



上海科技大学
ShanghaiTech University

THESIS

Subject: Multi-sensor Gesture Recognition and Human-computer
Interaction Graphic Control

Student Name: Chengqi Liu

Student ID: 2018533249

Year of Entrance: 2018

School: School of Information Science and Technology

Major: Computer Science and Technology

Advisor: Yuyao Zhang

ShanghaiTech University

Date: 06 / 2022

ShanghaiTech University

Statement of Academic Integrity of Bachelor's Thesis

I solemnly declare that the Bachelor's Thesis (Design) submitted is the result of my independent work under the guidance of my faculty advisor. This paper does not contain any work that has been published or written by any other individual or group, except for the content that has been quoted in the text. Individuals and groups that have made important contributions to the research of this article have been clearly identified in the text. I am fully aware that I am obligated to undertake the legal consequence of the statement.

Author's Signature:



Date: 01/06/2022

ShanghaiTech University

Copyright Authorization of Bachelor's Thesis

The author of the Bachelor's Thesis (Design) agrees that ShanghaiTech University will retain and send the copy and electronic version of the thesis to the relevant national departments or institutions, allowing the thesis to be consulted and borrowed. I authorize ShanghaiTech University to compile all or part of the contents of this graduation thesis (design) into the relevant database for retrieval and can save by means of photocopying, reduction or scanning.

The Bachelor's Thesis is

Confidential☐. This authorization shall apply after ____ years.

Public☒.

Author's Signature:



Date: 01/06/2022

Advisor's Signature:



Date: 01/06/2022



MULTI-SENSOR GESTURE RECOGNITION AND HUMAN-COMPUTER INTERACTION GRAPHIC CONTROL

ABSTRACT

Human-computer interaction (HCI) is a computer technology that studies the interaction between users and computers through various interfaces. Its purpose is to create a more natural and direct user interface to increase the ease of use of computer products. The HCI using multi-sensor gesture recognition is a promising human-computer interaction technology. It is widely used and will not cause trauma to users. Pattern recognition is an effective way to detect small-scale gestures through sensors and extract a large amount of information from them. Compared with the traditional single-view deep learning algorithm, the multi-view convolutional neural network (CNN) can extract highly discriminative features from the data, so as to improve classification accuracy. On the basis of the existing research, we collected the surface electromyographic signals (sEMG) of the upper arm of the subjects by using a variety of sensors carried by the gForcePro+ armband produced by OYMotion Technology, and constructed five data sets based on different gesture schemes. We then design a multi-view CNN model (based on Python 3.9, Pytorch 1.10, and CUDA 11.3) for gesture recognition, select the most appropriate gesture scheme, and finally implement a user-friendly HCI graphics control software. The software, developed based on Qt 5.15.2 and MSVC 2019-64bit, will facilitate surgeons to adjust medical images while manipulating surgery robot, and can also be widely used in other application scenarios.

Key words: HCI, pattern recognition, sEMG, multi-view CNN, gesture recognition



Catalog

Chapter 1	Introduction	1
1.1	Human-Computer Interaction	1
1.2	Pattern Recognition	2
1.2.1	Pose Estimation	2
1.2.2	Gesture Recognition	2
1.3	Description of Experiment	3
1.3.1	Scenarios and Requirements	3
1.3.2	Workflow	4
1.3.3	Difficulties	5
1.3.4	Final Effect	5
Chapter 2	Materials and Methods	7
2.1	Construction of EMG Data Set	7
2.2	Model Training	11
2.3	Software Building	12
2.3.1	Real-Time Data Acquisition Part	12
2.3.2	Gesture Recognition Part	12
2.3.3	Graphic Control Part	13
2.3.4	Inter-Process Communication	14
Chapter 3	Result	15
Chapter 4	Discussion	17
4.1	Discussion on Experiment	17
4.1.1	Workflow	17
4.1.2	Gesture Scheme	17
4.1.3	Acquisition	17
4.1.4	Sliding Window Size	18
4.2	Improvement Directions	18
4.2.1	Adding Accelerometer Data	18
4.2.2	Increasing Sampling Rate	18
4.2.3	Optimizing Inter-Process Communication	18
4.2.4	Combining with Attitude Estimation	19
Chapter 5	Conclusion	20
References		21
Acknowledgements		23



Chapter 1 Introduction

1.1 Human-Computer Interaction

Human-computer interaction (HCI) is a computer technology that studies the interaction between users and computers through various interfaces. Its purpose is to create a more natural and direct user interface to increase the usability of computer products [1] (Fig. 1-1). Although the traditional human-computer interaction paradigm based on WIMP (window, icon, menu, and pointer) has been widely adopted, with the miniaturization of computer equipment and the diversification of use functions, various new HCI methods have also been proposed [1].

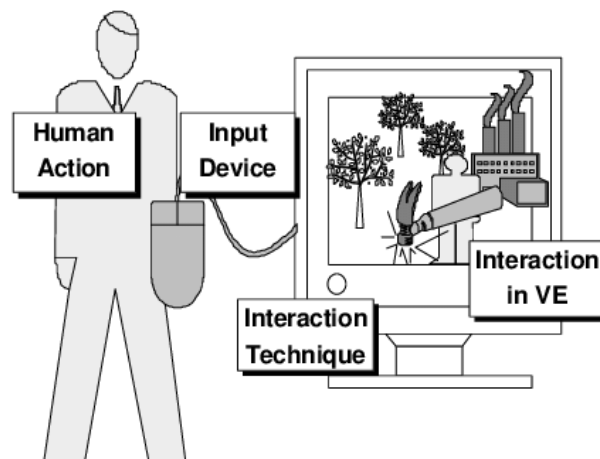


Fig. 1-1 Principal of HCI

HCI using multiple sensors for gesture recognition has broad prospects. This technology can use a variety of sensors (such as cameras, electrodes on the skin, and accelerometers) to collect signals and recognize the user's current gestures to control the computer and perform corresponding functions. This interactive technology is easy to use and will not cause injury. It can be designed as an armband or other accessories that can be worn at any time. It can also be designed on watches, clothing, and other daily necessities. It not only can be used for amputees who have difficulty using traditional HCI equipment (such as a mouse and keyboard) to help them control prosthetic limbs or rehabilitation equipment (Fig. 1-2), but also can be used for people without amputation to enhance the user experience or provide some extended functions for traditional computer equipment [2].

