Recognition and Control of a Chinese Pinyin Sign Language Robot via a Cognitive Robotics Approach

Hanzhang Cui[1], Yang Luo[2][0000-0002-7478-1159], and Fan Zhang[1][0000-0001-6228-940X]

[1] School of Robotics, Xi'an Jiaotong-Liverpool University, Suzhou 215213, China
[2] School of Intelligent Manufacturing Ecosystem, Xi'an Jiaotong-Liverpool University, Suzhou 215213, China
fan.zhang@xjtlu.edu.cn

Abstract. Technological advancements in AI and Robotics are facilitating the development of sign language robots, which are resolving communication barriers for people living with disabilities. The current work aims to develop a sign language robot that mimics how human communicate in Chinese Sign Language. Two primary developments are presented in this paper, namely the real-time translation of sign language and the movement production of a simulated dexterous hand. Real-time translation was performed using computer vision and deep learning techniques, and the movement production of a dexterous hand to represent the sign language gestures implemented forward kinematics. The robot vision system can detect and tracking human hand, allowing real-time highly accurate recognition of hand positions and shapes. Employing the VGG16 architecture and customised training dataset, it was further developed to identify sign language gestures, achieving an accuracy rate of 80% between ambiguities. The simulation of a bionic dexterous hand was realised in MATLAB Simulink. The simulated bionic dexterous hand can provide real-time feedback on displacement data and adjusting input signals for enhanced control. The two developments of the current work could be easily integrated and prototyped as robotic hand which could behave on par with human-like communications via sign languages beyond Chinese Pinyin. It could also lead to the development of better human-robot interaction systems for people living with hard-of-hearing disabilities.

Keywords: Chinese Pinyin Sign Language, Robot Vision, Dexterous Hand, Cognitive Robotics

1 Introduction

In the current digital era, ongoing technological advancements are opening new avenues to tackle various societal challenges. Nevertheless, the communication barriers encountered by the disabled community, especially the hard-of-hearing disabled people, remain a significant concern. Often, hard-of-hearing disabled people face communication challenges with people who can hear, which can result in obstacles in their education, employment, and social interactions [1]. The scarcity of effective communication tools and technologies frequently restricts sign language communication for the hearing impaired. Traditional sign language interpreting services, typically costly and inconvenient, pose difficulties for this group in accessing information and engaging in social activities [2]. Consequently, there is a pressing need for more efficient sign language interpreting services [3]. The development of an intelligent, user-friendly, and practical sign language robot system is crucial to close this communication gap. The exploration of sign language robots is aimed at enhancing the daily lives of the hard-of-hearing disabled people. By offering straightforward and practical sign language interpreting tools, it becomes possible to forge more opportunities for this unique community, facilitating their better integration into society. Thus, the deployment of sign language robots represents a pioneering effort to mitigate these issues. The sign language robot not only recognizes and understands sign language, but also expresses sign language through bionic hand movements, thus greatly facilitating effective communication between hard-of-hearing disabled people and normal people. It recognizes sign language for interpretation and responds with sign language through bionic hand movement. This is a promising research area for sign-based recognition, learning and interpretation [2].

Chinese Sign Language (CSL) is mainly expressed through gestures and movements as well as facial expressions, and contains a Pinyin-like alphabetic spelling system. During the recognition process, key information is encapsulated in human gestures, specifically, robots need to recognize the orientation and shape of the hand [4]. Since 1987, when scientists began using glove sensors to determine hand position and orientation, sign language recognition has transitioned to today's use of deep learning methods. Deep learning models are constructed using Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) and extract features from video and image data [5]. Many similar sign language interaction robots customize interaction directly through big data [6]. Others implement human-robot interaction by inputting their respective training sets and training models [7-9]. Two popular forms of translation are machine translation and speech translation [10]. The former uses text as output and the latter uses speech as output. In this sign language robot, the camera translates the gestures into text by recognizing them. On the other hand, the robot outputs sign language by mechanically simulating hand movements. In this way, the robot realizes the human-robot interaction function of sign language.

