[1,*]Qianqiu Shi

Research on the Application of IU-EKF Algorithm for Feature Extraction of Flute Performers Based on Deep Learning

Journal of Electrical Systems

Abstract: - In order to achieve a more standardized teaching of flute playing gestures, a deep learning based application research on flute player hand shape feature extraction was proposed. Based on the theory of deep learning, a fusion network model based on VGG-16 was designed. The IU-EKF algorithm for deep learning was used to model, pose, learn, and train the hand shape of flute players. Experimental data analysis was conducted on the learning algorithm, and the experimental results showed that the model can extract gesture features for flute performance, whether on the back of the hand. The various angles of the upper and lower joints of the fingers have higher precision recognition than traditional methods. The feature extraction of flute gesture performance can also more effectively recognize the changes in different pitch angles of the gesture at different times, which has important reference value for the standardization of flute hand performance.

Keywords: Flute Performance, Deep Learning, Gestures, Model Research, IU-EKF Algorithm.

I. INTRODUCTION

Artificial Intelligence is currently a very hot area of research. As the theory of AI matures, its areas of application continue to expand, and its methods continue to innovate and evolve. Deep learning is an important branch of machine learning that has received extensive academic and industrial attention since it was proposed by Hinton et al. Initially, deep learning was inspired by human neurological disciplines because the human brain is able to express what it sees, and deep learning has a similar function. Deep learning is better at extracting features than other methods.

Traditional feature extraction methods usually rely on manual completion, whereas deep learning can be active, and with deeper learning, basic features are gradually combined into more complex and semantically expressive features. Deep learning is powerful. Since the origin and development of deep learning is related to neural networks, the network structure of deep learning is similar to the features of neural networks.

Deep learning methods have been the most widely used methods to study image classification problems in recent years, because the network architecture in the field of deep learning has advantages for extracting features from images. Deep learning based flute playing gesture recognition transforms the gesture recognition problem into an image classification problem, and uses the algorithmic framework of deep learning to deepen the correlation between flute playing images and gesture movements based on the deep learning VGG-16 network IU-EKF algorithm, which promotes the improvement of flute playing recognition accuracy.

With the rapid development of deep learning in the past decade, neural networks have become more and more widely used in the field of image and speech. In terms of dealing with long-term series problems, deep learning has obvious advantages over traditional algorithms. Through deep learning, the training of visual images can be realized, and the recognition rate of image objects can be improved, so deep learning algorithms for different neural networks have begun to be widely used [1]. In the field of object vision, deep learning methods have great advantages, at this stage, deep learning has been able to effectively identify static object images, but for dynamic image applications, deep learning is still in a relatively early stage of development, so many scholars have introduced the concept of transfer in the field of deep learning, and through continuous transfer learning, deep learning has gradually developed to a relatively complete stage [2]. Deep transfer learning no longer relies too much on manual feature extraction, and has strong generalization ability, which can usually be applied to multi-feature and multi-domain recognition. The recognition effect of this method is significantly higher than that of other methods.

The traditional teaching method of flute performance is one-on-one teaching through teachers, which makes the financial level of a large number of students unbearable [3]. And the current number of teachers does not allow for one-on-one teaching of a large number of students. Therefore, students can use wearable sensors to obtain the

[1] Guangxi Arts University, Nanning, Guangxi, China
*Corresponding author: Qianqiu Shi

performance status in real time and conduct independent learning, reducing the time spent by teachers and teaching Teachers do not need to supervise students all the time [4-7], but only need to analyze students' practice data through the Internet, so that they can obtain a comprehensive picture of students' learning and training. Teachers can tailor their teaching plans to different students' performance data, enabling students to receive a more targeted approach to teaching [7-10]. Therefore, through an effective recognition method, the gesture dynamics of students playing the flute can be recognized to provide effective help for teachers to teach. For example, Poliner and Ellis [11] studied the recognition of gesture changes through feature action sequences, but this method is used in gesture recognition only can be recognized for specific actions, can not fully recognize the detailed changes of each joint of the hand, and the accuracy of recognition varies greatly; In order to solve the problems of gradient vanishing and gradient explosion, Jurgen Schmidhuber proposed the Long Short-Term Memory Network (LSTM) in 1997), add input and output gates to the input and output layers, and the information can optionally pass through the hidden layer [12]. In 1999, Ger proposed a new forgetting gate, a network structure that allowed LSTM models to handle long-term textures. Both the RNN and LSTM models can only propagate in one direction, and can only use past information, not future information [13].

In this paper, the feature extraction is carried out for the flute gesture performance problem, the model design of the algorithm is carried out based on deep learning, the gestures in the model are effectively extracted at different times and different angles, and the feature extraction experiment is carried out on the algorithm model, and the experimental results show that the model algorithm can effectively and accurately recognize the gestures of the performance, which has important practical significance for the practice and teaching of performance.

II. Gesture Recognition Algorithm Based on Deep Learning VGG-16 Network with IU-EKF

The process of IU-EKF's gesture recognition algorithm based on deep learning VGG-16 network includes the steps of data pre-processing, representation extraction of flute playing gesture image, construction and training of flute playing network model, and application of the model for gesture recognition. First, the original flute playing signal is pre-processed to generate a flute playing signal image suitable for deep learning. Second, the flute playing image is segmented into multiple sub-images according to the multi-stream representation method, which is used as input to the network model. Then, the VGG-16 network model proposed in the paper is constructed and trained. Finally, the trained network model is applied to classify the flute playing signal samples to achieve the recognition of flute playing gesture movements. The overall architecture of the algorithm is shown in Figure 1.

