Research Article

Gesture Detection and Recognition Based on Pyramid Frequency Feature Fusion Module and Multiscale Attention in Human-Computer Interaction

Min Tu

Teaching Department of Science, Jiangxi University of Technology, Nanchang 330098, China

Correspondence should be addressed to Min Tu; 200599043@jxut.edu.cn

Received 7 April 2021; Revised 27 April 2021; Accepted 30 April 2021; Published 7 May 2021

Academic Editor: Chenxi Huang

Copyright © 2021 Min Tu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at the problem of the absence of detail texture and other high-frequency features in the feature extraction process of the deep network employing the upsampling operation, the accuracy of gesture recognition is seriously affected in complex scenes. This study integrates object detection and gesture recognition into one model and proposes a gesture detection and recognition based on the pyramid frequency feature fusion module and multiscale attention in human-computer interaction. Pyramid fusion module is used to perform efficient feature fusion and is proposed to obtain feature layers with rich details and semantic information, which is helpful to improve the efficiency and accuracy of gesture recognition. In addition, the multiscale attention module is further adopted to adaptively mine important and effective feature information from both temporal and spatial channels and embedded into the detection layer. Finally, our proposed network realizes the enhancement of the effective information and the suppression of the invalid information of the detection layer. Experimental results show that our proposed model makes full use of the high-low frequency feature fusion module without replacing the basic backbone network, which can greatly reduce the computational overhead while improving the detection accuracy.

1. Introduction

Recently, with the rapid development of science and technology, the interaction between humans and machines has appeared in more scenarios [1, 2]. The user's requirements for the friendliness, usability, and high-efficiency of human-computer interaction methods have been further improved [3]. Traditional mouse and keyboard-based interaction methods have been difficult to meet the needs of usage, prompting many scholars to study intelligent interaction methods, such as face recognition, voice recognition, gesture recognition, and motion recognition [4]. Gesture-based interaction has the advantages of being friendly to people with dysphonia and conforming to human communication habits. It conforms to the development direction of human-computer interaction of "human-centered" and "human-computer harmony," which has become a major research hotspot in the field of human-computer interaction [4].

Gesture recognition can be adopted to distinguish different types of gestures and then converted into robot control commands, which have great prospect in human-computer interaction [5]. However, the nonrigid characteristic, the complex, and changeable shape of the human-hand, the background change, and the illumination interference during the interaction increase the difficulty of gesture recognition, which makes gesture recognition become a very challenging problem. Research on high-characterization feature extraction and high-discrimination classification methods for gestures will provide inspiration for other recognition problems [6].

Since the computer vision-based gesture recognition method is intuitive, simple, and noncontact, researchers at home and abroad have turned their attention for research [2, 4–6]. Neverova et al. [7] proposed a recognition method based on active learning and guidance, which uses prior knowledge of skin color to track gestures and uses boost

classifiers to detect and recognize gestures. The detection rate, tracking rate, and recognition rate in a dynamic environment are, respectively, 97%, 99%, and 70%. Gleeson et al. [8] first adopted the background subtraction method to obtain the object, used the entropy information to distinguish the hand from the complex image, extracted the contour of the hand with the help of chain code, and used the center of the gravity contour method to improve the recognition effect. Experimental results show that the average recognition accuracy can reach 95% on the six gestures set. Mazhar et al. [2, 9] proposed a gesture recognition method based on hand skeleton, which uses RealSense depth camera to acquire the joint information of the hand skeleton, uses the geometrical shape of the hand to extract valid descriptors, uses the Gaussian mixture model to obtain the Fisher vector to encode each descriptor, and finally uses the support vector machine (SVM) classifier which realizes the gesture classification. Starner et al. [10] regard gesture recognition as an image processing problem and used the hidden Markov model to process gesture features and then realize gesture recognition with an accuracy rate of 94%.

With the rapid growth of information data, there is a huge demand for data intelligent analysis. The artificial neural network has been widely used to meet the information processing needs in big data era. Deep learning is an important branch of artificial neural networks. Since its theory was proposed by Hinton et al. [9, 11], deep learning has developed rapidly to extract deep features, which is used in speech recognition, image classification, and other types of pattern recognition problems and achieved better results. The deep convolutional network is a type of deep learning, which shares weights through convolution kernels and extracts deep features at a small computational cost [12].

Gesture-based human-computer interaction is a friendly and natural way. Gesture detection and recognition is its core technology. In the field of gesture recognition, scientists have made relatively fruitful results and proposed various gesture recognition methods, but there is still a certain gap with actual engineering applications. Neto et al. [13] used the 3D Hopfield neural network to recognize 15 kinds of gestures and obtained good results in the test data, but the effect in actual engineering applications was mediocre. Wong et al. [14] used convolutional networks to automatically extract gestures and upper body features of the human body and achieved better recognition results. However, the existing deep learning methods are also easily interfered by complex backgrounds, and their ability to recognize complex scenes is weak. The basic backbone for gesture recognition tasks is derived from classification networks in deep learning. With the development of deep neural networks, the network structure is getting deeper and deeper. For example, the ResNet [15] and DenseNet [16] have emerged and shown better feature representation capabilities. In order to solve the above problems and improve the accuracy of the deep gesture recognition algorithm, the Ges_Net model replaced the VGG-16 backbone structure of the original SSD algorithm with ResNet-101 and introduced a deconvolution layer to aggregate context information so as to enhance the high-level semantics of shallow features in gesture

