k)}(p)$ is an indicator function that returns 1 when the point $p$ is located within the voxel, and 0 otherwise. Maintaining the 3D structural information, the voxelized representation helps simplify the input data. Direct input of the voxelized data into the CNN for feature extraction will enable CNN to aid to extract significant spatial features by means of convolution operation.

These feature extraction techniques efficiently gather the rich information in 3D space. Combining several representations, such point cloud and voxel grid, offers strong feature support for next classification and recognition. These extracted characteristics will offer a strong basis for the integration of random forests and model training, therefore enabling effective gesture recognition [22].

2.3. **Random forest integration.** Especially in 3D gesture recognition, Random Forest is a successful integration learning method extensively applied in classification and regression problems. Building several decision trees and aggregating the prediction outcomes of these trees will help to increase the accuracy and resilience of the whole model.

Random Forest trains decision trees using several randomly chosen multiple sub-samples from the original data set using the "bagging" technique [23]. Features are also randomly chosen for node splitting in the building process of every tree, therefore improving the generalisation capacity of the model and possibly lowering the correlation between trees. Figure 1 shows the Random Forest implementation procedure.

Usually, the information gain, or Gini index, is employed as a dividing criterion while building a decision tree [24]. The Gini index computed using the formula:

$$Gini(D) = 1 - \sum_{c=1}^{C} p_c^2 \tag{9}$$

where $D$ is the dataset, $C$ is the total number of categories, and $p_c$ is the probability that category $c$ appears in the dataset. By calculating the Gini index, the random forest can select the best features for node splitting, thus improving the accuracy of the decision tree.

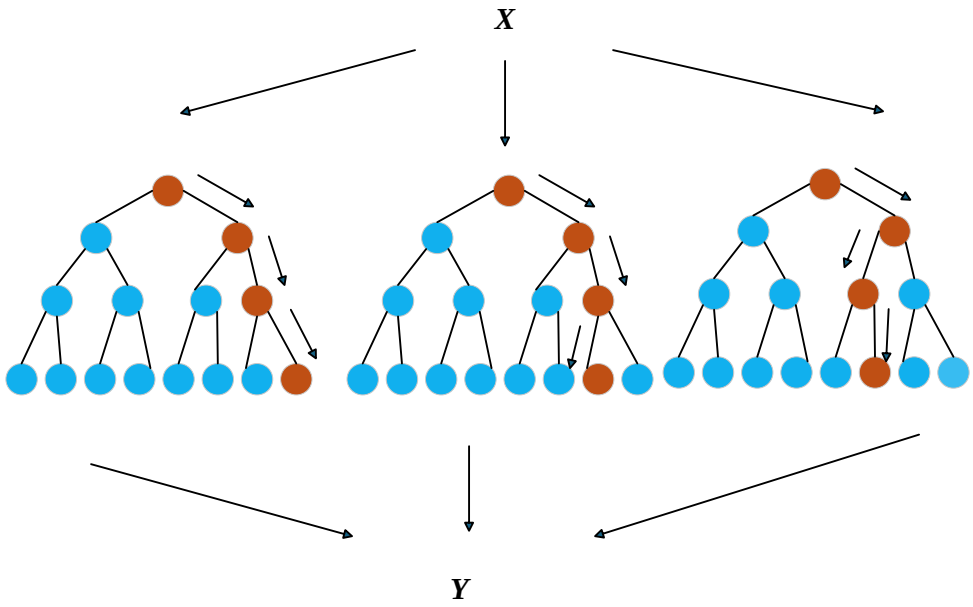


Figure 1. Flow of random forest implementation

The voting mechanism of every tree generates the random forest's last predicted outcome. The formula for estimating categories in the classification issue is:

$$\hat{y} = \text{argmax}_c \left( \sum_{i=1}^{T} w_i \cdot Indicator(y_i = c) \right) \tag{10}$$

where $\hat{y}$ is the final prediction category, $T$ is the total number of decision trees, $w_i$ is the weight of the $i$th tree, and $Indicator(y_i, c)$ is the indicator function. It returns 1 when the prediction of the $i$th tree is $c$, otherwise it returns 0. In this way, Random Forest can effectively integrate the judgement of multiple trees and reduce the risk of overfitting.