Unlike industry robotics that chases cost-saving and production efficiency, in Cognitive Robotics we design robots that mimic human behaviours. This is of vital importance in Human-robot Interaction (HRI) research, especially for sign language robots. Cognitive Robotics presents a novel approach in understanding of human behaviour, by building computational models to capture the human perception and actions (e.g. [11,12]), even mimic the error pattern typically exhibited in human decision-making and relevant behaviours (e.g. [13]).

Moreover, sign language databases, computer vision, convolutional neural networks, and real-time sign language translation are all essential in sign language recognition and translation projects. Calinon and Billard have noted that the Hidden Markov Model (HMM) is a widely utilized method for gesture recognition by researchers, yet it demands considerable time for recognizing sign language [14]. Thus, the exploration of an alternative method for sign language recognition is essential. Expressed primarily through gestures and movements, along with facial expressions, Chinese Sign Language (CSL) also incorporates an alphabetic spelling system akin to Pinyin. In the process of recognition, the critical information is encapsulated in human gestures, specifically, the direction and shape of the hand need to be recognized by the robot [4]. Since 1987, when scientists began using glove sensors to determine the position and orientation of the hand, there has been a transition to today's use of deep learning methods for sign language recognition. Currently, the two prevalent forms of translation are machine translation and speech translation [6]. The former is text as output and the latter is speech as output. In this sign language robot, the camera translates it into text by recognizing gestures. On the other hand, the robot outputs the sign language through mechanical simulation of hand movements. In this way the robot can realize the human-robot interaction function of sign language.

Sign language is characterized by its use of visual-gestural rather than auditory-phonetic patterns [15]. The primary users and target audience are individuals who are hard-of-hearing or have speech impairments. While not widespread among the general population, sign language is a common mode of communication among the hard-of-hearing disabled people. Sign languages are not universally the same; rather, like spoken languages, they have evolved into various regional-specific sign languages. With the proliferation of these sign languages, there are now hundreds commonly employed by the hard-of-hearing disabled people community. The formation of signs in sign language is accomplished through the coordinated use of various body parts, including fingers, palms, and arms [16].

The gesture depicted in Fig. 1 represents finger spelling. Chinese Sign Language is categorized into three types: finger spelling, isolated words, and continuous sign sentences [19], with this model focusing specifically on finger spelling. A comparison between Chinese Sign Language (CSL) and American Sign Language (ASL) reveals significant differences between the two. Apart from the letters C and O, the rest are essentially distinct. The focus of this project is directed towards the recognition of Chinese one-handed sign languages, encompassing Chinese letters and various signs bearing Chinese meanings.

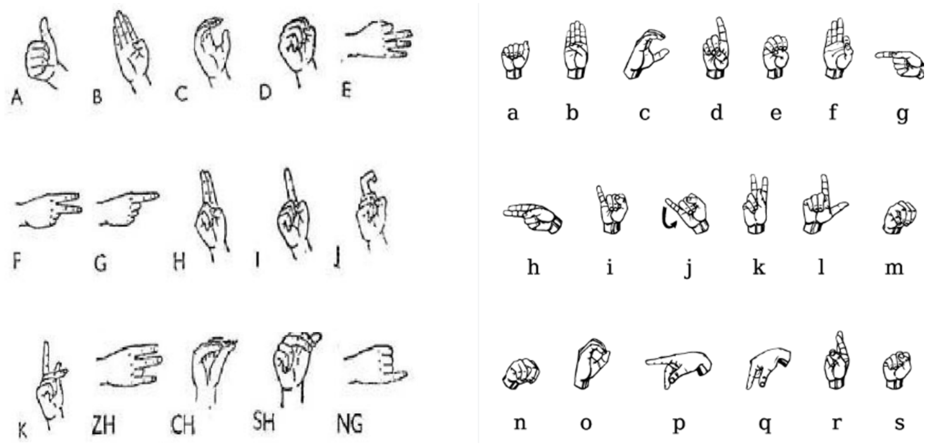

Fig. 1. *A comparison between Chinese Alphabet (Pinyin) and American Alphabet. Left: Chinese Pinyin sign language, adapted from [17]. Right: American sign language, adapted from [18]*