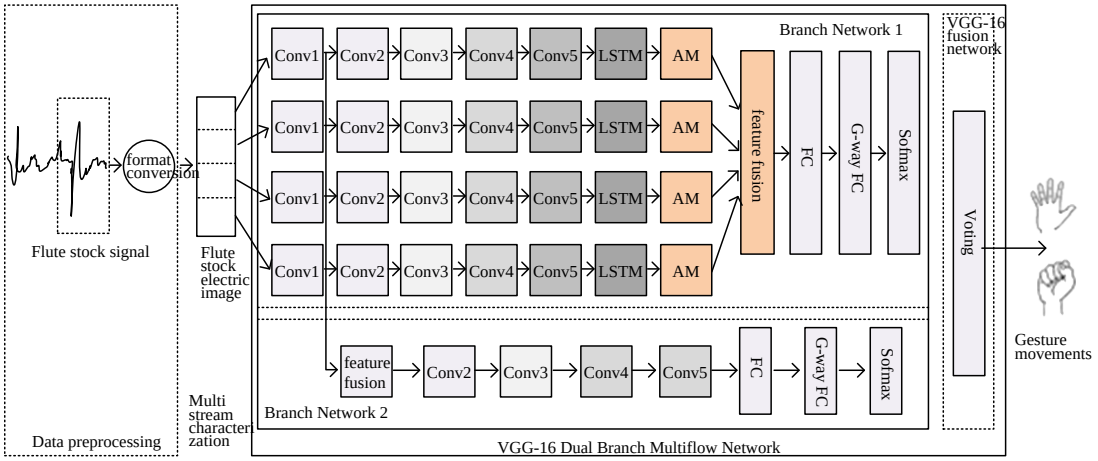


Figure 1: Framework of Gesture Recognition Algorithm for IU-EKF Based on Deep Learning VGG-16 Network

As shown in Figure 1, the network model proposed in this chapter constructs two VGG-16 branching network modules based on deep learning networks by introducing a multi-stream fusion learning strategy. The two branching networks analyse the correlation between flute playing EMG signals and the corresponding flute playing gesture movements from the partition and whole perspectives, respectively. Among them, Branch Network 1 is designed based on a multi-stream parallel architecture, which analyses the implicit correlation between each muscle tissue and the flute gesture action from the partitioning perspective. Branch network 1 stacks convolutional neural network, long and short-term memory network and attention mechanism to extract spatio-temporal features in the flute playing EMG signals, and then the deeper features extracted by multiple streams are used for gesture classification after feature fusion; Branch network 2 analyses the correlation between the EMG signals and the flute playing gesture movements from the perspective of the whole, and after fusing the shallow features extracted by the convolution of multiple streams into a whole, the data is learned with features from the flute playing gesture

movements using a single channel stream. Data for feature learning and flute gesture classification. Based on the construction of two branching network modules, the fusion network structure uses a voting mechanism to determine the final gesture movement category by analysing the outputs of the two branching networks.

The VGG-16 Branching Network1 analyses the implicit correlations between each group of muscle myotomes and flute playing gesture movements from a partitioning point of view in the form of channel streams through a multi-stream parallel architecture. The module uses convolutional neural networks, recurrent neural networks and attention mechanisms to learn the spatio-temporal features in the flute-playing EMG signals, and the deeper features extracted from the multi-streams are used for classification and recognition after feature-level fusion. The constructed branch network model1 is shown in Figure 2, where Conv is the convolutional layer, LSTM is the long and short-term memory layer, AM is the attention mechanism layer, Concatenate is the feature fusion layer, FC is the fully connected layer, G in the G-way fully connected layer represents the number of total classes of gestures to be recognised, and Softmax is the classifier.

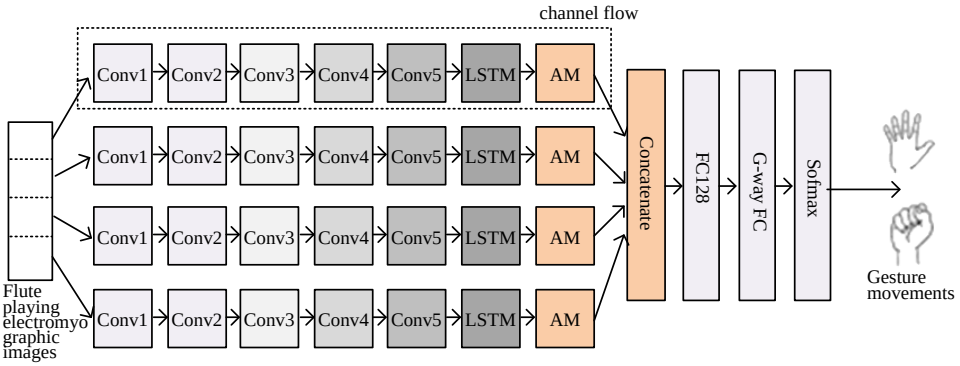


Figure 2: VGG-16 Branch Network 1 Model for Deep Learning

VGG-16 Branching Network 2 analyses the correlation between EMG signals and flute movements from a holistic perspective. After the module fuses the multi-channel shallow features extracted by parallel 1D convolution into a whole, the features are learned and analysed using a single channel stream to give the result of gesture recognition. The constructed Branch Network 2 model is shown in Figure 3, where Conv is the convolution layer, Concatenate is the feature fusion layer, FC is the fully connected layer, G in the G-way fully connected layer represents the total number of gesture categories to be recognised, and Softmax is the gesture classifier.

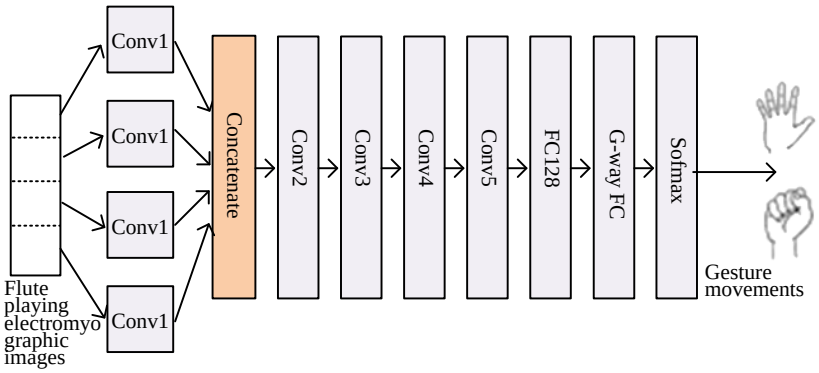


Figure 3: VGG-16 Branch Network 2 Model for Deep Learning

The Branch Network2 module consists of a parallel 1D convolutional layer, a feature fusion layer, four 1D convolutional layers, two fully connected layers and a softmax classifier. The number of parallel 1D convolutional layers is equal to the number of input myoelectric images, which are batch normalised and connected to a maximum pooling layer. The feature fusion layer performs a stitching operation on the features output from each channel stream to generate a new feature map, which is then used in the subsequent convolutional layers for feature extraction. After the feature fusion layer, four 1D convolutional layers are connected, which are similar in structure, and the size of the convolutional kernel becomes smaller as the layers deepen, while the number of convolutional kernels continues to increase. Among them, the first two 1D convolutional layers use batch normalisation and are connected to the maximum pooling layer, and all convolutional layers use ReLU as the activation function, which is designed to reduce overfitting and accelerate the convergence speed of the model. After the convolutional layers, the module cascades 2 fully connected layers for dimensional transformation. Finally, the network module uses