recognition [17]. As we all know, the parameter efficiency is higher than that of ResNet and DenseNet, so DSOD can be used as the basic backbone network to train the gesture recognizer from scratch. However, the complex network structure means huge network parameters, which will lead to a performance reduction in detection speed. It is worth noting that researchers have found that the deep features in the basic backbone network have more semantic information, while the shallow features have more partial-detail descriptions. In other words, the shallow feature information and the deep feature information are complementary in gesture recognition. For example, the feature pyramid network (FPN) uses horizontal connections and top-down architecture with lateral connections to create feature pyramids and achieve more powerful deep feature representations [18]. In recent years, some improved FPN have combined different feature layers by designing a rainbow connection method that uses pooling and deconvolution operations at the same time. Feature fusion Single Shot Multibox detector generates a large-scale feature by fusing multiple shallow feature layers with different scales to generate a large-scale feature and then constructs a new feature pyramid for detection by downsampling on this large-scale feature layer. Although these deep methods have effectively improved the accuracy of traditional gesture recognition algorithms, their complex feature fusion methods have greatly reduced the detection speed [19]. Therefore, how to use the shallow and deep features to effectively integrate the feature pyramid representation determines the detection and recognition performance.

In this study, in order to improve the accuracy of gesture recognition as much as possible with less computational overhead and avoid the limitations and shortcomings of existing gesture recognition methods, a feature fusion feature fusion module based on the feature pyramid network to perform efficient feature fusion is proposed to obtain feature layers with rich details and semantic information. It is helpful to improve the efficiency and accuracy of gesture recognition. In addition, this study further adopts a multiscale attention module to adaptively mine important and effective feature information from both temporal and spatial channels and embeds it in the detection layer. Finally, our proposed network further realizes the enhancement of the effective information and the suppression of the invalid information of the detection layer. Therefore, this study makes full use of the feature information between the detection layers without replacing the basic backbone network, which can greatly reduce the computational overhead while improving the detection accuracy.

2. Feature Pyramid Network

In the convolutional neural network, the inputted image is composed of high-frequency part and low-frequency part, where the high-frequency part has obvious edge details, and the internal characteristics of the low-frequency image information are complete. The low-frequency information is upsampled layer by layer to obtain the low-frequency content, so as to obtain stable low-frequency feature

information [20]. With the continuous iterative training of the deep network, the loss value is reduced through convolution calculation and tends to be stable finally. In the process of backpropagation of the deep network, the missed high-frequency information is constantly compensated due to the extraction of low-frequency information [21]. By separating the high- and low-frequency information of the feature, fully extracting the relevant feature information, and effectively fusing it, the feature expression ability can be enhanced, as shown in Figure 1. As can be seen, the feature pyramid network is divided into a three-level structure. The first level is composed of the primary feature pyramid, and the residual module is used as the primary feature extraction, so the deep features are gradually extracted. In the second level, the network is divided into different frequencies and performs frequency extraction for the low-frequency and the high-frequency feature pyramid. In the third level, adaptive weighted fusion is performed according to the high- and low-frequency information, and the fused feature results are output. Compared with hourglass, the feature pyramid network makes up for the lack of texture information in the feature transfer process. The missing texture information can not only retain the strong geometric representation ability of high-frequency features under the premise of ensuring the calculation efficiency but also integrate effective low-frequency features to further improve the feature information, maintain the stability of feature transfer, and effectively improve the quality of object recognition [22].

There are 2 branches inside the primary feature pyramid module where the residual pyramid structure is adopted and the original scale feature is computed through a shortcut connection. Therefore, multiscale information can be obtained through a series of upsampling. Since the ResNet model is used as the primary feature extraction module, the original features can be retained and the hierarchical information can also keep the input and output dimensions unchanged. The residual pyramid structure can be written as follows:

$$\begin{aligned} F_t(x) &= W_2 * \sigma(W_1 * X_{t-1}), \\ Y &= F_t(X_{t-1}) + X_{t-1}, \end{aligned} \quad (1)$$

where $X, Y \in R^{c \times h \times w}$ are the input and output feature tensors, respectively; h , w , and c represent the spatial dimension and the number of channels; $F_t(x)$ is used as the basic mapping for the fitting of several stacked layers in the primary feature pyramid; $l \in Z^+$ represents the feature layer; $\sigma(\cdot)$ is the ReLU activation function; W_1 and W_2 are the weight matrices. The mapped feature $F_t(x)$ and the original feature X are used to perform the superposition and fusion of pixels, which can alleviate the disappearance of gradients and improve the efficiency of feature transfer.