The aim of the current work is to develop a sign language robotic system that allows the hard-of-hearing disabled people to engage in natural and fluent communication with the robot via sign language. The paper is structured into two main sections. First, we present the use of Python vision and deep learning for real-time sign language translation. The process of real-time sign language recognition is unveiled. Initially, an overview of the technologies to be utilized, such as CNN and OpenCV, is provided as the backdrop. Subsequently, we present the use of MATLAB to simulate a robotic arm for sign language output, in which the outcomes of the real-time sign language recognition are presented and discussed comprehensively. Specifically, we set forth the following objectives and presented the work as the following:

① Accomplish gesture tracking, capturing, and storage. The positions and shapes of gestures are to be recognized using computer vision techniques [14]. Following recognition, the images are to be stored.

② Construction of a training model for learning gesture features and classification. This model will be utilized to identify various features of hand gestures, enhancing the computational efficiency and accuracy of the network [15].

③ Execute real-time sign language detection and textual translation. A camera, enabled by computer vision, is deployed to detect gestures in real time [19]. The model's accuracy in recognizing sign language is then evaluated and calculated.

④ Conduct MATLAB simulations of the robotic hand. The robotic hand simulates human finger movements to produce the corresponding sign language gestures.

2 Robot Vision – Sign Language Recognition

This section describes the entire sign language recognition process and modelling methods, specifically the use of the VGG-16 training model [20]. It customised videos as training dataset. Comparative experiments were conducted to test the accuracy and usefulness of the system.

2.1 Methods

There were three main steps in Sign Language Recognition. First, a gesture data acquisition script was created using Python code. The gestures in the camera images were captured using the OpenCV library and the cvzone library [21]. They saved as image files. The cvzone library used in this case is based on a high-level wrapper for the 'Media Pipe Hands' library. ' Media Pipe Hands' has a model based on Graph Convolutional Network that detects 21 key points of the hand. These key points included the wrist, the joints of all fingers, etc., resulting in a complete structural map of the hand. Such recognition was more inclined towards feature recognition and is more accurate. As shown in Fig. 2, the whole hand was labelled by node links. To capture the hand in the image, cvzone is utilized to directly detect the hand and draw the boundaries and points. The hand bounding box information was obtained, and the image was captured and saved to the folder 'datatrain/cC'. These images were categorized into different folders as a training set, which is the basis for the next step of training the model.

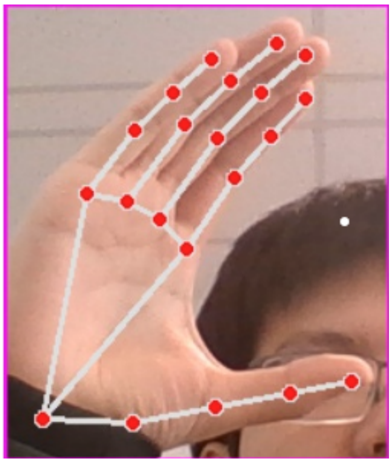

Fig. 2. *An example of image collection - This gesture means C*

Secondly, through the steps of preprocessing images, loading data, constructing models and training models, the deep learning model that can be used for sign language recognition is finally obtained. Defining a dictionary of label mappings for the gesture dataset was the first step, i.e. creating 'gesture' dictionary maps the filename prefix of the gesture to the name of the gesture. Then, 'gesture_map' was utilized to map the gesture names to the digit labels used for model training. Next, 'gesture_names' was defined to do the reflective mapping of 'gesture_map' for easy understanding and visualization. The image file will then be resized to 224×224 pixels and converted to NumPy array [22]. For better training of the neural network, the data needs to be narrowed down from 0-255 to 0-1 pixels and labelled with one-hot coding for classification. The images required for training are saved in the 'datatrain' folder. Each file was checked with the 'os.walk' function. The first two characters of the file name was recognised, and 'gesture' was received from the corresponding gesture. Lastly, the image was passed to the 'process data' function for processing. It utilizes a mapping system to find the corresponding sign language meaning (Table 1).