Softmax as a classifier, which receives the output features from the fully connected layers and classifies the gestures based on the extracted features and outputs the recognition results.

III. Deep Learning Flute Performance Model Training

Deeplearning4J is a deep learning framework open-sourced and maintained by Sky mind, which natively supports distributed model training due to its JVM-based features.

In recent years, major internet companies have released their own deep learning frameworks, including Google's Keras, Facebook's Torch, BVLC's Caffe, and Baidu's PaddlePaddle. However, these frameworks are all based on Python or C/C++ [11-12]. While frameworks are efficient, most IT companies, especially domestic ones, tend to build applications in the Java ecosystem of open source projects like Spring and Structs. In addition, many distributed frameworks, including Hadoop and Spark, run on the JVM. When it comes to data storage, Hive, HDFS, HBase, and other storage media are often used. These may include Hive, HDFS, HBase, and more. As a result, the Deeplearning4J framework is suitable for adapting to the domestic environment and enabling rapid development and deployment of production projects.

Deeplearning4J is based on the theory of data parallelization, which makes distributed modeling possible. Model parallelization can be used to train multi-layer neural networks hierarchically, that is, the parameters of part of the network layer are concentrated on a node in the cluster for training, and the parameters between each node are scheduled through a scheduler [14-18]. Data parallelization can save a copy of the network model on each compute node, and train their own batch data separately, and then update the global network parameters according to the synchronous or asynchronous mechanismDeeplearning4J mainly uses data parallelization solutions, Deeplearning4J currently supports the following two data parallelization strategies: parameter synchronization average scheme and decentralized gradient sharing scheme [19-24]. Figure 4 shows the schematic diagram of the synchronous average parametric scheme.

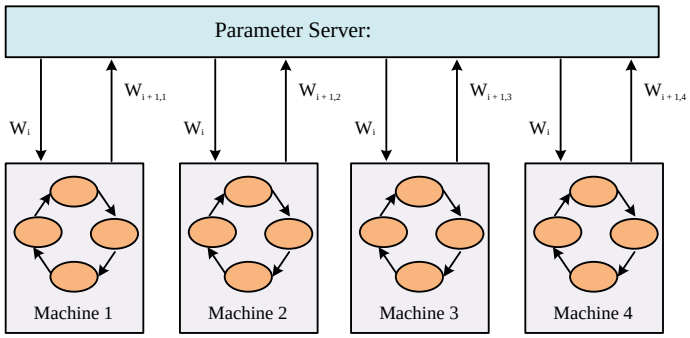


Figure 4: Schematic Diagram of the Server Parameter Synchronization Average Scheme

Define a parameter server instance in Driver, which collects training parameters from multiple nodes, uses the weighted average algorithm to calculate the updated model parameters, and then makes the updated parameters into broadcast parameters and passes them to each node. It should be noted that the Parameter Server must receive all node parameters before the weighted average update of the model parameters, but the network latency of each node is different, and the slowest node will become the bottleneck of the entire cluster. The schematic of the decentralized gradient sharing scheme is shown in Figure 5.

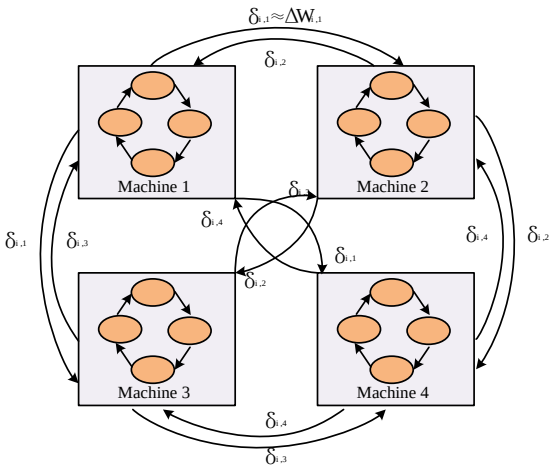


Figure 5: Schematic Diagram of a Decentralized Gradient Sharing Scheme

This is an asynchronous gradient update scheme in which multiple nodes are connected in pairs and can update parameters to each other. In order to solve the bottleneck problem in network transmission, the decentralized gradient sharing scheme defines a threshold for each gradient change, and updates the gradient parameters only when a certain gradient change is greater than the threshold.

IV. Performance Gesture Recognition Technology Based on Deep Learning

A. Flute Performance Feature Recognition Based on Deep Learning

The extracted multimodal features are fed into the VGG-16 network to complete feature migration in the isomorphic space. Compared to other neural network algorithms, the VGG-16 algorithm has a higher depth. Figure 1 illustrates the structure of the VGG-16 network, into which an RGB color image of feature extraction is fed. The VGG-16 network consists of 13 convolutional layers, outperforming other networks in achieving accurate data convolution. In addition, the network consists of 3 fully connected layers and 5 downsampling layers. The maximum pooling technique is used to implement downsampling in convolution.