While processing low-frequency information, it is superimposed and fused with the multiscale features returned by the residual connection. While keeping the feature size unchanged, the low-frequency information is continuously extracted and fused, so that the low-frequency

information is effectively extracted. Therefore, the low-frequency feature pyramid can be denoted as

$$\begin{aligned} x_L &= F_L(F_L(X_{l-1}) + X_{l-1}), \\ Y &= \frac{(U(U(X)) + X_{l-1})}{N}, \end{aligned} \quad (2)$$

where $X \in R^{c \times h \times w}$ is the input feature tensor, and $Y_L \in R^{c \times h \times w}$ is the low-frequency pyramid part, h and w denote the spatial dimension; c represents the number of channels; $F_L(x)$ is used as the basic mapping for the fit of the stacked layers in the low-frequency pyramid; $X_{l-1} \in \{x_0, x\}$ is the $0, \dots, l-1$ feature map generated in the layer; $l \in Z^+$ denotes the feature layer; U is the nearest neighbor upsampling function; X is the fusion estimate obtained by the low-frequency pyramid fitting matrix; and $N \in R^{c \times h \times h}$ is the normalized parameter.

Corresponding to the low-frequency pyramid feature, the function of the high-frequency pyramid feature is to enhance the texture information of the image. The lack of high-frequency information will make the feature edge detail unclear, and the object is blurred, and overlaps, the contrast is small, which makes the detection ability of the model reduced.

In order to ensure the complete and accurate acquisition of high-frequency information, it is only necessary to extract high-frequency features that have been extracted from the primary feature pyramid through shortcuts. Using deconvolution operation, high-frequency information is continuously extracted by multiconvolutional layers, so as to obtain more feature information, retain feature detail information to a large extent, make up for missing feature information, and enhance high-frequency information.

$$\begin{aligned} x_H &= F_H(F_H(X_{l-1}) + X_{l-1}), \\ Y_H &= \frac{1}{N} (W_2 * \sigma(D(x_H)) + W_1 * x_H), \end{aligned} \quad (3)$$

where x_H is the estimated value of the fitting matrix, $D(x)$ is the deconvolution operation, N is the normalized parameter, $l \in Z^+$ represents the feature layer, $\sigma(\cdot)$ is the ReLU activation function, and W_1 and W_2 are the weight matrices.

In order to fuse different multifocus features effectively, it is necessary to calculate the similarity estimation between the current pixel and all the pixels in the feature map, so as to enhance the spatial global information. The calculation process of high- and low-frequency fusion tensor is as follows:

$$\begin{aligned} Y &= \alpha Y_H + \beta Y_L \\ &= \alpha \sum_{j \in N_k} W_{i,j}^T Y_H^{p+i,q+j} + \beta \sum_{j \in N_k} W_{i,j}^T Y_L^{[p/2]+i, [q/2]+j}, \end{aligned} \quad (4)$$

where $Y_H^{p,q}$ and $Y_L^{p,q}$ denote the high- and low-frequency feature tensors of (p, q) , respectively, $N_k(i, j)$ denotes the local region of (i, j) , α and $\beta \in (0, 1)$ are the fusion coefficients for high-low frequency feature. According to the

FIGURE 1: Flow-chart of the feature pyramid network.

value of frequency feature energy, the frequency fusion coefficient is determined, namely,

$$\alpha = \frac{A}{\sqrt{2\pi}} \exp\left(-\frac{(S(Y_H) - 0.1)^2}{2}\right),$$

$$\beta = \frac{A}{\sqrt{2\pi}} \exp\left(-\frac{(S(Y_L) - 0.1)^2}{2}\right),$$
(5)

where $A, B \in [0, 1]$ is the scaling factor; $Y_H = f_H(x, y)$, $Y_L = f_L(x, y)$.

Since the feature pyramid network combines the output of the residual network module to make up for the lack of detail information due to the upsampling of the network, it retains the high-frequency image features to a great extent. Through the fusion, it can output more edge detail information, enhance the spatial dimension feature expression ability, and improve the stability of frequency feature transmission. Therefore, in order to improve the accuracy of gesture recognition as much as possible with less computational overhead and avoid the limitations and shortcomings of existing gesture recognition methods, we adopt the feature pyramid network to perform efficient feature fusion, so as to obtain feature layers with rich details and semantic information in gesture recognition.

3. Our Proposed Pyramid Frequency Feature Fusion Module and Multiscale Attention

The basic idea of the gesture recognition network proposed in this study is to improve the feature pyramid network without changing the basic backbone network and use high-low frequency feature fusion module (HLFFF) and multiscale joint attention module (MAM) to improve the insufficient problem that the deep network is not robust enough for gesture recognition in complex background, so as to improve its detection and recognition performance as much as possible. The overall structure of the proposed model is shown in Figure 2. HLFFF is used to fully mine the feature information between the traditional deep feature pyramid layers, so that the fused feature layer can contain rich geometric details and powerful semantic information, while the multiscale joint attention (MJAM) can further distinguish the importance of different areas and feature channels for the feature layer and emphasize the different importance by giving different attention weights, so as to quickly select more important information from them, and

then use the most effective information to guide the optimization of the model and improve the detection and recognition accuracy.