Table 1. *Gesture-Letter Mapping System Gesture-Letter Mapping System - Identifiers such as "aA" map letters to unique integer indexes via "mapping". The "name" is a reflection of the "mapping".*

Gesture	Map	Name
aA	0	A
bB	1	B
cC	2	C
dD	5	D
eE	26	E

After processing, the dataset was divided into training and test sets in a ratio of 4:1. A checkpoint was set during training for saving the best model. An early stop object for stopping training early was then created when the model performance no longer improved. To retain the best model, the VGG16 model pre-trains the weights and adds a fully connected layer to build the top-level structure of the model [23]. In terms of local connectivity, the VGG16 model network reaches a depth of 16-19 layers, showed good performance. The VGG16 network structure contains parameters with 13 convolutional layers, 5 pooling layers, and 3 fully connected layers, excluding the activation layers. It starts with 64 convolutional kernels, and as the network deepens, the number of convolutional kernels increases all the way up to 512, making it a powerful network. Comparisons were made between the training and test sets, and early stopping methods and model checkpoint callback functions were used to ensure accuracy.

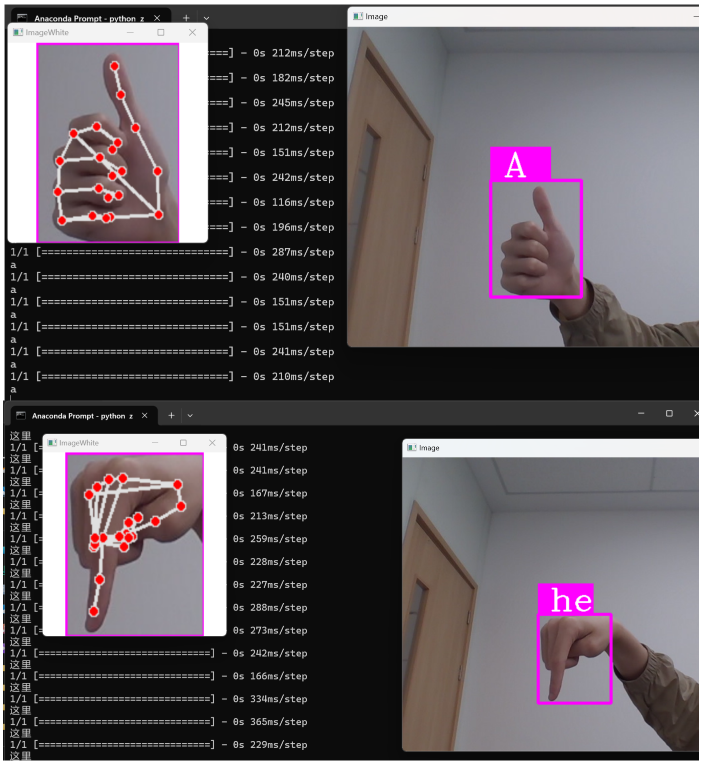

Fig. 3. *Screenshot of the testing the CV running in real-time. Top: Recognizing Sign Language – 'A' stands for the gesture whose Chinese meaning is 'A'. Bottom: Recognizing 'he' stands for the sign meaning 'here' in Chinese. In these testings, gestures recognised were marked in the pictures showed on the right on top of the hand that was framed. One the left the feature points were marked and superimposed on a close-up of the hand.*

Finally, real-time sign language translation (Fig.3). Instant text translation while recognizing gestures. It was specifically designed to perform the gesture recognition task in a separate thread. The design improved the responsiveness and efficiency of the application. Then, a dictionary was created to record the mapping of the corresponding Chinese characters or letters. Next, in the "run" function, the mapping can be automatically called, and the task executed. Finally, after recognizing the hand in real time, its image was captured, and the replica was scaled and predicted for classification. The scaling value depended mainly on the ratio of the predefined image size to the height of the hand region. With this ratio, it can be scaled to a predefined size. When the size was the same, the corresponding data was captured based on the index of the prediction result and its corresponding content was output.

To explore the hand position and gesture recognition accuracy of the sign language interpreting system, two item comparison experiments were conducted. The main feature recognition used here. To test the accuracy of feature recognition, fingers of the hand such as the index finger were exposed, and the palm of the hand was covered. Observe whether it recognized the subject of the shot. Secondly, two similar sign language detections were used to test whether the system eclipses can recognize gestures with a high degree of similarity. The accuracy of the system was determined through three sets of experiments.