Random forest is very appropriate for processing high-dimensional feature data in the 3D gesture recognition problem [25], particularly in the presence of data noise, so displaying good robustness. Furthermore, Random Forest can assess feature importance to help to find the ones most likely to enable gesture recognition [26]. Calculating the value of every feature in the tree splitting process using the formula helps one to accomplish this evaluation procedure:

$$I_{feature} = \sum_{t=1}^{T} \frac{N_{left} - N_{right}}{N} \Delta Gini \tag{11}$$

where $N_{left}$ and $N_{right}$ are the number of samples in the left and right subsets, respectively, after the split, $N$ is the total number of samples, and $\Delta Gini$ is the change in the Gini index before and after the split. This feature importance assessment can help researchers understand which features play a key role in model decision making.

By use of several decision trees, the Random Forest integration approach offers a strong categorisation capacity overall for 3D gesture detection. Its benefits when handling high-dimensional features make it perfect for deep learning following feature extraction; it greatly increases recognition accuracy and lowers overfitting.

## 3. **Model framework and implementation: DL-3DGR-SFRFI.**

3.1. **3D Gesture Recognition Process.** The gesture signals gathered by every camera from various points of view constitute the data source of 3D gesture modelling; hence, its detection depends on great accuracy in positioning and gesture tracking. Every characteristic of the hand skeletal movement consists in the direction, displacement modifications, and varying degrees of freedom of the fingers, palms, and finger joints. The environment greatly influences three-dimensional gesture recognition; hence, in the process of feature acquisition, methods like rotation and translation are included to solve the problem about the distance and angle between the camera and the gesture; but, the tracking problem of gesture movement trajectory is not sufficiently handled with high accuracy, thus a more accurate description of gesture changes requires more features.