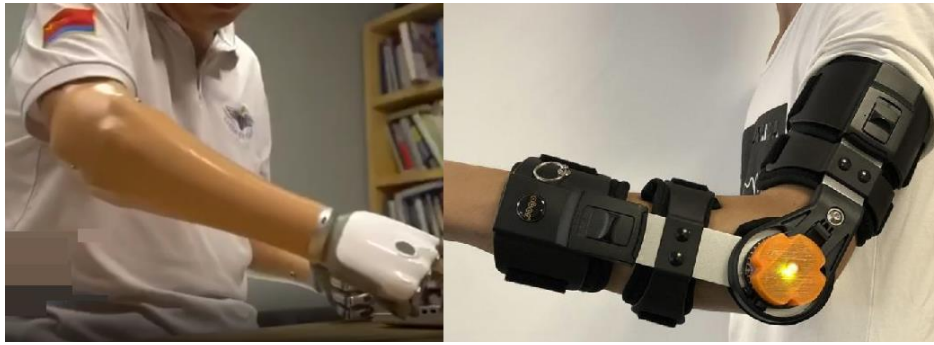


Fig. 1-2 Prosthetics and Rehabilitation Equipment with Multi-Sensor Gesture Recognition

1.2 Pattern Recognition

Pattern recognition is considered to be an effective way to detect a small range of gesture movements through sensors and extract a large amount of information from them [3]. In the aspect of realizing HCI through hand movements, based on the selection of different sensors such as cameras, electrodes on the skin and accelerometers, there are currently two main working directions: gesture recognition and pose estimation.

1.2.1 Pose Estimation

Pose estimation generally uses the camera to capture the user's gestures, and then uses computer vision for pattern recognition (Fig. 1-3). This method is considered to be difficult. The main obstacle is that the limbs block each other when shooting [4]. The user's hand position may also fail to recognize due to the influence of light, background, and other factors [5], thus affecting the HCI effect. At present, pose estimation has made extensive progress in a variety of recognition methods based on appearance, motion, and skeleton, and the recognition accuracy of many methods can reach more than 90% [5].

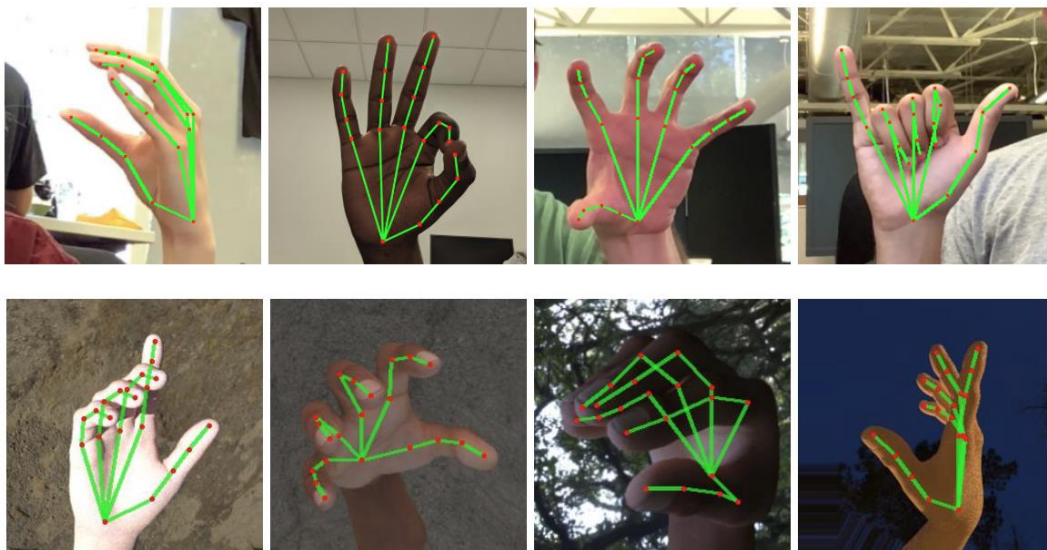


Fig. 1-3 Hand Pose Estimation Using Computer Vision

1.2.2 Gesture Recognition

Gesture recognition generally uses multiple sets of electrodes close to the skin to collect the user's surface electromyography (sEMG), and then uses machine learning methods for pattern recognition (Fig. 1-4). Some studies also add accelerometers to collect signals at the same time, and use data from both the accelerometer and sEMG to enhance recognition accuracy [4]. The main obstacle of this method is that the muscle and fat distribution of different subjects may vary greatly, so it is usually necessary to train the model separately for each subject. In addition, the sEMG signals of multiple muscles of the user interfere with each other, which may affect the recognition effect [4]. At present, the methods based on support vector machine (SVM), random forest (RF), linear discriminant analysis (LDA), and convolution neural network (CNN) have made extensive progress and can achieve more than 90% recognition accuracy [4] [6].

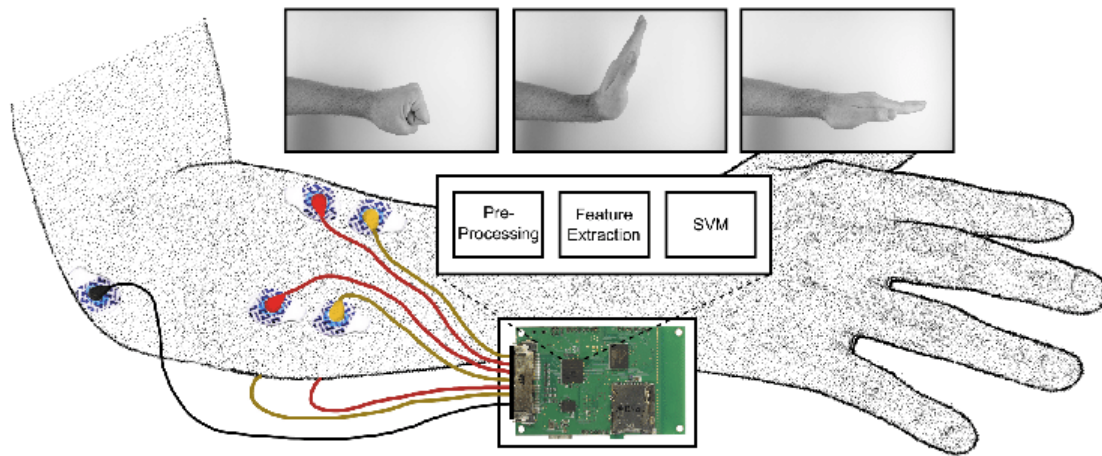


Fig. 1-4 Gesture Recognition Using SEMG Signal

1.3 Description of Experiment

1.3.1 Scenarios and Requirements

Medical surgery robot is an intelligent surgical tool, which not only reduces the risk of infection, shortens the hospitalization time, and reduces the discomfort caused by surgery for patients, but also enhances the accuracy, flexibility, and visualization of surgery for surgeons [7] (Fig. 1-5). The development of medical surgery robots in China has been in a period of rapid growth, and the market scale is also increasing year by year [8]. The doctor needs to adjust the field of vision displayed by the medical image at any time during the operation, so as to locate the operation site. However, in the process of manipulation, it is not convenient to control the medical image at the same time because of the limited range of hand movement. To solve this problem, HCI graphic control technology can identify the additional simple actions made by doctors in the process of manipulation, so as to assist doctors to manipulate the medical surgery robot and adjust the medical image at the same time, which may provide a solution.