3.1. High-Low Frequency Feature Fusion Module. The network structure of high-low frequency feature fusion module (HLFFF) is shown in Figure 3. First, the module uses pyramid structure to fuse the deep and shallow features in the detection layer of the traditional feature pyramid network to obtain the shallow features with rich details and semantic information. Second, in order to obtain the deep feature which contains rich feature information, the module also designs a high-low frequency feature fusion structure, where the shallow feature layer is downsampled and fused with the original deep feature layer by different resolutions. The size of receptive field can roughly express the degree of using context information in high-frequency feature, while the shallow feature layer will lose the context information of the deep feature layer due to simple downsampling. Therefore, as shown in Figure 3, this study designs a hierarchical global prior structure-based pyramid pooling module (PPM) to fuse the shallow feature layer. The shallow feature layer can contain rich feature information after the pyramid pooling module and downsampling [23, 24]. Pyramid pooling module uses information with different scales to fully mine the prior knowledge of global context and construct a global prior representation. This structure can effectively reduce the loss of context information between different resolution regions and is an effective global context prior model.

As shown in Figure 2, high-resolution (S_1) and low-resolution (S_2) are the two feature layers that need to be fused. High-resolution pooling (P-1) and low-resolution pooling (P-2) are the output feature layers after the high-low frequency feature fusion module. First, the feature pyramid network is used to fuse S_2 and S_1 to get the P-1 layer. Before feature fusion, S_2 needs to be upsampled to get S_2 up to make its scale consistent with S_1 . Meanwhile, S_2 and S_1 are fused to get the P-2 layer. Different from the previous step in the feature pyramid network, S_1 is selected to generate S_1 -down feature by downsampling. Before downsampling, S_1 needs to go through pyramid pooling module to convert S_1 -P. And then S_1 -down is obtained by downsampling. The size of the S_1 -down is consistent with that of S_2 , and finally, it is fused with S_2 to form the P-2 layer. The whole high-low frequency feature fusion module can be described by the following equation:

FIGURE 2: Overall structure of the proposed model.

FIGURE 3: The network structure of the high-low frequency feature fusion module.

$$\begin{aligned} P_1 &= \text{Concat}(S_1, F_H(S_2)), \\ P_2 &= \text{Concat}(F_L(\text{HLFFF}(S_1)), S_2), \end{aligned} \quad (6)$$

where F_H and F_L represent the upsampling high-resolution and downsampling low-resolution operations, respectively; HLFFF represents the high-low resolution feature fusion module, and $\text{Concat}()$ represents the dot addition operation. In the pyramid pooling module, the initial feature layer first generates three different size feature maps through pooling operation, directly samples the low-resolution feature map by bilinear interpolation, and obtains the feature map with the same size as the original feature map. Finally, the final global pyramid feature layer s is obtained by cumulative average for the three different levels of features. Feature fusion includes three basic operations: downsampling, upsampling, and feature concatenation. The upsampling and downsampling are used to regulate the scale of each layer, while the feature concatenation is realized by cumulative average. In addition, a 3×3 convolution layer is needed to eliminate the aliasing effect caused by upsampling. In order to connect the two feature layers together, we need to use the 1×1 convolution layer to unify the number of channels. In addition, we also choose max-pooling and bilinear interpolation for downsampling and upsampling, respectively, to avoid high complexity.

3.2. Multiscale Attention Module. The network structure and flow-chart of the multiscale attention module (MAM) are shown in Figure 4. Multiscale attention module extracts the global dependence between input sequence and output sequence through a nonlinear transformation, which consists of multiscale dot-product attention (MDA) and squeeze and excitation block. Multiscale dot-product attention is an attention mechanism that uses normalized point multiplication to calculate spatial distribution similarity, while squeeze and excitation block is a kind of network structure which can accurately model the correlation between each channel in the convolution feature layer through adaptive learning so as to select useful feature channels from global information, suppress useless feature channels, and make the extracted feature information more directional. The multiscale attention module in our proposed model combines the advantages of multiscale pyramid fusion and visual attention mechanism, which can simultaneously mine the correlation information in the input sequences from both spatial and channel perspectives, and then extract the useful feature information for the recognition image.

It is assumed that $X = [x_1, x_2, \dots, x_c] \in R^{H \times W \times C}$ is the input feature of the multiscale attention module, H and W are the scales of the input feature layer, C is the number of feature channels, and $x_c \in R^{H \times W}$ is denoted as the c^{th} channel of the input feature layer. Squeeze and excitation block includes two mapping functions: squeeze and excitation. Therefore, we can use $F_{\text{sq}}(\cdot)$ to stand for squeeze mapping, which can encode the whole spatial feature on all the channels into a global feature. In practice, it is often implemented by global average pooling. First, the input feature x_c is mapped into the global feature space $Z \in R^{H \times W \times C}$ by squeeze mapping $F_{\text{sq}}(\cdot)$ to get the c^{th} global feature.

$$F_{\text{sq}}(x_c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W x_c(i, j). \quad (7)$$

After obtaining the global feature, squeeze and excitation block can adaptively learn the nonlinear relationship

FIGURE 4: The network structure and flow-chart of the multiscale attention module (MAM).

between channels by using excitation mapping. The whole process can be expressed by the following equation:

$$F_{\text{ex}}(Z, W) = \sigma(W_2 \mathbf{ReLU}(W_1 Z)), \quad (8)$$

where W_1 and W_2 are the weight matrices and defined in equation (1). In order to reduce the complexity of the model and improve the generalization ability, we use the bottleneck structure including two full connection layers (FC). The first FC layer is the dimension reduction layer, the parameter is W_1 , and $g(\cdot)$ is the ReLu activation function. The last FC layer is used to restore the original dimension, whose parameter is W_2 . The sigmoid activation function is used to get the weight coefficient. Finally, the weight vector s of each channel is multiplied by the original input feature X for feature channel to get the output feature space $\bar{X} = [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_c] \in R^{H \times W \times C}$, where $\bar{x}$ represents the c^{th} convolution kernel of the output feature space, and s_c represents the c^{th} convolution kernel of the output feature space. Thus, we can get $\bar{x}_c = s_c \times x_c$.