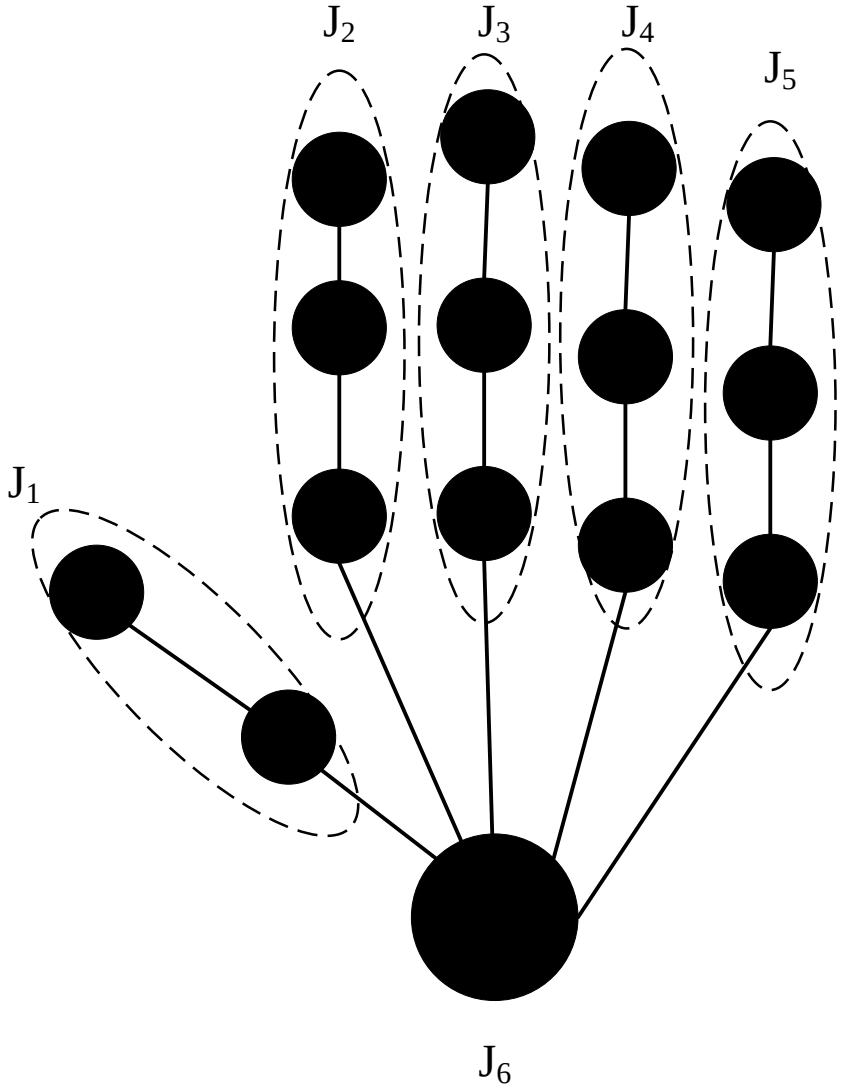


Figure 2. Structure of hand joint points

Figure 2 shows that the joint points of a single hand are distributed as follows: thumb $J_1$ has two joint points; index finger $J_2$, middle finger $J_3$, ring finger $J_4$, and little finger $J_5$ each have three joint points [27]; wrist $J_6$ has one joint point, thereby totalling 15 joint nodes. The angle of forward rotation of the 15 nodes around the $X, Y$, and $Z$ axes determines the hand joint part of the feature vector and thereby measures the hand movement based on the spatial coordinates of these 15 joint points. Regarding whether

these feature vectors are redundant or not, intelligent algorithms or correlation analysis can help to extract more representative features so enhancing the accuracy of the model.

When both hands are simultaneously completing the interaction action, gesture recognition will be more about whether these feature vectors are redundant or not. Correlation analysis or intelligent algorithms can help to extract more representative features to increase the accuracy of the model by means of their optimisation. Create complexity. In this regard, one should take into account the relative location, angle and action synergy between the two hands in addition to the joint motion of one hand. By now the high-dimensional character and complexity of the data call for more sophisticated feature extraction methods to adequately capture the gesture dynamics.

3.2. **A 3D gesture recognition model framework combining deep learning and random forests.** To accomplish more effective gesture identification, the 3D gesture recognition model framework suggested in this work blends random forest approaches with deep learning. Targeting to increase recognition accuracy and real-time performance, the framework has four primary sections: feature extraction, deep learning model, feature fusion and classifier.

CNN is applied to handle input 3D gesture data at the feature extraction level. The CNN effectively captures the spatial aspects of the gesture by means of convolutional procedures. One may depict the result of the convolutional layer by the following equation:

$$F_{out}(i,j,k) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\sum_{o=0}^{O-1} F_{in}(i+m, j+n, k+o) \cdot K(m,n,o) \tag{12}$$

where $O$ is the output feature map, $I$ is the input gesture data, $K$ is the convolution kernel, and $p, q$, and $r$ are the half-width, half-height, and half-depth of the convolution kernel. This step generates the foundation for later processing and removes the local spatial information of the gesture.

In addition, to further enhance the features, adding Batch Normalisation (BN) can be expressed as:

$$\hat{F} = \frac{F - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{13}$$

where $\mu$ and $\sigma^2$ are the mean and variance of the current batch and $\epsilon$ is a small constant to avoid division by zero.

DNN is applied for feature learning in the deep learning model component; the output of DNN may be articulated as:

$$H^{(l)} = f(W^{(l)}H^{(l-1)} + b^{(l)}) \tag{14}$$

where $H^{(l)}$ is the output feature of the $l$th layer, $W^{(l)}$ is the weight matrix of the $l$th layer, $H^{(l-1)}$ is the output feature of the $l-1$th layer, $b^{(l)}$ is the bias term of the layer, and $f$ is the activation function (e.g. ReLU or Sigmoid).

The Dropout strategy is proposed for regularisation to help the model to be more generalised:

$$H^{(l)}_{drop} = H(l) \odot D \tag{15}$$

where $D$ is the Dropout mask with elements of 0 or 1, which controls the retention and dropping of nodes.