Fig. 1-5 Da Vinci Surgery Robot

We interviewed a doctor from The First Affiliated Hospital of the Naval Military Medical University (Shanghai Changhai Hospital), who had many years of experience in using the Da Vinci surgery robot, and made field measurements of the surgery robot. He hopes to control the medical image directly through simple hand movements similar to the manipulation of the surgery robot. However, the control position of the medical imaging device is far away from the surgery robot, resulting in inconvenience during the operation. He also hopes to minimize or avoid the modification of surgery robots. At present, there is no existing product on the market that can fully meet this demand. Our on-the-spot investigation found that the operating space of the surgery robot was relatively narrow, and the space had already contained two joysticks, multiple buttons, and pedals (Fig. 1-6). The measurement shows that there is only a horizontal distance of about 20 cm between the doctor's manipulation position and the robot's main frame (Fig. 1-7).



Fig. 1-6 Manipulation Position of Da Vinci Surgery Robot



Fig. 1-7 Measurement Results

At this distance, if the camera is installed in the manner of pose estimation, only at most two cameras can be installed (one at the front and one at the right), and the process of installing the camera probably requires minor modifications to the surgery robot. However, if the method of gesture recognition is adopted, HCI can be carried out with the help of the myoelectric armband transmitted by wireless signals. Therefore, gesture recognition is our preferred solution.

1.3.2 Workflow

Based on the existing research, we designed five feasible HCI gesture schemes, then collected the sEMG signals of the upper arm of the subjects using a variety of sensors carried by gForcePro+ EMG armband, and constructed a data set based on five different gesture schemes. We then designed a multi-view CNN model. After integrating the model recognition accuracy and various factors that affect the user experience, we selected the most appropriate gesture scheme from five schemes, and finally implemented an HCI graphic control software (Fig. 1-8). The software will facilitate surgeons to adjust medical images while manipulating medical surgery robots, and can also be widely used in other scenarios.

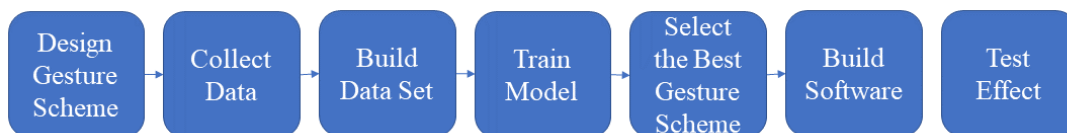


Fig. 1-8 Flow Chart of Experiment

1.3.3 Difficulties

Firstly, the difficulty of the experiment is designing and selecting a gesture scheme with strong practicability, high accuracy, and user-friendliness. So, we had carefully discussed this before designing the gesture scheme, and then trained and screened the designed gesture scheme based on the existing research. Secondly, the difficulty lies in the integration of functions implemented under



different programming languages and frameworks. We investigated and tried many solutions and finally realized the most appropriate one based on user needs. In addition, the experiment involves a large number of parameters that need to be adjusted. If each is adjusted and tested, it requires a huge workload. For this, we first investigate the existing research, determine the optimal range of some parameters in advance, then fix some parameters to control the variables, and only adjust the other parameters that are most likely to affect the result.

1.3.4 Final Effect

In our final work, with the data collected and the model trained, the user can control the 3D reconstruction medical image to translate, rotate, scale, and reset on the remote device in real time by making 12 different gestures through the wireless EMG armband on his upper arm. The software can draw the received upper arm sEMG signal images in real time (Fig. 1-9). The EMG armband is easy to wear and can transmit Bluetooth signals. The user only needs to insert a Bluetooth receiving device at the USB interface of the computer device and run the software to control it. After testing, the remote wireless operation can be conducted within a distance of about 12 meters, with an identification accuracy of about 90% and a delay of about 300 milliseconds.

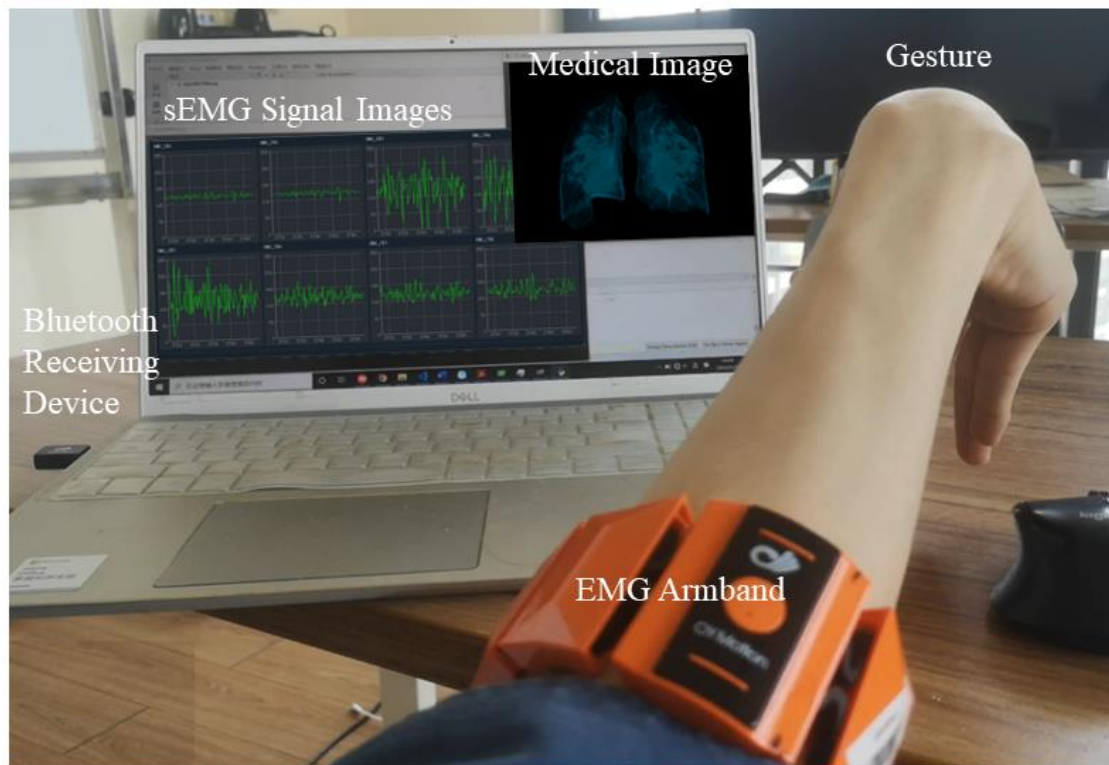


Fig. 1-9 Software Use Effect

Chapter 2 Materials and Methods

2.1 Construction of EMG Data Set

We used the gForcePro+ EMG armband produced by OYMotion Technologies to collect the sEMG signals from the subjects. The device consists of 8 highly sensitive sEMG sensors with differential dry electrodes and a 9-axis inertial measurement unit (IMU) motion sensor distributed in an axisymmetric manner, and can communicate via Bluetooth BLE 4.2 (Fig. 2-1) (see [9] for relevant parameters). The 8 differential dry electrodes of the device are connected by elastic bands so that they can cling to the upper arm muscle surface to collect sEMG signals. These electrodes do not need to be coated with conductive gel, so it is convenient for data acquisition and use.