The VGG-16 network uses a 3 × 3 convolutional filter in the computational process. This approach can identify local features more deeply, improve the ability to determine functions, and reduce the total number of parameters. In equations (1) and (2), the weight of the l-layer convolution kernel is determined by $\ker nel_{ij}^{i}$. The input sample is xj, and the input sample for layer l is x_j . The number of layers is l, and there is a causal relationship between the layers, resulting in a logical flow of information.

$$x_j^i = f'\left(\sum_{i\in M_j} x_j^{i-1} \times \ker nel_{ij}^i + b_j^i\right) \tag{1}$$

$$\delta = \beta_j^{i+1}(f'(\mu_j^i).up(\delta_j^{i+1})) \tag{2}$$

The convolutional layer in deep learning consists of the input feature map of the previous layer, represented as x_j^i . The offset term for each layer is b_j , and the offset term for layer l is b_j^i . The activation function permutation matrix is f', and the error of the convolutional layer at layer l is δ_j . The output of the convolutional layer is accompanied by the output of the β_1 convolutional layer in the downsampled layer β_j^{i+1} and the characteristics of the input samples μ_j . $i+1$ The backpropagation output of the downsampled layer is $1_{n\times n}$. Assuming that the input sample in the VGG-16 network is $1_{n\times n}$, it becomes easy to update the weights. $i+1$ The error of the backpropagation output neuron $i+1$ in the downsampled layer is while the state matrix δ_j of the input samples in the upsampled layer $1_{n\times n}$ is , and equation (3).

$$up(x) = x_j \otimes 1_{n\times n} \tag{3}$$

In a VGG-16 network, for weight updates to be implemented, a convolutional layer must be connected to the downsampled layer l+1. According to the VGG-16 network principle, the sensitivity of each neuron determines the network error. The upsampling technique is denoted by up, while the element multiplication is denoted. Equations (4) and (5) calculate:

$$\frac{\alpha E}{\alpha b_j} = \sum_{\mu.r}(\delta_j)\mu \tag{4}$$

$$\frac{\alpha E}{\alpha b_{ij}^i} = \sum_{\mu.r}(\delta_j)\mu\left(x_j^{i-1}\right)\mu.r \tag{5}$$

In equations (4) and (5) above, the error of the training sample is expressed as E, the neuronal error of the backpropagation output at layer l is expressed as δlj, and the position coordinates of the output convolutional layer are expressed as μ and r. The downsampling process is calculated using equation (6), where down represents the downsampling function. Use equation (5) to calculate the sensitivity of the base, and then use equations (7) and (8) to calculate the gradient of the weights of the downsampled layer.

$$x_j = f'(\beta_j^{'} down(x_j^{i-1} + b_j^i) \tag{6}$$

$$d_j^i = down(x_j^{i-1}) \tag{7}$$

$$\frac{\partial E}{\partial \beta_j} = \sum_{\mu.r} \left(\delta_j^i . d_j^i\right) \mu.r \tag{8}$$

In equations (7) and (8), the output of the convolutional layer is βlj, and the error during downsampling is DLJ. When there are obvious data differences between neighbors, deep learning based on homogeneous spaces can recognize various gesture features. Migrating these features from the target dataset ensures that they can be more fully applied to enhance the recognition of performance gestures.

B. Gesture Recognition Model for Flute Performance Based on Transfer Learning

In this study, a deep transfer learning model based on VGG-16 network was used to recognize gestures during performance. The architecture of the model is shown in Figure 6, which shows the deep transfer learning of features in the target dataset after the VGG-16 network is trained. This approach generates a deep transfer learning model rooted in the VGG-16 network, which is a key factor in ensuring the full application of gesture features in the flute performance recognition process. The VGG-16 network is shown on the right and the migration model is shown on the left. The convolutional filter size for the migration model on the left remains the same at 3x3. These fragments make up the deep transfer learning model. The sections within the model are consistent in composition and order, and are divided into two parts: the first 13 layers consist of a relocated convolutional layer and a downsampled layer, and the last 3 layers consist of a relocated, fully connected layer.

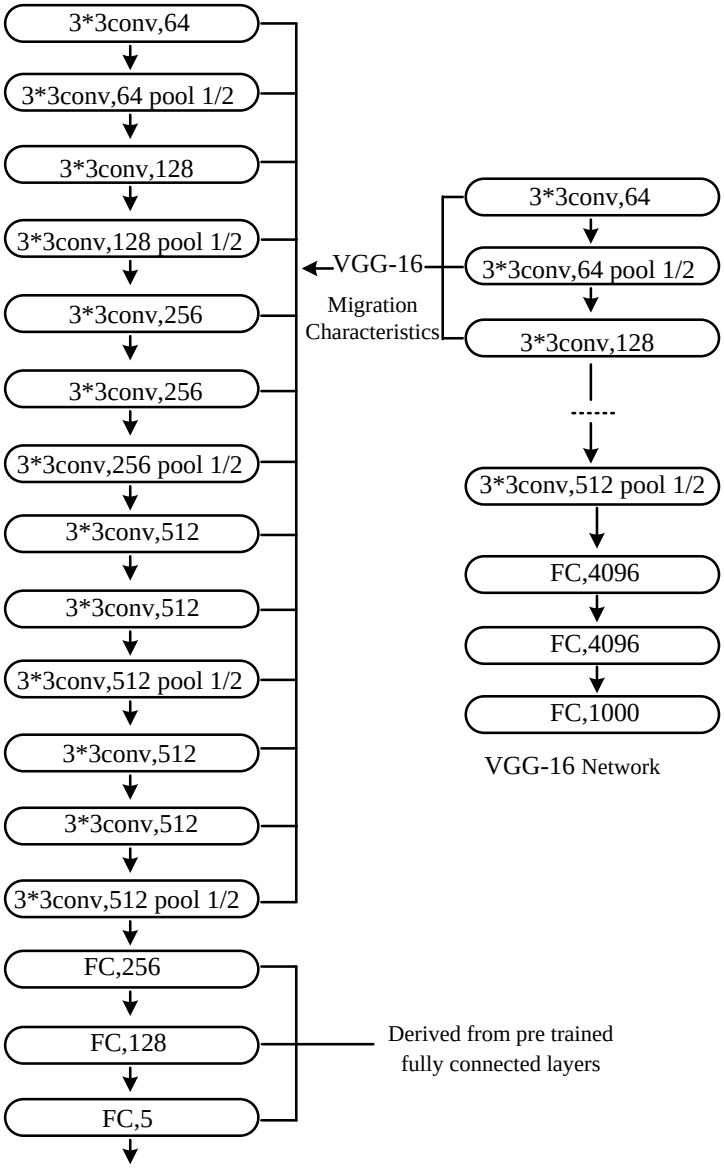


Figure 6: A Flute Playing Network Model for Deep Learning

V. FLUTE PERFORMANCE GESTURE RECOGNITION FEATURE EXTRACTION TECHNOLOGY

In this paper, we introduce a performance gesture recognition technology that uses deep learning and multimodal features to enhance the recognition effect. The process includes using:

(1) Micro-inertial sensors and infrared detectors to collect data on flute performance gestures, so as to obtain gesture data on different parts of the hand. The state-space model estimates the gestures and then fuses the gestures using the IU-EKF algorithm.