The feature fusion phase combines the features from the deep learning model with the geometric features of the gesture. The feature fusion process can be represented as:

$$F_{combined} = \sum_{j=1}^{n} \alpha_j F_j \tag{16}$$

where $F_{combined}$ is the final fused feature, $F_j$ is the $j$th feature, and $\alpha_j$ is the corresponding weight. By weighted summation, the model is able to combine feature information from different sources.

They are normalised with the softmax function to learn the weights of every characteristic:

$$\alpha_j = \frac{e^{F_j}}{\sum_{k=1}^{n} e^{F_k}} \tag{17}$$

This guarantees that the fused characteristics are more like-minded as all weights equal 1.

At last, the model forecasts the categories using Random Forest as a classifier applying a formula:

$$\hat{Y} = mode(T_1(X), T_2(X), \ldots, T_T(X)) \tag{18}$$

where $T_i(X)$ is the prediction of input feature $X$ by the $i$th decision tree, and *mode* denotes the operation of taking the plural. This process can effectively improve the robustness and accuracy of classification by combining the outputs of multiple decision trees. In addition, in order to calculate the output probability of each tree, the following formula can be used:

$$P(Y = c|X) = \frac{1}{T} \sum_{i=1}^{T} Indicator(T_i(X) = c) \tag{19}$$

where $P(Y = c|X)$ denotes the conditional probability of the given input feature $X$ and the prediction category is $c$. $Indicator(T_i(X) = c)$ is the indicator function, which returns 1 when the prediction result of the $i$th tree is the category $c$, and 0 otherwise.

Combining the strong feature extraction capabilities of deep learning with the classification advantage of random forest will help the 3D gesture recognition model framework built above to efficiently enhance the accuracy and robustness of gesture recognition. Through deep convolutional neural networks, the feature extraction layer thoroughly evaluates the spatial aspects of the gesture data; the following deep learning model enhances the feature expression capacity even more. The geometric features and the deep learning extracted features are merged in the feature fusion step, and random forest performs last choice classification to generate a whole recognition process.

## 4. **Performance Evaluation and Analysis.**

4.1. **Experimental setup.** An example simulation is conducted to confirm the 3D gesture recognition model integrating deep learning and random forest performs. The major data source for gesture recognition in this experiment is four cameras gathering gesture data from various angles, including immediately in front, left, right and directly above.

First downscaled using the Random Forest technique, the 3D gesture features were sorted in terms of importance in order to obtain the primary features most likely to affect gesture identification in the testing procedure. Twenty features—including the gesture's shape and the spatial coordinates of all the joints—were chosen using this technique. For a total of 20 characteristics for training, these comprise the edge nodes of the contour of the gesture and also the 3D spatial coordinates of the 15 joints of $J_1$, $J_2$, $J_3$, $J_4$, $J_5$ and $J_6$. Figure 3 shows the first designed gesture observable in Pycharm.

4.2. **Random Forest Feature Extraction.** (1) **Accuracy simulation:** We extract features from the gesture data using the random forest algorithm in the process of 3D gesture recognition in order to increase the recognition speed and accuracy. The features in this experiment are first sorted in terms of downscaling and relevance; the top 10