Since the feature space is transformed into three different spaces as $Q = W_q^T X$, $K = W_k^T X$, and $V = W_v^T X$, we can get the attention score matrix B , which is obtained by the multiplication of Q and K . Finally, the multiscale attention module will be modeled from two different angles of feature channel and space, respectively, to get the key feature information $\bar{X}$ and $A = B \cdot V$ and add them to get the output $Y = \bar{X} + B$ of the whole multiscale attention module. The output feature space of the proposed model links the long-term dependency of all positions in the input feature and then mines the global context information of the feature map. It highlights the relevant parts of the feature map and uses the refined information to guide the detection and recognition task in human-computer interaction. The output feature space shows the interdependence between the channels through adaptive learning, strengthens the useful feature information on the important channel, and weakens the useless feature information on the nonimportant channel, which is very suitable for gesture recognition in complex environment.

3.3. Loss Function. The main reason why the recognition accuracy of the two-step detection algorithm is higher than that of the one-step detection algorithm is that the proportion of

positive and negative samples in the image dataset is seriously unbalanced during the training process of the one-step detection algorithm, and a large number of simple negative samples dominate the network optimization process, which leads to the lower recall rate. In order to solve this problem, some literatures propose focal loss to reduce the proportion of simple negative samples in the loss function [25]. The specific training process is as follows:

$$L_{\text{fl}} = y\alpha(\bar{y} - 1)^y \mathbf{log} \bar{y} + (y - 1)(\alpha - 1)\bar{y}^y \mathbf{log}(1 - \bar{y}), \quad y \in \{1, 0\}. \quad (9)$$

In the training process, the total objective loss function is the weighted sum of the loss L_{conf} used for classification and the loss L_{loc} used for regression. Therefore, total loss function equation is written as follows:

$$L(x, c, l, g) = \frac{1}{n} (L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g)), \quad (10)$$

where n is the number of positive samples of anchor box; $x_{ij}^p = 1$ indicates the probability p that the i^{th} anchor box matches the real box with category j ; otherwise, $x_{ij}^p = 0$; c is the predicted value of category confidence; L is the position prediction information of positive samples; G is the position parameter of the true bounding-box; and α is the weight coefficient. Therefore, this study mainly focuses on the application of the deep convolutional neural network theory in gesture recognition so as to achieve fast, accurate, and robust gesture detection and the recognition effect and to improve the accuracy of human-computer interaction.

$$L_{\text{conf}}(x, c) = - \sum_{i \in \text{Pos}} x_{ij}^p \mathbf{log}(c_i^p) - \sum_{i \in \text{Neg}} \mathbf{log}(c_i^0), \quad (11)$$

$$c_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}.$$

4. Experimental Results and Analysis

4.1. Dataset. Since gesture recognition is a supervised classification task, for training data, it is necessary to provide

gesture images and their corresponding tags. In order to prove the effectiveness of our proposed method in human-computer interaction, this study verifies it on four benchmark gesture recognition datasets. The overview and characteristic of the three gesture recognition datasets is given in Table 1, and the specific introduction of each dataset is given in this section.

The ChaLearn organization was established to promote machine learning research and to attract university teams and researchers by organizing various challenges [26]. In some of these years, they organized a series of challenges related to human behavior recognition. ChaLearn 2013 was collected by Kinect. The participating teams were asked to use the data to construct a multimodal classifier. Each team was free to use tools and resources to achieve the results. 9 teams with the best scores were selected and announced the results. The frame rate of the video is 20 fps, and the resolution is 640×480 pixels. The database is divided into three parts: training set, verification set, and test set, which contain 6850, 3454, and 3579 samples, respectively. Each sample contains the following four types of data: RGB, depth, mask, and skeleton.

EgoGesture is a multimodal large-scale public dynamic gesture recognition dataset, which contains 83 gestures [27]. This dataset provides the test-bed not only for gesture classification in segmented data but also for gesture detection in continuous data. In addition, we also collected a large number of character images as self-built dataset.

4.2. Experiment Configuration and Parameter Setting

4.2.1. Network Structure. Consistent with the traditional deep gesture recognition algorithm, the proposed algorithm selects the VGG-16 network pretrained on the ImageNet dataset as the basic backbone network to extract the detection layer, and the last two fully connected layers Fc6 and Fc7 are, respectively, converted into convolutional layers Conv_6 and Conv_7. As shown in Figure 2, feature pyramid Conv4_3, Conv7, Conv8_2, Conv9_2, Conv10_2, and Conv11_2 is, respectively, denoted as S1, S2, S3, S4, S5, and S6. Since the structure of feature fusion does not perform well in high-level semantic feature fusion, our improved model chooses the first four layers (S1, S2, S3, and S4) to use the high-low frequency feature fusion module (HLFFF) for feature fusion according to the actual situation, where S1 and S2 are a pair of feature layers, F1 and F2 are generated through HLFFF, S3 and S4 are another pair of feature layers, and F3 and F4 are generated through HLFFF, while S5 and S6 remain unchanged and are denoted to F5 and F6. Then, use multiscale joint attention (MAM) to generate P1, P2, P3, P4, P5, and P6 for F1, F2, F3, F4, F5, and F6 and form the detection layer of the our proposed deep gesture recognition. The detection layer of the proposed algorithm is consistent with the traditional deep learning algorithm in terms of scale and number of feature channels.