(2) In order to obtain valuable gesture data from the moving part of the hand playing the flute, the detection data of the infrared detection strip is obtained through a sliding window of fixed width. This results in a multimodal feature of the flute-playing gesture, as shown in Figure 2, which is based on a deep transfer learning model using the VGG-16 network.

(3) The first step is to use the infrared detection rod to classify the extracted multimodal feature data. Then, the deep learning machine model (VGG-16) was used to train the feature model on four gesture features, namely time-domain statistical features, interdigital coupling features, spatial eigenvalues and auxiliary features, so as to achieve accurate classification and recognition of flute playing gestures.

A. Gesture Estimation and Pose Determination for Flute Playing Based on State Space Model

In this paper, we utilize inertial sensors from microelectromechanical systems (MEMS) to capture hand movement data when the flute plays gestures. Designing a state-space model for a performance gesture requires a comprehensive description of the pose. This can be achieved through quaternions and other suitable methods. The concept of Euler angle is understandable, but the presence of its "singularity" leads to incomplete gesture estimation. Utilizing directional cosine matrices requires a lot of computation and is therefore impractical, so in this paper we use quaternions for gesture descriptions and the accompanying parameters as the state of the sensor system. The computational complexity required to exploit directional cosine matrices makes it impractical, so in this paper quaternions are used for gesture description, and the subsequent parameters are taken as the state of the sensor system. Utilizing directional cosine matrices requires a large number of calculations and is therefore impractical, so a quaternion is used for gesture description, and the subsequent parameters are taken as the state of the sensor system. Equation (9) illustrates this state.

$$x_i = \left[q_e^T \; v_e^T \; b_{g,e}^T \; b_{a,e}^T \right] \tag{9}$$

In equation (9), the unit quaternion of the gesture gesture is $q_e = \left[q_{0,e} \; v_{1,e} \; b_{2,e} \; b_{3,e} \right]^T$; The velocity vector of the download body is $v_e = \left[q_{east,e} \; v_{north,e} \; b_{up,e} \right]^T$, and the velocity components along the sky, east, and north directions in the navigation coordinate system v_{up} , v_{east} , v_{north} are represented by ;The accelerometer offset is $b_e = \left[b_{ax,e} \; b_{ay,e} \; b_{az,e} \right]^T$; The gyroscope drifts to $b_{g,e} = \left[b_{gx,e} \; b_{gy,e} \; b_{gz,e} \right]^T$. T is the transpose mark. According to the quaternion principle, the relationship between the attitude quaternion and the carrier angular velocity vector w can be determined as shown in equation (10):

$$\dot{q} = \frac{1}{2}\Omega(w) q_e = \frac{1}{2} \begin{bmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & -w_z & -w_y \\ w_y & w_z & 0 & -w_x \\ w_z & w_y & w_x & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{10}$$

In equation (10), the antisymmetric matrix of the carrier angular velocity vector is Ω (w), and the elements of the antisymmetric matrix of the carrier angular velocity vector are used by

w_x , w_y , w_z Representation, the elements of the matrix after the transposition of the unit quaternion of the gesture pose are denoted q_0 , q_1 , q_2 , q_3 by the gyroscope output by w = w-bg-ηb, w, and the gyroscope measurement noise is denoted by ηb.

The strapdown inertial derivative force equation is expressed by equation (11):

$$v = R_b^n f^b + G_0 \tag{11}$$

In equation (11), the unit quaternion R_b^n describes the rotation matrix from the flute-playing gesture coordinate system to the navigation coordinate system. In addition, the compensating offset ratio in the flute playing gesture coordinate system is fb, where G0 represents the gravitational acceleration vector. As shown in Equation (12).

$$R_b^n = 2\begin{bmatrix} 0.5- & q_2^2- & q_3^3q_1q_2- & q_0q_3q_1q_3+ & q_0q_2 \\ q_1q_2+ & q_0q_3 0.5- & q_1^2- & q_3^2q_2q_3- & q_0q_1 \\ q_1q_3- & q_0q_2q_3+ & q_0q_1 0.5- & q_1^2- & q_2^2 \end{bmatrix} \tag{12}$$

In Equation (12), it can be $q = 9.82 m.s^2$ calculated by Equation (13):

$$f^b = f^b \tilde{} - b_{a,e} - \eta_a \tag{13}$$

In Equation (13), the value obtained when taking an acceleration measurement $\tilde{f}^b$, noise η_a.

Gyroscopes and accelerometer offsets are used for modeling, and first-order Markov models are constructed, such as show equations (14) and (15):

$$b_g = \frac{1}{\eta_g} b_{g,e} + \eta_{b,g} \tag{14}$$

$$b_g = \frac{1}{\eta_a} b_{a,e} + \eta_{b,a} \tag{15}$$

In equations (14) and (15), the first-order Markov models of gyroscope and accelerometer bias are b, respectively·g, b·a, the relevant time is represented η_a by η_g, Gaussian white noise is represented in turn $b_{g,e}, \eta_{b,a}$.

B. Flute Performance Gesture Fusion and Posture Fixing Based on IU-EKF Algorithm Based on Deep Learning

Using the above pose estimation model and the IU-EKF algorithm, perform the following steps to achieve pose fixation for flute playing gestures.