Fig. 2-1 Internal Structure of gForcePro+ EMG Armband

Some studies show that the pattern recognition method using machine learning is not sensitive to the specific position of the electrodes, and only needs to ensure that the electrodes are in the same position when the subject collects data and uses the armband [10]. Therefore, we did not make strict restrictions on the position of subjects wearing EMG armbands. We only asked the subjects to wear



the EMG armband at the obvious position of the upper arm near the elbow when collecting data (Fig. 2-2), and keep the position unchanged during the collection process. Then we used an oil pen to mark the subject's arm and took photos of the wearing position to ensure the same position when the subject used the armband later.



Fig. 2-2 Wearing Position of EMG Armband

In the process of data collection, the subjects wore armbands and sat in front of the notebook computer. The computer played a video containing prompt information and gesture-related images. The video was set to give prompts according to certain time nodes. The subjects' arm was naturally stretched forward and placed on a platform with appropriate height (Fig. 2-3). We set up a small camera nearby to capture the whole process of data acquisition and used the video information to segment the collected EMG signals to build a data set. The subjects should agree to participate in the experiment voluntarily before participating in the experiment.

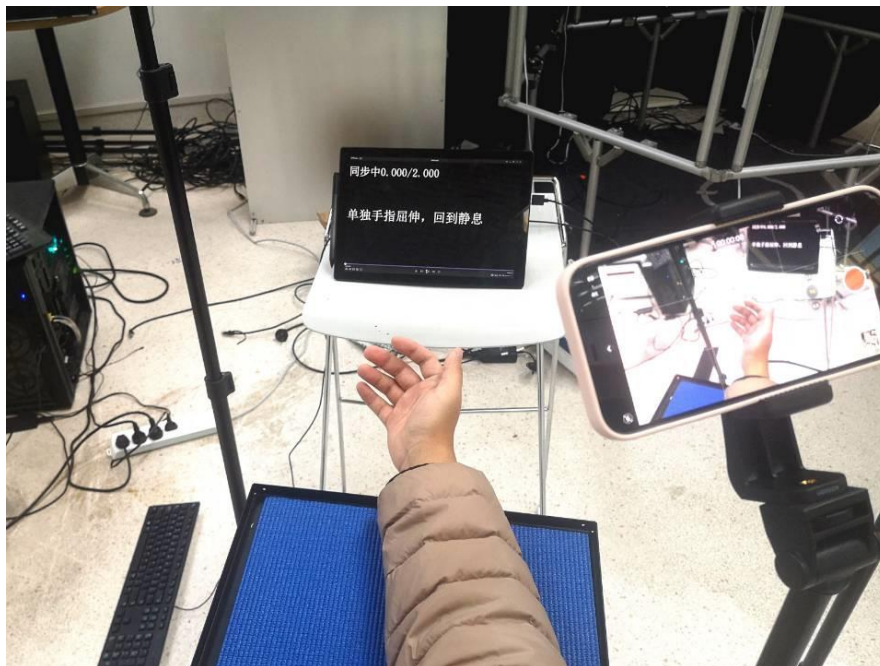


Fig. 2-3 Data Collection Scenario



We designed five different gesture schemes based on some commonly used gestures in gesture recognition (Fig. 2-4) [11] and 11 functions required by the software, as shown in Table 2-1. Among them, gesture scheme 1 required subjects to make dynamic gestures, and gesture schemes 2 to 5 required subjects to make static gestures and maintain muscle strength. Then we collected the forearm sEMG signal data of the same subject wearing the armband at the same position, let him make gestures according to each gesture scheme, and finally constructed five data sets. The relevant parameters used to collect data for each gesture scheme are shown in Table 2-2. In order to prevent subjects' fatigue during repetitive gesture actions and thus affect the data quality, we asked the subjects to rest for 2 seconds between repetitive gestures and 15 seconds between gesture groups. The prompt information was automatically shown to the subject through a pre-edited video.

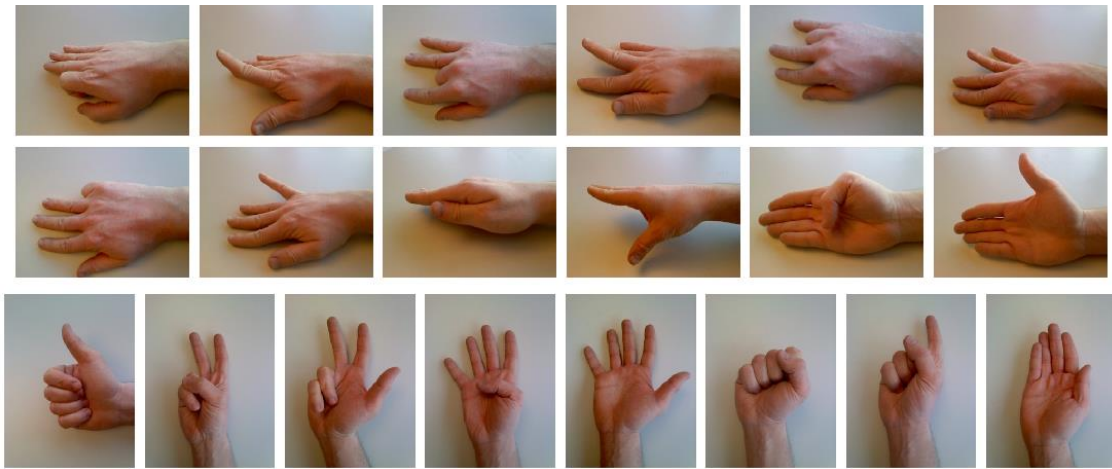


Fig. 2-4 Basic Gestures



Table 2-1 Five Gesture Schemes and Corresponding Functions

Number	Function	Scheme 1	Scheme 2	Scheme 3	Scheme 4	Scheme 5
1	Enlarge					
2	Narrow					
3	Translate Left					
4	Translate Right					
5	Translate Up					
6	Translate Down					
7	Rotate Left					
8	Rotate Right					
9	Rotate Up					
10	Rotate Down					
11	Reset					
12	Static State					

Table 2-2 Data Acquisition Parameters

Name	Value
Number of Subjects	1
Number of Gestures	15
Repetitions	9
Total Data Numbers	135
Gesture Duration	3s
Sampling Rate	500 Hz
Data Precision	12-bit
Number of Nodes	8

We preprocessed the data (Fig. 2-5). The sEMG signal data were first filtered by the 20-500 Hz hardware bandpass filter included in the gForcePro+ EMG armband to eliminate high-frequency random noise interference [9]. The EMG armband stores the collected raw data in a file in binary format. We exported it and represented each data point in integer form. Then we drew the sEMG signal data into a line graph to generate a dynamic video, and used Adobe Premiere to align the EMG signal video with the experiment video recorded by the camera. We then segmented the sEMG data according to the alignment information, discarded the data of the subjects at rest, and labeled the signals of different gestures. We used a sliding window with a length of 200 data points (0.4 seconds), and then standardize each data within the range of $[-1, 1]$.