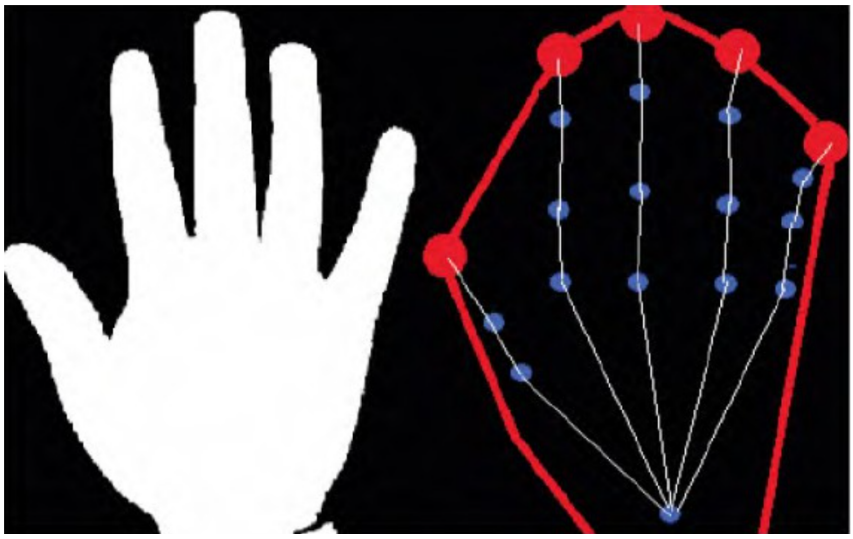

Figure 3. Initialised gesture recognition diagram

features are chosen by the random forest algorithm; and Table 1 shows the optimisation outcomes.

Table 1. Feature Importance Ranking

| **Rank** | **Feature** | **Importance (%)** |
|---|---|---|
| 1 | Wrist joint $J_6$ with respect to the Y-axis angle | 16.137 |
| 2 | Wrist joint $J_6$ with respect to the X-axis angle | 14.271 |
| 3 | Index finger third joint $J_{23}$ with respect to the Y-axis angle | 13.712 |
| 4 | Middle finger third joint $J_{33}$ with respect to the Y-axis angle | 13.247 |
| 5 | Wrist joint $J_6$ with respect to the Z-axis angle | 9.051 |
| 6 | Thumb second joint $J_{12}$ with respect to the X-axis angle | 8.715 |
| 7 | Ring finger third joint $J_{43}$ with respect to the Y-axis angle | 3.392 |
| 8 | Little finger third joint $J_{53}$ with respect to the Y-axis angle | 3.217 |
| 9 | Thumb first joint $J_{11}$ with respect to the X-axis angle | 2.103 |
| 10 | Thumb second joint $J_{11}$ with respect to the Y-axis angle | 2.295 |

Table 1 reveals that whilst the remaining ten features have a value of just 13.86%, the top ten features have a total importance of 86.14%. With an importance of 16.137%, among them the positive angle between the wrist joint point and the Y-axis is judged as most significant.

(2) **Computational time simulation:** This analysis guides us to choose the ten features for next random forest training and display the gesture recognition accuracy in Figure 4.

The experimental results reveal a notable declining trend in the recognition accuracy as gesture types rise. The major causes of this phenomena are the growing complexity of the model resulting from the variety of gestures and the declining distinction between gestures, therefore raising the likelihood of misrecognition. When training, the Random Forest optimised samples are more likely than the original feature samples to produce better recognition accuracy. The recognition accuracy stays at 91.016% when there are

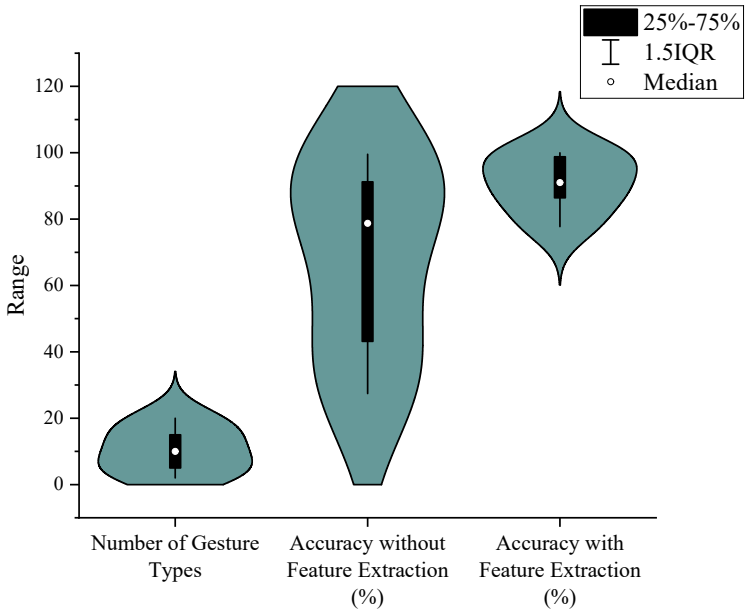


Figure 4. Comparison of gesture recognition accuracy with and without feature extraction

ten different gesture kinds. Therefore, the method suggested in this research is able to retain a high identification accuracy when the number of gesture types to be distinguished is fewer than 10, thereby offering a suitable performance basis for practical uses.