After obtaining the pose estimation measurement data, the measurement data was updated with an N-step pseudo-time, where N was set to 5. The Kalman gain for each update is shown in equation (16) at $i = 1 \to N$.

$$K_k^{(i)} = \frac{1}{N}(p_k^{(i-1)}) H_k^{(i)T} + c_k^{(i-1)} (w_k^{(i)})^{(-1)} \tag{16}$$

In Eq. (16), each parameter is defined as follows $w_k^{(i)}$: a Jacobian matrix for the state vector $H_k^{(i)}$, and a Jacobian matrix for the distance measurement function $p_k^{(i-1)}$ is the Jacobian matrix of the input noise, and $c_k^{(i-1)}$ is the covariance matrix of the system noise.

$$w_k^{(i)} = H_k^{(i)} p_k^{(i-1)} H_k^{(i)T} + R_k + H_k^{(i)} c_k^{(i-1)} + c_k^{(i-1)} H_k^{(i)T} \tag{17}$$

$$H_k^{(i)} = \begin{bmatrix} \dfrac{\partial h_1}{\partial_{q_k^{(i)}}} & 0_{3\times 3} \\ \dfrac{\partial h_2}{\partial q_k^{(i)}} & \dfrac{\partial h_2}{\partial v_k^{(i)}} \end{bmatrix} \tag{18}$$

In equations (17), (18), Rk is the measurement noise covariance matrix, which is the $v_k^{(i)}$ measurement noise, is the process noise $q_k^{(i)}$, is $h_1, h_2\ q_k^{(i)}$ and of the $v_k^{(i)}$ transfer function.

When the step I measurement is updated, the posterior estimation of the model state is shown in equation (19). The posterior error covariance is shown in equation (19).

$$\hat{x}_k^{(i)} = \hat{x}_k^{(i-1)} + K_k^{(i)} (y_k - h(\hat{x}_k^{(i-1)}) \tag{19}$$

In equation (19), $h(.)$ the measurement function of the nonlinear system is the measurement function of the nonlinear system, the measurement function of the nonlinear system is the state estimation vector, and $\hat{x}_k^{(i)}$ the y_k noise variance is measured.

$$p_k^{(i)} = (I_{n\times n} - K_k^{(i)} H_k^{(i)}) p_k^{(i-1)} (I_{n\times n} - K_k^{(i)} H_k^{(i)})^T + K_k^{(i)} H_k^{(i)T} - (I_{n\times n} - K_k^{(i)} H_k^{(i)}) \quad (20)$$

In equation (20) above, is the $I_{n\times n}$ system discrete state matrix.

C. Modeling and Extraction Methods of Flute Playing Gesture Features

The micro-inertial sensor can obtain the change information of the flute playing gesture in real time, providing time-dimensional information. Due to the variability of flute gestures and the large range of movements, a single feature parameter cannot effectively and accurately capture the characteristics of hand playing. Therefore, in this paper, in order to extract the features of the gesture data collected by the inertial sensor, the multimodal feature extraction of the piano gesture is explored, and various forms of coupling features of the gesture data obtained in the previous steps are extracted. In addition, an infrared detection rod was installed during the flute performance during the modeling to extract the coupling features of the different forms of the detected hands. Different forms of coupling features are extracted from the gesture data obtained in the above steps. In addition, an infrared detection rod is installed during flute performance to extract the gesture data detected by the detection rod, which is then used as auxiliary information for additional features. The steps for performance gesture extraction are as follows:

1) Statistical characteristics in the time domain: In flute playing, the movement of the player's fingers will change significantly, resulting in a corresponding change in the amplitude of the movement of the back of the hand during playing. In order to analyze these hand movements from different perspectives, this paper considers the standard deviation and range of gestures, as well as the differences before and after keystrokes.

2) Extract the dynamic information of the fingers and the back of the hand, and calculate the difference between the angle of the hand posture and the angle of the finger joints when pressing the key.

3) Coupling characteristics between fingers: There are also certain differences between the different fingers of the performer during daily performance [13], so the acceleration and angular velocity data between adjacent fingers are extracted in this paper.

3) Coupling features between fingers: This study extracted acceleration, angular velocity, and other relevant data between adjacent fingers because of the obvious differences in their daily performance.

4) Accessibility: The infrared detection strip can effectively detect each key in real-time, so that the finger movements in the current performance can be analyzed based on this information. Due to the specific time difference of hand movements during performance, this paper adopts a sliding window method with a fixed time width to manage the detection data. The time window width is set to 100 milliseconds, and the window width is used to extract data features. These features are then used as auxiliary features for gestures. In this paper, the above features are standardized in order to achieve them more effectively in the identification process. This can be expressed by equation (21) as follows.

$$\mathrm{p}_{\text{new-i}} = \frac{p_{new-i} - p_{\min}}{p_{\max} - p_{\min}} \quad (21)$$

In equation (21), the result of normalization is described, the maximum value of the feature is described, the p_{new-i} minimum value of the feature is denoted and $p_{\max}$ the $p_{\max}$ feature dimension is described p_{new-i} 。 Through the modeling and extraction method of flute playing gesture features, the recognition of flute playing gestures can be realized.

VI. Analysis of Experimental Results

In this paper, the performance data of five flute pieces, including "Venice Carnival" and "Cuchulainn", were collected through simulation experiments A total of 420 performance samples were obtained by playing these five songs three times, and these samples were divided into training sets and test sets for testing. The action features of different fingers in the process of performance were extracted, and the method of this paper was used to analyze the ability of flute performance gesture feature extraction. Table 1 shows the results of the analysis of feature extraction ability of different finger movements.