4.2.2. Training Details. The scale of the input image of the improved algorithm is divided into 300×300 and 512×512 .

TABLE 1: The overview and characteristic of the three gesture recognition datasets.

Dataset	Categories	Samples	Training	Testing
ChaLearn 2013	24	52381	41904	10477
EgoGesture	44	474278	38000	94278
Self-built	12	6281	5025	1256

The batch gradient descent algorithm (BGD) is used for optimization. The batch size is set to 32, and the maximum number of iterations is 120 k. The warmup strategy is used to set the learning rate, which helps to slow down the over fitting phenomenon of mini batch in the initial stage and keep the distribution stable and the deep stability of the model. The learning rate of the first 1 k iteration is 10^{-4} . From 1 k to 80 k times, the learning rate is increased to 10^{-3} , then from 80 k to 100 k times, the learning rate is reduced to 10^{-4} , and finally from 100 k to 120 k times, the learning rate is reduced to 10^{-5} . In order to ensure the fairness of the experiment, our proposed model first trained and evaluated the traditional comparison algorithms and then performed the corresponding simulation experiment on the proposed algorithm.

4.2.3. Experimental Environment. All experiments in our study are performed on the same hardware, namely, Nvidia GTX 1080 Ti GPU and Intel i7-7800 $\times$ 3.50 GHz CPU. The software environment is Python 3.5 and TensorFlow 1.4 as deep learning framework under the Ubuntu 16.04 system and CUDA 8.0.

4.3. Evaluation Metrics. It is well known that recognition accuracy, $F1$, and recall are commonly used as criteria to evaluate model performance in image classification tasks [28]. $F1$ score is an index used to measure the accuracy of the classification model in statistics, which is used to measure the accuracy of unbalanced data. It takes into account the accuracy and recall of the classification model. $F1$ score can be regarded as a weighted average of model accuracy rate and recall rate. Its maximum value is 1 and its minimum value is 0. However, $F1$ is not suitable for multicategory tasks [29]. Therefore, we use macro $F1$ and micro $F1$ scores to describe precision and recall. Macro $F1$ score is calculated by averaging the $F1$ of all categories and by averaging the precision and recall of all instances. Macro $F1$ score is defined as follows:

$$\text{Macro } F1 = \frac{2\text{Precision}_{\text{macro}}\text{recall}_{\text{macro}}}{\text{Precision}_{\text{macro}} + \text{recall}_{\text{macro}}},$$

$$\text{Precision}_{\text{macro}} = \frac{1}{c} \sum_{i=1}^c \frac{TP_i}{TP_i + FP_i}, \quad (12)$$

$$\text{Recall}_{\text{macro}} = \frac{1}{c} \sum_{i=1}^c \frac{TP_i}{TP_i + FN_i},$$

where FP_i , FN_i , TN_i , and TP_i are the number of the gestures classified incorrectly into a class C_i , the number of the

gestures belonging to C_i classified incorrectly into other classes, classified correctly into other classes excluding C_i , and classified correctly into C_i . Similarly, the definition of MacroF1's equation is similar to it.

4.4. Ablation Analysis. As given in Table 2, in order to verify and evaluate the effectiveness of the gesture recognition algorithm, high-low frequency feature fusion module (HLFFF) and multiscale attention module (MAM) are proposed in this study, where this study performed a series of ablation analysis in the ChaLearn 2013 dataset. In order to ensure the fairness of the experiment, this study first trained and evaluated the traditional deep gesture algorithm (TDGR). The results show that for a 300×300 input image, the proposed algorithm has a mAP of 79.1%, which is 1.6% higher than the traditional TDGR. The increase of the parameter amount decreased slightly, reaching 39.6. For a 512×512 input, the proposed algorithm has a mAP of 81.0%, which is 1.2% higher than the traditional TDGR, and the parameter amount has increased by 24.57 M, so the FPS has also dropped slightly to 20.8. The results of further ablation experiments showed that the mAP obtained by using HLFFF alone to improve the TDGR algorithm was 78.8% and the FPS was 47.1, the parameter amount increased by 14.04 M, and the mAP obtained by using MAM alone to improve the SSD algorithm was 78.6% and the FPS was 48.1, the corresponding increased parameter amount is 10.53 M, and the use of the SEB network to improve the gesture algorithm alone has mAPs of 78.2 and 78.4, respectively, FPS is 53.1 and 51.5, respectively, and the amount of parameters has increased by 6.24 M and 4.29 M, respectively. It can be found that the high-low frequency feature fusion module (HLFFF) and multiscale attention module (MAM) proposed in this study are very effective, and MAM is obviously more effective than SEB because it combines the advantages of the two, which can be obtained from mining key information at the same time in both directions of space and passage. In terms of parameter quantity, HLFFF is more complicated than MAM, and its increased parameter quantity is relatively more. Since MAM module combines SEB, its increased parameter quantity is the sum of the two increased parameter quantities. On the whole, all the parameters added by the proposed algorithm are derived from HLFFF and MAM. Compared with the use of complex basic backbone networks and complex feature fusion methods, the TDGR algorithm in this study has increased a small amount of computational overhead, while achieving greater performance improvement is very effective.