4.3. **Feature sensitivity analysis.** This work aims to assess the individual feature contribution to the Random Forest model in 3D gesture recognition to guarantee the model achieves the optimal balance between performance and complexity.

First, all 20 characteristics are used for training the random forest model; so, the baseline reference is recorded as their recognition accuracy. Each feature is then eliminated one at a time, and the model is re-trained to note the identification accuracy without that feature eliminated. There will be twenty repetitions of this procedure deleting one feature at a time. Most by means of pre- and post-feature removal comparison of the recognition accuracy, we can ascertain the respective importance of every characteristic. Analysing the degree of the accuracy loss following feature removal will enable us to pinpoint the elements most influencing the model performance.

Figure 5 displays the experimental results, showing the variation in recognition accuracy both before and following various feature rejection.

Figure 5 shows that eliminating the wrist pinch angle feature results in the highest drop in accuracy, suggesting that this function is rather crucial for gesture detection. Although eliminating some elements has less effect on the accuracy rate, this indicates that these aspects contribute only little. This study offers a foundation for later feature selection meant to maximise the model's feature set, hence enhancing the general recognition performance.

4.4. **Cross-validation experiment.** This work aims to investigate the individual feature contribution to the random forest model in 3D gesture recognition to guarantee that the model reaches an optimal balance between performance and complexity.

This work evaluates the random forest model using a k-fold cross-valuation technique to increase the generalisation capacity of the model and lower the overfitting phenomena.

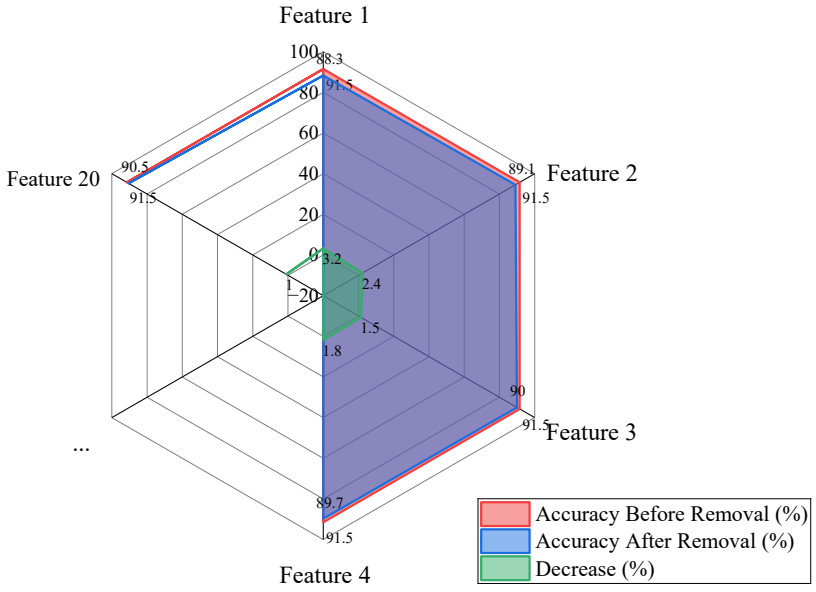


Figure 5. Results of feature sensitivity analysis

First, the dataset is consistently split into $k$ subsets—typically $k = 5$ or $k = 10$ is selected. For every training session, $k - 1$ subsets are then used for training and evaluated on the one left subset. The procedure is done $k$ times such that once each subset is utilised as a test set once. At last, the average accuracy and standard deviation are computed together with the recorded recognition accuracy for every test to evaluate the performance stability of the model on several sets. Table 2 shows the experimental findings, thereby illustrating the end average's accuracy as well as each fold's accuracy.