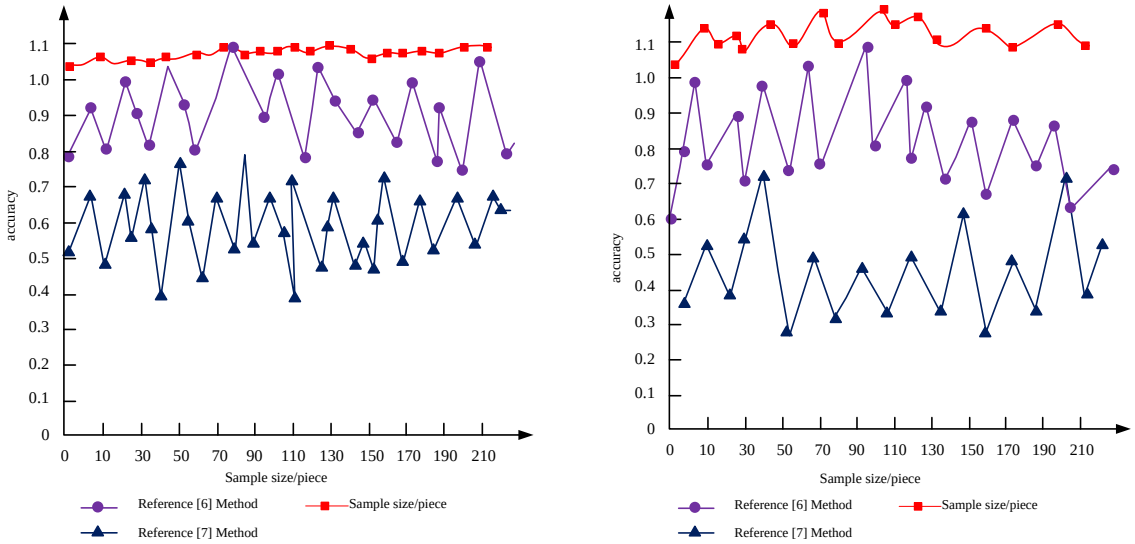
Table 1: Analysis of the Ability to Extract Features from Different Finger Movements of Flute Playing by Deep Learning

feature extraction		finger				
		Thumb	Index finger	Middle finger	Ring finger	Little finger
	Standard deviation of pitch angle	√	√	√	√	√
	Extremely poor pitch angle	√	√	√	√	√
the back of the hand	Standard deviation of roll angle	√	√	√	√	√
	Extreme roll angle difference	√	√	√	√	√
	Acceleration standard deviation	√	√	√	√	√
Inferior joint of fingers	Standard deviation of angular velocity	√	√	√	√	√
	Angular velocity range	√	√	√	√	√
	Pitch angle button front and rear difference	√	√	√	√	√
	Extremely poor pitch angle	√	√	√	√	√
	Standard deviation of roll angle	√	√	√	√	√
	Extreme roll angle difference	√	√	√	√	√
Superior joint of fingers	Acceleration standard deviation	√	√	√	√	√
	Standard deviation of angular velocity	√	√	√	√	√
	Angular velocity range	√	√	√	√	√
	Standard deviation of angular velocity	√	√	√	√	√
	Angular velocity range	√	√	√	√	√

According to Table 1, it can be seen that by applying the method of this paper, the playing features of each position of the hand can be effectively extracted, and the standard deviation and range of each joint in the playing process can be extracted, and the recognition of the movement features of each finger can be effectively completed, and there is no obvious error in the recognition process, which indicates that the method of this paper can better extract the change characteristics of each joint of the hand in the hand playing gesture recognition.

In this paper, the dynamic gesture recognition method based on feature action sequences in Ref. [6] and the CSI gesture recognition method using LSTM in Ref. [7] are selected and compared with the methods proposed in this paper. The recognition accuracy of different methods was analyzed and compared by identifying on different training and test sets, and the results are shown in Figure 7.

As shown in Figure 7, in this study, the recognition accuracy of flute playing gestures is very high, indicating a significant improvement in recognition after dataset training, learning and retesting to the VGG-16 network. At the same time, the recognition rate of the method [7] has been low. Although the recognition rate of the method literature [6] is slightly higher than that of the method literature [7], there is still a gap compared with the method used in this study. Under different sample sizes, the recognition rate of the above method is better than that of the method literature [6] and the method literature [7], which indicates that the application of this method greatly improves the accuracy of flute gesture recognition.



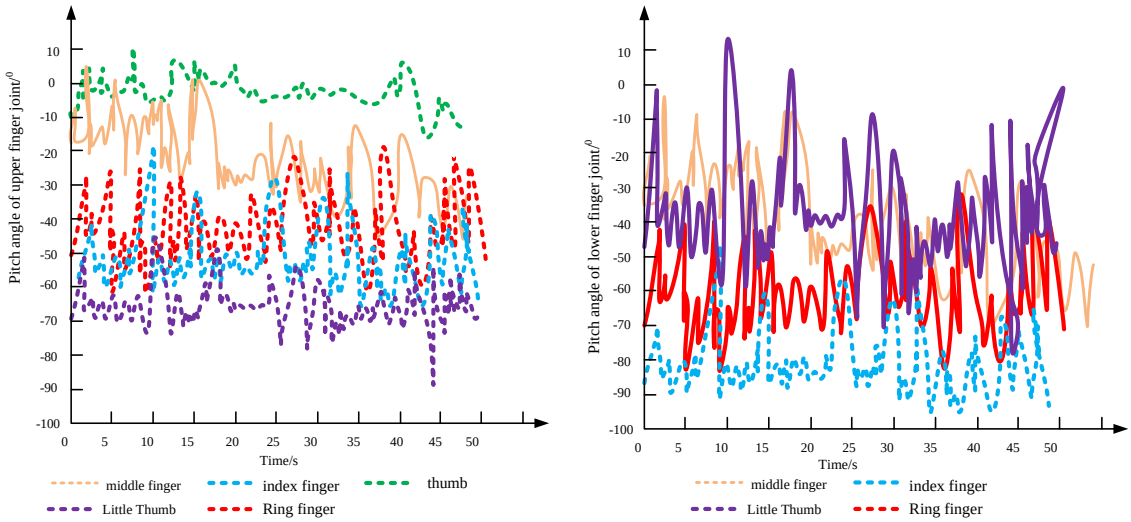
(a) Training Concentration and Precision (b) Test Set Accuracy

Figure 7: Two Datasets Identify Effect Analysis

Figure 8 shows the effect of gesture recognition during the “Venice Carnival” performance, as well as the change in the elevation angle of the finger joints throughout the recognition process.