4.5. Qualitative and Quantitative Comparison. Recently, there have been many gesture recognition algorithms for natural images, but these algorithms are based on the detected hands for gesture recognition. The purpose of our proposed model is to integrate hand detection and gesture recognition to improve the precision of gesture recognition and reduce the influence of background interference. Since the detection algorithm based on deep learning has achieved great results in the field of natural images. In order to further verify and evaluate the gesture recognition algorithm in this experiment, it is compared with some other advanced gesture recognition

TABLE 2: Performance analysis of different modules.

HLFFF	MAM	mAP	#Paras (M)	FPS	Precision (%)
1	0	76.2	56.38	53.1	67.2
0	1	79.4	54.43	51.5	69.5
0	0	71.6	60.67	48.1	61.1
1	1	79.6	53.67	54.1	72.8

algorithms, such as GestNet [16], GMRCNN [17], Gst_SSD [18], Dense_Ges [19], and FMD + DTW [20]. All selected comparison algorithms are analyzed using the source code given by the author. Although these deep models are not aimed at gesture recognition, these models do not distinguish object categories. As long as sufficient training data are given, the corresponding detection results can be obtained. All comparison models are experimented with the same training set and test set so as to facilitate fair and fair comparative analysis of all comparison algorithms. The quantitative comparison results for recognition accuracy are given in Table 3.

Gst_SSD is based on the multiscale convolutional detection SSD algorithm and introduces the feature fusion idea of different convolution layers. After the dilated convolution downsampling operation and the deconvolution upsampling operation, the fusion of the shallow visual convolutional layer and the deep semantic convolutional layer are fused in the network structure, so as to replace the original convolutional layer for gesture recognition, which can improve the recognition accuracy. However, the Gst_SSD model cannot recognize the small-scale gestures. GMRCNN uses the gesture matrix as a parameter to input into the neural network for deep learning pattern recognition, uses the fully connected network structure and parameter selection to prevent overfitting, and then combines fingertip detection and elimination algorithms to improve recognition accuracy. It is worth mentioning that the accuracy of the GMRCNN algorithm is completely dependent on the gesture matrix. GestNet uses the PNN neural network for classification and combines the transfer learning method to transfer the deep learning model to the constructed model. The verification experiment conducted on the public dataset Keck Gesture shows that the deep network has certain advantages for gesture recognition, but it still has shortcomings and is not suitable for engineering applications. Dense_Ges obtains the gesture anchor of the training set through K-means dimensional clustering [30] and is responsible for detecting gestures of different scales; finally, it uses transfer learning and fine-tuning methods to train the gesture recognition model. The average recognition rate of gestures with large interference in complex backgrounds is only 49.2%. FMD + DTW is a method that combines DTW and deep learning for gesture recognition. By analyzing the extracted skeleton information, real-time coarse-grained recognition of continuous gestures is realized, and then, ROI is extracted through joint point calculation, and the feature extraction is performed on ROI information using a 7-layer CNN network, and fine-grained recognition is obtained.

TABLE 3: Recognition accuracy for different models.

Datasets	Models					
	GestNet (%)	GMRCNN (%)	Gst_SSD (%)	Dense_Ges (%)	FMD + DTW	Proposed (%)
ChaLearn 2013	53.32	64.34	86.61	67.98	67.12%	70.06
EgoGesture	52.41	77.43	70.15	72.00	52.00%	77.14
Self-built	78.21	84.18	80.05	76.93	82.1%	83.79

5. Conclusion

In order to improve the accuracy of gesture recognition as much as possible with less computational overhead and avoid the limitations and shortcomings of existing gesture recognition methods, a multiscale feature fusion module is based on the feature pyramid network to perform efficient feature fusion so as to obtain feature layers with rich details and semantic information. It is helpful to improve the efficiency and accuracy of gesture recognition. In addition, this study further adopts a multiscale attention module to adaptively mine important and effective feature information from both temporal and spatial channels and embeds it in the detection layer. Finally, our proposed network further realizes the enhancement of the effective information and the suppression of the invalid information of the detection layer. Experimental results show that our proposed model makes full use of the high-low frequency feature fusion module without replacing the basic backbone network, which can greatly reduce the computational overhead while improving the detection accuracy.

Data Availability

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

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

This work was supported by Jiangxi University of Technology.