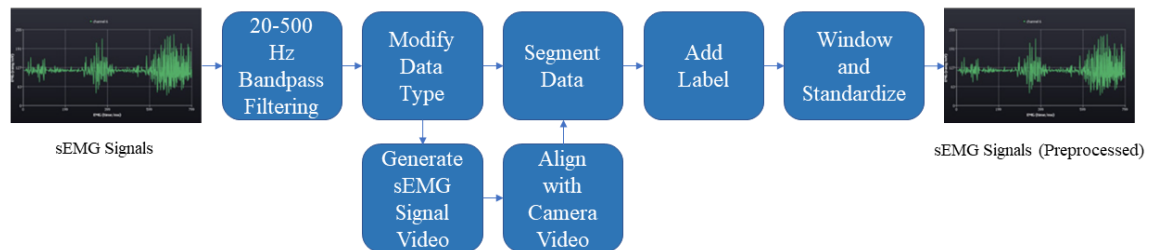


Fig. 2-5 Data Preprocessing Flow Chart

2.2 Model Training

Relevant research shows that many traditional machine-learning methods can achieve more than 90% recognition accuracy in gesture recognition, and deep-learning methods can also achieve similar or even better results [4]. Wentao et al. proposed a deep-learning method based on multi-view feature extraction and achieved better results than traditional end-to-end neural networks in multiple gesture recognition tasks [12]. Based on a similar principle, we designed a CNN with the structure shown in Fig. 2-6 and 2-7. The model used four convolution cores of different sizes for convolution to extract four features of different scales from the sEMG signals, and then transferred these features into a simple network composed of a full connection layer and rectified linear unit (ReLU) for classification. The whole process can be seen as data extraction and classification based on four different perspectives. Multi-view CNN can extract discriminative features from the data, thus improving the classification accuracy of the model, and these features are easily ignored by the traditional single-view deep learning method [12].

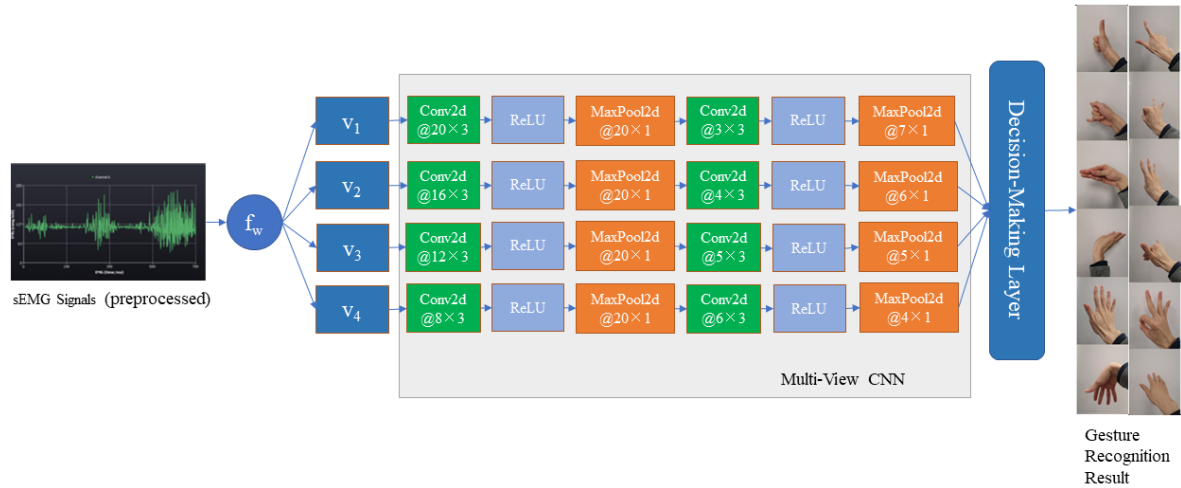


Fig. 2-6 General Structure of Neural Network



Fig. 2-7 Decision-Making Layer Structure

Wentao et al., when comparing the classification accuracy under a different number of views, found that the classification accuracy increased with the increase of view numbers, but after the number of views exceeded 3, the further improvement of the accuracy was very small [12]. After weighing the accuracy and the complexity of the model, we set the number of perspectives to 4.

We trained 200 epochs for each data set and verified the classification accuracy of the model using a 9-fold cross-validation method. We stored the weights of each node when the neural network achieved the optimal recognition accuracy in the verification set during the training process and used them for the gesture recognition part of the subsequent software.

2.3 Software Building

We divided the architecture of the HCI graphic control software based on multi-sensor gesture recognition technology into three parts: real-time data acquisition, real-time gesture recognition part and graphic control part. The three parts were encapsulated into processes respectively, and then we realized mutual Inter-Process Communication (IPC).

2.3.1 Real-Time Data Acquisition Part

The gForcePro+ EMG armband produced by OYMotion Technologies provides an open-source software oym8CHWave, which is developed based on Qt5.15.2 and MSVC2019-64bit, and realizes a real-time data transmission function between the EMG armband device and computer through Bluetooth BLE 4.2 [9]. The sEMG signal data collected by the armband in real-time is transmitted to the computer through Bluetooth and stored in a binary format file. The transmission delay is estimated at 100 milliseconds.

We modified the open-source software oym8CHWave to change the original batch file storage mode



into real-time segmented storage mode. The software was set to collect 256 data points for data storage once to facilitate real-time data processing.

2.3.2 Gesture Recognition Part

We built the part for gesture recognition based on the trained CNN model and Python 3.9, Pytorch 1.10 and CUDA 11.3. This part reads the binary data file in real-time and uses the data preprocessing method that is completely consistent with the data set construction. Then we transmitted the preprocessed data into the CNN to predict the gesture results. Every 256 data points were processed into 56 groups of sEMG signal data through a window with a length of 200. Each group of data generated a prediction gesture result, and 56 prediction results were obtained finally. The software then selected the mode of 56 results, converted the gesture signal into keyboard signals, and transmitted them to the next graphic control process. We used the mode selection judgment to reduce the prediction error and obtained a smooth manipulation effect, thus enhancing the human-computer interaction experience.

2.3.3 Graphic Control Part

Based on Python 3.9, we developed the graphics control part using the algorithm library of OpenGL and PyQtGraph. This part can control the 3D reconstructed medical image and perform a real-time operation on the size and position of the image. Based on the way to control the image, we abstracted 12 specific graphic control functions, and bound each function with the key at the appropriate position on the keyboard (Table 2-3), so that it can receive the signal sent by pressing the key to perform corresponding operation on the medical image.

Table 2-3 12 Graphic Control Functions and Corresponding Keys

Number	Function	Key
1	Enlarge	Q
2	Narrow	E
3	Translate Left	J
4	Translate Right	L
5	Translate Up	I
6	Translate Down	K
7	Rotate Left	A
8	Rotate Right	D
9	Rotate Up	W
10	Rotate Down	S
11	Reset	O
12	Static State	Z

2.3.4 Inter-Process Communication

Because the three parts of this software involve different programming languages and frameworks and there are many complex data structures in each part that need to be initialized, it is difficult and inefficient to integrate them directly into one software. Therefore, we encapsulated the three parts as processes and then met the final requirements through mutual communication between processes.