Table 2. Cross-validation experiment results

| **Fold** | **Accuracy (%)** |
|---|---|
| 1 | 90.12 |
| 2 | 91.5 |
| 3 | 90.85 |
| 4 | 89.95 |
| 5 | 91.2 |
| **Average** | **90.52** |
| **Standard Deviation** | **0.56** |

Table 2 shows that, with an average accuracy of 90.52%, each fold has very constant accuracy. With a standard deviation of 0.56 the model consistently performs on many datasets. This experimental result strongly supports the dependability of gesture recognition in useful applications and confirms the good generalising capacity of the Random Forest model.

5. **Conclusion.** The main contribution of this study is to propose a 3D hand gesture recognition method which combines depth learning and random forest, the accuracy and

robustness of gesture recognition are greatly improved. Combining spatial feature extraction and random forest integration approaches, a 3D gesture identification method based on deep learning is proposed in this work, so improving the accuracy and robustness of gesture recognition. This work constructs an efficient hand gesture recognition model by automatically extracting the high-dimensional spatial features of hand gestures using deep convolutional neural networks, and subsequently performing feature degradation and importance ranking with the help of the random forest algorithm. Experimental results reveal that the model shows tremendous potential in real-time gesture recognition applications and preserves good recognition accuracy under several gesture kinds and sample sizes. Furthermore supporting the generalisation capacity and stability of the model are feature sensitivity analysis and cross-valuation tests.

This study has some restrictions even if its findings show great success. First of all, the quality and variety of the training data greatly influence the performance of the model. The generalisation capacity of the model could be compromised in situations whereby the dataset is small or the samples are not representative. Second, deep learning models usually need large computational resources, which would restrict their use on devices with limited capabilities. Furthermore, especially in dynamic and changing real-world application situations, the models' adaptation to the complexity and changes of gestures should be strengthened.

Future studies can investigate the following paths to help to overcome the aforesaid restrictions:

(1) Data set enhancement: Collecting new data and applying data improvement techniques will help the model to become more generalising and flexible.
(2) Complex Environment Adaptation: Investigate the model's capacity to manage complex lighting situations, background interference, and gesture occlusion to raise the model's stability and accuracy in practical uses.
(3) Cross-domain application: Apply the model to a larger spectrum of domains, including autonomous driving, robot interaction, etc., to assess and maximise the performance in several conditions.
(4) Integrated Learning: Integrated learning is a topic of future research on how best to mix deep learning with other machine learning methods to raise the general model robustness and performance.

All things considered, this work offers a fresh technical solution in the field of 3D gesture recognition, which is quite important for advancing related technologies' development and implementation. Future studies will keep looking for ways to get around current obstacles and improve the practicality and performance of gesture recognition technologies.

## REFERENCES

[1] T. Vuletic, A. Duffy, L. Hay, C. McTeague, G. Campbell, and M. Grealy, "Systematic literature review of hand gestures used in human computer interaction interfaces," *International Journal of Human-Computer Studies*, vol. 129, pp. 74–94, 2019.

[2] A. Dyana and S. Das, "Trajectory representation using Gabor features for motion-based video retrieval," *Pattern Recognition Letters*, vol. 30, no. 10, pp. 877–892, 2009.

[3] L. Radford, "Gestures, speech, and the sprouting of signs: A semiotic-cultural approach to students' types of generalization," *Mathematical Thinking and Learning*, vol. 5, no. 1, pp. 37–70, 2003.

[4] Q. Kong, K. Kuriyan, N. Shah, and M. Guo, "Development of a responsive optimisation framework for decision-making in precision agriculture," *Computers & Chemical Engineering*, vol. 131, p. 106585, 2019.

[5] A. S. Al-Shamayleh, R. Ahmad, M. A. Abushariah, K. A. Alam, and N. Jomhari, "A systematic literature review on vision based gesture recognition techniques," *Multimedia Tools and Applications*, vol. 77, pp. 28121–28184, 2018.