References

- [1] G. Li, H. Wu, G. Jiang, S. Xu, and H. Liu, "Dynamic gesture recognition in the internet of things," *IEEE Access*, vol. 7, pp. 23713–23724, 2018.
- [2] J. Kofman, X. Wu, T. J. Luu, and S. Verma, "Teleoperation of a robot manipulator using a vision-based human-robot interface," *IEEE Transactions on Industrial Electronics*, vol. 52, no. 5, pp. 1206–1219, 2005.
- [3] M. Tölgyessy, P. Hubinsk, L. Chovanec, F. Duchoň, and A. Babinec, "Controlling a group of robots to perform a common task by gestures only," *International Journal of Imaging and Robotics*, vol. 17, pp. 1–13, 2017.
- [4] M. Tölgyessy, M. Dekan, F. Duchoň, J. Rodina, P. Hubinský, and L. u. Chovanec, "Foundations of visual linear human-robot interaction via pointing gesture navigation," *International Journal of Social Robotics*, vol. 9, no. 4, pp. 509–523, 2017.
- [5] P. G. Jung, G. Lim, S. Kim, and K. Kong, "A wearable gesture recognition device for detecting muscular activities based on air-pressure sensors," *IEEE Transactions on Industrial Informatics*, vol. 11, pp. 485–494, 2015.
- [6] H. S. Park, D. J. Jung, and H. J. Kim, "Vision-based game interface using human gesture," in *Proceedings of the Pacific-Rim Symposium on Image and Video Technology*, pp. 662–671, Hsinchu, Taiwan, December 2006.
- [7] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," in *Proceedings of the 2014 European Conference on Computer Vision*, pp. 474–490, Zurich, Switzerland, September 2014.
- [8] B. Gleeson, K. MacLean, A. Haddadi, E. Croft, and J. Alcazar, "Gestures for industry intuitive human-robot communication from human observation," in *Proceedings of the 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 349–356, Tokyo, Japan, February 2013.
- [9] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2014.
- [10] T. Starner, J. Weaver, and A. Pentland, "Real-time American sign language recognition using desk and wearable computer based video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1371–1375, 1998.
- [11] O. Mazhar, B. Navarro, S. Ramdani, R. Passama, and A. Cherubini, "A real-time human-robot interaction framework with robust background invariant hand gesture detection," *Robotics and Computer-Integrated Manufacturing*, vol. 60, pp. 34–48, 2019.
- [12] R. A. Rensink, "The dynamic representation of scenes," *Visual Cognition*, vol. 7, no. 1–3, pp. 17–42, 2000.
- [13] P. Neto, D. Pereira, J. N. Pires, and A. P. Moreira, "Real-time and continuous hand gesture spotting: an approach based on artificial neural networks," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation*, pp. 178–183, Karlsruhe, Germany, May 2013.
- [14] W. K. Wong, F. H. Juwono, and B. T. T. Khoo, "Multi-features capacitive hand gesture recognition sensor: a machine learning approach," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 8441–8450, 2021.
- [15] B. Lodhi and J. Kang, "Multipath-densenet: a supervised ensemble architecture of densely connected convolutional networks," *Information Sciences*, vol. 482, pp. 63–72, 2019.
- [16] J. Dolz, K. Gopinath, J. Yuan, H. Lombaert, C. Desrosiers, and I. B. Ayed, "Hyperdense-net: a hyper-densely connected CNN for multi-modal image segmentation," 2018, <https://arxiv.org/abs/1804.02967>.
- [17] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proceedings of the 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 65–72, Beijing, China, October 2005.

- [18] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proceedings of the BMVC 2009-British Machine Vision Conference*, London, UK, September 2009.
- [19] H. Wang, A. Kläser, C. Schmid, and C. Liu, "Action recognition by dense trajectories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3169–3176, Colorado Springs, CO, USA, June 2011.
- [20] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the 2013 IEEE International Conference on Computer Vision*, pp. 3551–3558, Sydney, Australia, December 2013.
- [21] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "A robust and efficient video representation for action recognition," *International Journal of Computer Vision*, vol. 119, no. 3, pp. 219–238, 2016.
- [22] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher vector: theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013.
- [23] V. Kantorov and I. Laptev, "Efficient feature extraction, encoding and classification for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2593–2600, Columbus, OH, USA, June 2014.
- [24] D. Wu, F. Zhu, and L. Shao, "One shot learning gesture recognition from RGBD images," in *Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 7–12, Providence, RI, USA, June 2012.
- [25] T. Y. Lin, P. Goyal, R. Girshick et al., "Focal loss for dense object detection," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.
- [26] Y. Li, Q. Miao, K. Tian et al., "Large-scale gesture recognition with a fusion of RGB-D data based on optical flow and the C3D model," *Pattern Recognition Letters*, vol. 119, pp. 187–194, 2019.
- [27] G. Zhu, L. Zhang, P. Shen, J. Song, S. A. A. Shah, and M. Bennamoun, "Continuous gesture segmentation and recognition using 3DCNN and convolutional LSTM," *IEEE Transactions on Multimedia*, vol. 21, pp. 1011–1021, 2018.
- [28] J. Wan, Q. Ruan, W. Li, and S. Deng, "One-shot learning gesture recognition from RGB-D data using bag of features," *Journal of Machine Learning Research*, vol. 14, pp. 2549–2582, 2013.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [30] Y. Zhang, C. Cao, J. Cheng, and H. Lu, "EgoGesture: a new dataset and benchmark for egocentric hand gesture recognition," *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1038–1050, 2018.