Figure 4 illustrates how this technique can effectively distinguish data related to flute playing gestures, revealing the dynamics of hand movements. Figures 4(a) and 4(b) show the precise fluctuations in the pitch angles of the upper and lower knuckles over time, illustrating the variation of the player’s finger position at different stages

of performance, emphasizing fine-grained identification. The method in this paper can effectively identify the flute performance gesture fluctuations and produce clear recognition results.



(a) Change in Pitch Angle of the Upper Finger Joint (b) Change in Pitch Angle of the Subfinger Joint

Figure 8: The Flute Plays the Curve of the Hand

VII. CONCLUSION

The recognition of flute playing gestures can help in several areas such as school teaching and student learning. During flute playing, the player's gestures vary widely and at a fast speed, so a high degree of recognition is required. At present, most gesture recognition methods are not suitable for the field of flute playing, so it is necessary to design a more effective and convenient recognition method to capture the performance process more accurately. Therefore, it is imperative to develop more efficient and convenient technology to recognize gesture changes during flute playing, so as to capture and analyze performance data more precisely. The focus of this research is to study flute gesture recognition technology based on deep learning. The focus of this study is to explore the flute gesture recognition technology based on deep learning. The project consisted of capturing flute playing gestures to extract features and laying the foundation for the recognition of these gestures. To achieve this, we employ a deep transfer learning approach. In addition, a performance dataset is used to verify whether the proposed method can effectively identify the changes of flute gestures during performance. The focus of this study is to examine a flute gesture recognition technology based on deep learning. Experimental results show that this method can produce excellent recognition results and is not limited to flute playing.

REFERENCES

[1] Zaharia M A, Chowdhury M, Franklin M J, et al. Spark: Cluster Computing with Working Sets, Book of Extremes, 2010, 15(1): 1765-1773.

[2] Chen Yin-state Evaluation method and system of piano performance. China, 201612(3):11-18.

[3] Qi J X, Jiang G Z, Li G F, et al. Surface EMG hand gesture recognition system based on PCA and GRNN. Neural Computing & Applications, 2020, 32(10): 6343-6351.

[4] Ding Qichuan, Xiong Anbin, Zhao Xingang, et al Research and Application Review of Motion Intention Recognition Methods Based on Surface Electromyography. Journal of Automation, 2022, 42 (1): 13-25.

[5] Thiruvengatanadhan R, Dhanalakshmi P. Indexing and retrieval of music using Gaussian mixture model techniques. International Journal of Computer Applications, 2016, 148(3).

[6] Xu H, Xiong A B. Advances and Disturbances in EMG-Based Intentions and Movements Recognition: A Review. IEEE Sensors Journal, 2021, 21(12): 13019-13028.

[7] Guo G, Li S Z. Content-based audio classification and retrieval by support vector machines. IEEE transactions on Neural Networks, 2023, 14(1): 209-215.

[8] Jamal M Z. Signal acquisition using surface EMG and circuit design considerations for robotic prosthesis. Computational Intelligence in Electromyography Analysis-A Perspective on Current Applications and Future Challenges, 2022, 18: 427-448.

[9] Li K X, Zhang J H, Wang L F, et al. A review of the key technologies for EMG-based human-robot interaction systems. Biomedical Signal Processing and Control, 2020, 62: 1-17.

[10] Li W, Shi P, Yu H L. Gesture Recognition Using Surface Electromyography and Deep Learning for Prostheses Hand: State-of-the-Art, Challenges, and Future. Frontiers in Neuroscience, 2021, 15: 20.

[11] Poliner G E, Ellis D P W. A discriminative model for polyphonic piano transcription. EURASIP Journal on Advances in Signal Processing, 2023, 20(07): 1-9.
[12] Liu Ying, Lei Yanbo, Fan Jiulun, et al Overview of Image Classification Techniques Based on Small Sample Learning. Journal of Automation, 2021, 47 (02): 297-315.
[13] Kong Q, Li B, Song X, et al. High-resolution Piano Transcription with Pedals by Regressing Onset and Offset Times. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021, 29(5): 3707-3717.
[14] Nakamura E, Ono N, Sagayama S, et al. A stochastic temporal model of polyphonic MIDI performance with ornaments. Journal of New Music Research, 2019, 44(4): 287-304.
[15] Nakamura E, Ono N, Saito Y, et al. Merged-output hidden Markov model for score following of MIDI performance with ornaments, desynchronized voices, repeats and skips. algorithms, 2022, 21(3): 23-29.
[16] Lashgari E, Liang D H, Maoz U. Data augmentation for deep-learning-based electroencephalography. Journal of Neuroscience Methods, 2020, 346: 1-26.
[17] Wei W T, Dai Q F, Wong Y K, et al. Surface-Electromyography-Based Gesture Recognition by Multi-View Deep Learning. IEEE Transactions on Biomedical Engineering, 2021, 66(10): 2964-2973.
[18] Hu Y, Wong Y K, Dai Q F, et al. sEMG-Based Gesture Recognition With Embedded Virtual Hand Poses Adversarial Learning. IEEE Access, 2020, 7(10): 4108-4120.
[19] Atzori M, Gijsberts A, Castellini C, et al. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. Scientific Data, 2022, 1(3): 1-13.
[20] Gijsberts A, Atzori M, Castellini C, et al. Movement Error Rate for Evaluation of Machine Learning Methods for sEMG-Based Hand Movement Classification. Ieee Transactions on Neural Systems and Rehabilitation Engineering, 2021, 22(4): 735-744.
[21] Zhang Kezhi, Wei Guoqiang, Feng Ze, et al. Research on the Application of Deep Learning Technology in Intelligent Image Processing . Modern Information Technology, 2021, 5(10): 15-26.
[22] Zou Qi, He Yueshun, Yang Xi, et al. Construction of a logging lithology recognition model based on ensemble learning. Intelligent Computer and Application, 2020, 10 (3): 91-94.
[23] Qiao Lianhua, Liu Minshi. Adaptive deep learning classification based on fusion of urban street tree feature selection model. Surveying and Mapping Bulletin, 2020, (6): 77-80.
[24] Li Na, Gu Qing, Jiang Feng, et al. A feature representation method for sandstone microscopic images based on convolutional neural networks. Journal of Software Science, 2020,31 (11): 3621-3639.