[6] A. L. S. Kawamoto and F. S. C. da Silva, "Depth-sensor applications for the elderly: a viable option to promote a better quality of life," *IEEE Consumer Electronics Magazine*, vol. 7, no. 1, pp. 47–56, 2017.
[7] R. Urtasun, D. J. Fleet, and P. Fua, "Temporal motion models for monocular and multiview 3D human body tracking," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 157–177, 2006.
[8] P. Smith, N. da Vitoria Lobo, and M. Shah, "Resolving hand over face occlusion," *Image and Vision Computing*, vol. 25, no. 9, pp. 1432–1448, 2007.
[9] Z. Tianxu, S. Nong, W. Guoyou, and L. Xiaowen, "An effective method for identifying small objects on a complicated background," *Artificial Intelligence in Engineering*, vol. 10, no. 4, pp. 343–349, 1996.
[10] I. Azcarate, J. Gutierrez, A. Lazkano, P. Saiz, K. Redondo, and L. Leturiondo, "Towards limiting the sensitivity of energy-efficient lighting to voltage fluctuations," *Renewable and Sustainable Energy Reviews*, vol. 59, pp. 1384–1395, 2016.
[11] Y. Xue, Z. Ju, K. Xiang, J. Chen, and H. Liu, "Multimodal human hand motion sensing and analysis—A review," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 2, pp. 162–175, 2018.
[12] B. van Amsterdam, M. J. Clarkson, and D. Stoyanov, "Gesture recognition in robotic surgery: a review," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 6, 2021.
[13] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE Transactions on Multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.
[14] Z. Cheng, M. Perillo, and W. B. Heinzelman, "General network lifetime and cost models for evaluating sensor network deployment strategies," *IEEE Transactions on Mobile Computing*, vol. 7, no. 4, pp. 484–497, 2008.
[15] S. P. Yadav, S. Zaidi, A. Mishra, and V. Yadav, "Survey on machine learning in speech emotion recognition and vision systems using a recurrent neural network (RNN)," *Archives of Computational Methods in Engineering*, vol. 29, no. 3, pp. 1753–1770, 2022.
[16] M. Alam, M. D. Samad, L. Vidyaratne, A. Glandon, and K. M. Iftekharuddin, "Survey on deep neural networks in speech and vision systems," *Neurocomputing*, vol. 417, pp. 302–321, 2020.
[17] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.
[18] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, pp. 1–74, 2021.
[19] A. Elboushaki, R. Hannane, K. Afdel, and L. Koutti, "MultiD-CNN: A multi-dimensional feature learning approach based on deep convolutional networks for gesture recognition in RGB-D image sequences," *Expert Systems with Applications*, vol. 139, p. 112829, 2020.
[20] Y. Xu, X. Tong, and U. Stilla, "Voxel-based representation of 3D point clouds: Methods, applications, and its potential use in the construction industry," *Automation in Construction*, vol. 126, p. 103675, 2021.
[21] L. Hao and H. Wang, "Geometric feature statistics histogram for both real-valued and binary feature representations of 3D local shape," *Image and Vision Computing*, vol. 117, p. 104339, 2022.
[22] E. Che, J. Jung, and M. J. Olsen, "Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review," *Sensors*, vol. 19, no. 4, p. 810, 2019.
[23] R. Genuer, J.-M. Poggi, C. Tuleau-Malot, and N. Villa-Vialaneix, "Random forests for big data," *Big Data Research*, vol. 9, pp. 28–46, 2017.
[24] L. E. Raileanu and K. Stoffel, "Theoretical comparison between the gini index and information gain criteria," *Annals of Mathematics and Artificial Intelligence*, vol. 41, pp. 77–93, 2004.
[25] Y.-J. Zhou and C.-A. Di, "Human motion recognition based on Kalman random Forest algorithm and 3D multimedia," *Multimedia Tools and Applications*, vol. 79, no. 15, pp. 9891–9899, 2020.
[26] S. Balli, E. A. Sağbaş, and M. Peker, "Human activity recognition from smart watch sensor data using a hybrid of principal component analysis and random forest algorithm," *Measurement and Control*, vol. 52, no. 1-2, pp. 37–45, 2019.
[27] Y.-W. Tseng, J. P. Scholz, and M. Valere, "Effects of movement frequency and joint kinetics on the joint coordination underlying bimanual circle drawing," *Journal of Motor Behavior*, vol. 38, no. 5, pp. 383–404, 2006.