Common IPC schemes include pipeline communication, local socket communication, file sharing, message queue, etc. We applied communication in the form of file sharing between the processes of the real-time data acquisition part and the gesture recognition part, and between the gesture recognition part and the graphic control part, we applied to simulate and receive keyboard signals, as shown in Fig. 2-8.

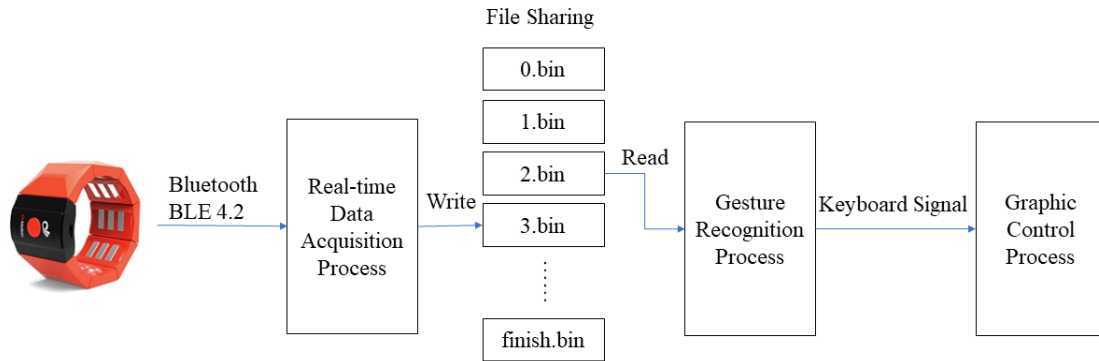


Fig. 2-8 IPC Method

Since the open-source software for data collection, oym8CHWave, stores the collected data in the form of binary files, we believed that the most direct way to use this mechanism is file sharing between the first two processes. Therefore, file sharing was used for communication between the data collection and gesture recognition processes. We set that the graphic control process can read only after the data collection process starts writing the next data file (that is after the next data file exists). Otherwise, the graphic control process will wait to avoid data conflicts.

Since we hope that the HCI graphic control software can be widely used in a variety of computer applications, and currently many of them support receiving the signals from the keyboard, we set the gesture recognition part to send keyboard signals, and set the graphic control part to receive keyboard signals. If the software needs to be applied to cooperate with other traditional computer software, it is only necessary to replace the graphic control part, and then modify the keyboard signal sent by the gesture recognition part.



Chapter 3 Result

We trained 200 epochs for five data sets, and verified the classification accuracy of the model using a 9-fold cross-validation method. This method uses 1/9 of each data set to build a verification set, and 8/9 to build a training set. During the training process, the recognition accuracy of the CNN model on the training set and verification set of five data sets are shown in Fig. 3-1. It can be seen that the number of 200 epochs training is enough to make the model converge.

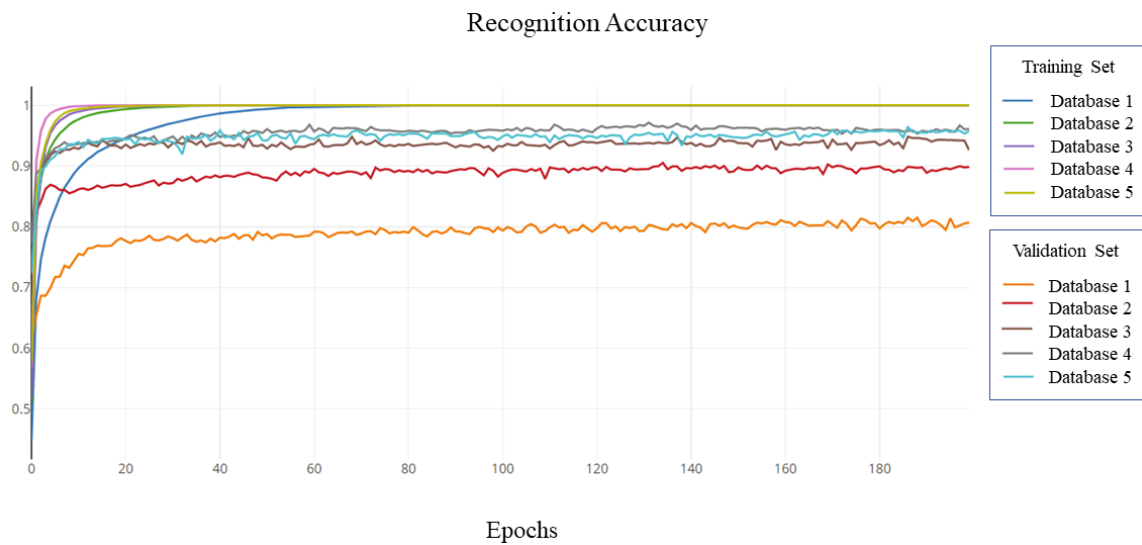


Fig. 3-1 Changes in Recognition Accuracy during Training

In the training process, the optimal recognition accuracy of the CNN model obtained on the verification set of five datasets is shown in Table 3-1. The recognition accuracy of this model on dataset 1 composed of dynamic gestures is significantly lower, and the convergence speed is also significantly slower, so we first exclude this gesture scheme. In the other four data sets composed of static gestures, the recognition accuracy is above 90%, among which datasets 3, 4, and 5 almost achieve the highest accuracy.

Table 3-1 Recognition Accuracy on Verification Set

Dataset	Accuracy
1	0.8150
2	0.9051
3	0.9483
4	0.9713
5	0.9638

We finally chose dataset 3 as our final gesture scheme for HCI software, because it is the most accurate and user-friendly. In terms of the control effect of the software, we hope that the slightly



different similar functions (such as translating the image in four directions) can be controlled by slightly different similar gestures. We also hope that the direction of the user's gestures can also be consistent with the direction of the medical image movement.

We asked the user to wear the EMG armband, and gradually stay away from the computer equipment under Bluetooth connection until the user had difficulty in manipulation, and measured the distance between the computer equipment and the user at this time. After testing, the user can conduct remote control within a distance of about 12 meters. We took videos of manipulation when the user repeated the normal gestures many times and count the recognition accuracy, as well as measured the delay between the user's two gestures and the software's response. After testing, the recognition accuracy was about 90%, and the delay was about 300 milliseconds.

The HCI software we finally implemented basically achieved the relevant functions required for graphic control. Although there was still a delay of 300 milliseconds, it would not significantly affect the user experience. Although we had developed various ways to make the software manipulation smoother to reduce error, sometimes it would misjudge the gesture made by the users, thus making a wrong operation on the 3D reconstructed medical image. In this case, the users can usually quickly make feedback and adjustment, so that the manipulation can return to the normal state. We consider that this accidental judgment error is acceptable.



Chapter 4 Discussion

4.1 Discussion on Experiment

4.1.1 Workflow

The first purpose of this study is to design a gesture scheme with good recognition accuracy and as user-friendly as possible for HCI graphic control. The second purpose is to realize the HCI graphic control software, and make its architecture versatile in common application scenarios. After specific experiments and analysis, we suggest that as many gesture schemes as possible be designed first, then build data sets for each scheme and train models, and finally select the best gesture scheme. To improve the user experience, we suggest using the mode of data to smooth the operation, and adjusting the window size and the amplitude of image movement after each signal reception.

4.1.2 Gesture Scheme

In our experiment, we found that because the gesture is recognized by sEMG signals rather than by camera, some gestures with different appearances but similar muscles may not be well distinguished. We suggest trying to avoid introducing too similar gestures when designing gesture schemes, otherwise it is likely to reduce the recognition accuracy of the model. In addition, in the process of collecting data, we found that some subjects were very embarrassed when making some difficult gestures, which made it difficult to make standard gestures (Fig. 4-1). Therefore, we should try to avoid introducing such difficult gestures in the HCI gesture scheme.

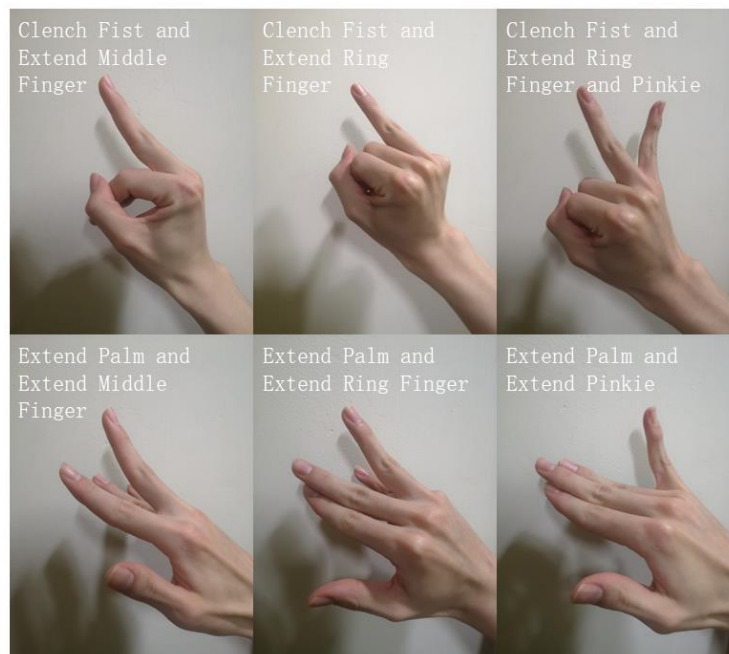


Fig. 4-1 Some Difficult Gestures



4.1.3 Acquisition

Since the constructed sEMG dataset directly affects the weights of each node of the CNN, and then affects the recognition accuracy of the model, thus affecting the final software effect, the quality of the data is very important when initially constructed. We suggest that the user should properly increase the rest interval between repetitive gestures during data collection, and avoid continuously collecting the gestures from the same subject for a long time to prevent hand muscle fatigue. We suggest that the subjects should be familiar with the video and related gestures before collection to avoid major mistakes. If the subject makes continuous mistakes in the data collection process, this group of data should be re-acquired immediately. Angkoon et al. suggested that during data collection, the data of the same gesture made by the subjects with different forces should be included in the dataset as far as possible to improve the final recognition accuracy [6]. So, we also suggest that when collecting data, the subjects should not repeat the same action with the same force frequently.

4.1.4 Sliding Window Size

The size of sliding window during data windowing is also an important factor affecting the recognition accuracy and software effect. Increasing the window size may improve the recognition accuracy, but it will also increase the delay caused by data processing. The final window size should be determined by weighing these two factors. Research shows that the size of the sliding window should be more than 200 data points and should be as large as possible [13], but the delay of data processing should not be more than 300 milliseconds [14]. Arjan et al. proposed a new measurement standard of Movement Error Rate (MER) to balance the two factors of recognition accuracy and delay, and believed that part of the recognition error was caused by delay [15]. However, in this experiment, we did not include the error caused by the delay in recognition accuracy, so we did not use the MER. Since the delay caused by Bluetooth transmission has reached the level of 100 milliseconds, we choose to set the window size to 200 data points to minimize the delay. Because the acquisition equipment used in different experiments is different, the environment generated is also different, and many different schemes can be used when building the software. Therefore, we suggest that the window size should be adjusted based on the actual situation so that users can have the best interaction experience.

4.2 Improvement Directions

4.2.1 Adding Accelerometer Data

Some studies have shown that adding accelerometer data can significantly improve the model recognition accuracy and reduce the prediction error [4] [15] compared with the prediction using only sEMG signals. After comparing and analyzing the signals before and after adding accelerometers, Robin et al. concluded that the addition of accelerometers can help the model reduce the false positive prediction error, and believed that this improvement effect can be widely applied to amputees and non-amputees [4]. The gForcePro+ EMG armband provides a 9-axis IMU motion sensor. However, since adding more signals may make the data processing process and CNN model more complex, we only collect the sEMG data of users for HCI graphic control. In the experiment



of Robin et al., the addition of accelerometer data increased the model recognition accuracy by about 0.5-1.5% in different data sets [4]; In the experiment of Arjan et al., the addition of accelerometer data significantly reduced the movement error rate of the model with the same delay [15]. Therefore, we believe that adding accelerometer data is a reasonable improvement direction to improve the recognition effect without increasing the delay.

4.2.2 Increasing Sampling Rate

Angkoon et al. conducted in-depth research on the impact of different sampling rates on the accuracy of model recognition when collecting sEMG signals, and found that almost all the information important for model classification is concentrated in the band of high-frequency signals [6]. If the sampling rate is reduced from 1000Hz to 200Hz, the recognition accuracy of the model will be significantly reduced [6]. Wilson et al. even found that if the high-pass filter is used to filter all the low-frequency sEMG signals of 20Hz-120Hz, it can even improve the model recognition accuracy [16]. We believe that although raising the sampling rate can improve the recognition accuracy, it also increases the data processing delay by increasing the total number of data points per unit of time. In addition, in the 12-bit data mode, the maximum sampling rate of gForcePro+ EMG armband can only be set up to 500Hz, and in the 8-bit data mode, the maximum sampling rate can be set up to 1000Hz. Although the accuracy of each data point in the 8-bit data mode will not be as high as that in the 12-bit mode, we still believe that using the 1000Hz and 8-bit data mode achieves a possibly high HCI effect. It is necessary to carry out further experiments to prove this.

4.2.3 Optimizing Inter-Process Communication

File reading and writing by the operating system will cause milliseconds of delay. The specific delay size will vary according to the file size and system architecture. Because we use the sharing file method to communicate between the data acquisition process and the gesture recognition process, continuous file reading and writing may cause significant delays. If the pipeline communication or other IPC provided by the operating system is used, it is possible to reduce the delay and improve the HCI effect. In addition, if all parts of the software are rewritten in the same programming language, it is easy to package them uniformly and interact with each other, but the cost of doing so is high. Further testing of other IPC methods may make progress in reducing latency.

4.2.4 Combining with Attitude Estimation

Many pose estimation methods using camera and computer vision technology can achieve more than 90% recognition accuracy [5]. Although it may be difficult to recognize limb occlusion when shooting with a single camera, this can be avoided by erecting multiple cameras with different angles. A maximum of two cameras can be installed in the scene of this experiment. We speculate that compared with the recognition of the sEMG signal of the upper arm, gesture estimation using computer vision can better recognize the different postures of the hand, and some gestures with different postures but similar muscles may also be better recognized. Therefore, it is necessary to try the attitude estimation scheme and compare the effect with the experimental results. We can even try to combine the two pattern recognition methods in subsequent experiments.



Chapter 5 Conclusion

We designed five different gesture schemes, then collected the sEMG signals of the upper arm of the subjects using various sensors carried by gForcePro+ EMG armband, and constructed a data set based on five different gesture schemes. We then designed a multi-angle CNN model. After integrating the recognition accuracy of the model and various factors affecting the user experience, we selected the most appropriate gesture scheme, and finally implemented an HCI graphic control software using the gesture scheme. The user can conduct remote wireless manipulation through gestures within a distance of about 12 meters. The recognition accuracy is about 90%, and the delay is about 300 milliseconds. This software will facilitate surgeons to adjust medical images while manipulating medical surgical robots, and can also be widely used in other application scenarios.



References

- [1] TURK M, ROBERTSON G G. Perceptual user interfaces (introduction) [J]. ACM, 2000, 43(3):32–34.
- [2] PHINYOMARK A, QUAIN F, LAURILLAU Y. The Relationship between Anthropometric Variables and Features of Electromyography Signal for Human-Computer Interface [M]// GANESH R. NAIK. Applications, Challenges, and Advancements in Electromyography Signal Processing. Pennsylvania: IGI Global, 2014:321–353.
- [3] OSKOEI M A, HU H. Myoelectric control systems—A survey [J]. Biomedical Signal Processing & Control, 2007, 2(4):275–294.
- [4] AYLING R, JOHNSON C G, LI L, et al. Pose Estimation from Electromyographical Data using Convolutional Neural Networks [J]. IEEE, 2000, 653-656.
- [5] OUDAH M, AL-NAJI A, CHAHL J. Hand Gesture Recognition Based on Computer Vision: A Review of Techniques [J]. Journal of Imaging, 2020, 6(8):73.
- [6] ANGKOON P, RAMI N K, ERIK S. Feature Extraction and Selection for Myoelectric Control Based on Wearable EMG Sensors [J]. Sensors, 2018, 18(5):1615.
- [7] UC Health. Benefits of Robotic Surgery [EB/OL]. (2022-05-29)[2022-05-29]. <https://www.uchealth.com/services/robotic-surgery/patient-information/benefits/>
- [8] 前瞻产业研究院. 预见 2022:《2022 年中国手术机器人行业全景图谱》(附市场规模、竞争格局和发展前景等) [EB/OL]. (2022-04-12)[2022-05-29]. <https://finance.sina.com.cn/roll/2022-04-12/doc-imcwipii3798216.shtml>
- [9] OYMotion Technologies Co., Ltd. gForce EMG Armband User Guide [EB/OL]. (2021-08-21)[2022-05-29]. <https://oymotion.github.io/assets/downloads/gForce-EMG-ARMBAND-User-Guide-202108.pdf>
- [10] HARGROVE L, ENGLEHART K, HUDGINS B. A training strategy to reduce classification degradation due to electrode displacements in pattern recognition based myoelectric control [J]. Biomedical Signal Processing & Control, 2008, 3(2):175–180.
- [11] ATZORI M, GIJSBERTS A, HEYNEN S, et al. Building the Ninapro database: A resource for the biorobotics community [C] // 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob). IEEE, 2012: 1258–1265.
- [12] WEI W, DAI Q, WONG Y, et al. Surface-Electromyography-Based Gesture Recognition by Multi-View Deep Learning [J]. IEEE Transactions on Biomedical Engineering, 2019, 66(10): 2964-2973.
- [13] YENTES J M, HUNT N, SCHMID K K, et al. The appropriate use of approximate entropy and sample entropy with short data sets [J]. Annals of biomedical engineering, 2013, 41(2):349–365.
- [14] ASGHARIOSKOEI M, HU H. Myoelectric control systems—A survey [J]. Biomedical Signal Processing and Control, 2007, 2(4):275–294.
- [15] GIJSBERTS A, ATZORI M, CASTELLINI C, et al. Movement Error Rate for Evaluation of Machine Learning Methods for sEMG-Based Hand Movement Classification [J]. IEEE



- Transactions on Neural Systems & Rehabilitation Engineering, 2014, 22(4):735-44.
- [16] WILSON A W, LOSIER Y G, PARKER P A, et al. A bus-based smart myoelectric electrode amplifier-system requirements [J]. IEEE Transactions on Instrumentation & Measurement, 2011, 60(10):3290 - 3299



Acknowledgments

I would like to thank Professor Zhang Yuyao of the School of Information Science and Technology (SIST) of ShanghaiTech University, for her careful, patient, and hard guidance on the research project and Bachelor's Thesis. I would like to thank Jiang Yingwenqi, Yao Yichen and Yang Jiawen of the SIST of ShanghaiTech University for their guidance and help. Thank OYMotion Technologies company for providing gForcePro+ EMG armband equipment and related open-source software, oym8CHWave, for this project. Thank Mr. Ni Hualiang, CEO of OYMotion Technologies, for his many online Q&A on the EMG armband technology during his busy schedule, and for providing strong support for the project. I would like to thank Professor Ren Shancheng of Shanghai Changhai Hospital for providing a valuable opportunity to visit the Da Vinci medical surgery robot and to watch him operate the surgery robot to complete tumor resection.

This research project coincided with the outbreak of COVID-19 in Shanghai. Therefore, the successful completion of this undergraduate research project requires special thanks to all the teachers, staff, student volunteers, medical personnel, and other staff who participated in the logistics support work of ShanghaiTech University during the epidemic. Most of the time when I finished my experiment and writing, I was isolated and confined in the dormitory building. If there were no canteen staff to make meals, volunteers to distribute materials, cleaning staff to keep the environment clean, and classmates to help each other in the most difficult times, it would be difficult for me and other project participants to maintain a normal life, and this scientific research project might not be completed. I would like to extend my sincere thanks to you.

Thanks to ShanghaiTech University for providing perfect infrastructure and rich scientific research materials, which played a vital role in completing this research project in the special period.

Finally, I would like to thank my classmates for their harmony with me for the past four years, and my parents for their support and help in completing my undergraduate study.