2.2 Results

The basic recognition features of the python code were tested. The recognition principle is to use the information of 21 nodes to recognize the position and shape of the hand. Overall, the system was found to be able to recognize sign language properly and output it as text.

Two testing were conducted. The first test was to cover palm of the hand and expose the fingers to test whether the system could observe it with occlusion. As shown in Appendix (Fig. 12), when the palm is covered and only two fingers are extended, the system can detect the shape of the entire hand. However, the system was unable to recognize when there was only one finger in the shot. The camera can only recognize the palm if the whole palm is in the lens.

Table 2 shows the results and evaluations using ‘c’ as an example. After ordinary recognition, multiple testing was conducted on colour recognition and feature recognition. The accuracy of the system was found to be 93.3%. There were errors when it appears in darker coloured environments.

Table 2. Number of accuracies in recognising sign language ‘c’

Number of identifications	Meaning	True/False
1	c	T
2	c	T
3	c	T
4	c	T
5	c	T
6	c	T
7	c	T
8	c	T
9	x	F
10	c	T

The second experiment was to test sign language in two similar poses. This was used to test whether the system could accurately recognize the gestures with ambiguities. In the training model, four training sets of images with different gestures, including the Pinyin 'aA', 'bB' , 'eE' and the word ‘he’ (representing “here”), were fed into the training model. Examples of the recognition is shown in Fig. 4. Note that the images stored were not necessarily the corresponding images in Chinese sign language. The images of “eE” and that of “he” were very similar in gesture shapes. The differences are that the index finger and thumb are positioned differently, and the other fingers are curled when the gesture with the index finger pointing downwards means “here”, while gesturing with an interval between the index finger and the thumb means “eE”.

Fig. 4. *Four images of gestures - The recognition from left to right are the Pinyin ‘A’, ‘B’, ‘E’, and the word ‘here’.*

Focusing on distinguishing 'eE' and 'he' gestures during startup program recognition, the test methods included multi-angle recognition, fuzzy recognition, and simulation recognition. Fuzzy recognition is used to test the program recognition error by showing the sign language with non-standard postures. Please see detailed procedures below.

Firstly, gestures were shown at three angles. The robot will recognize them and output the corresponding meaning of the sign language. Through this experiment, the accuracy of this visual recognition system in recognizing sign language from multiple angles is evaluated. From there, its feasibility of recognizing sign language in practical applications was explicitly explored. See in Fig. 5 for the results of the test.

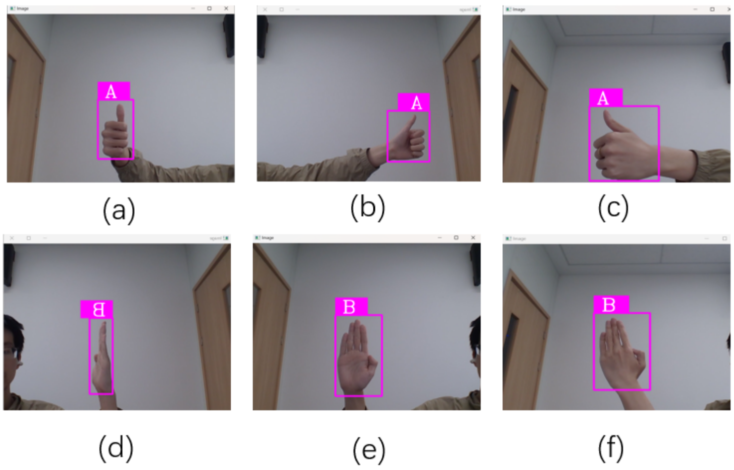

Fig. 5. *Recognition from different angles - The three figures (a), (b), (c) represent the gesture meaning ‘A’ in Chinese recognised from three angles. The three diagrams (d), (e), (f) represent the gesture meaning ‘B’ in Chinese recognised from three angles.*

It can be concluded that the multi-angle recognized gestures were essentially error-free and their accuracy was about 95%. Analysing from a methodological point of view, the recognition system mainly relies on the spatial distribution of hand gestures for recognition, which was based on the relative coordinates of 21 key nodes. By comparing the relative positions of these nodes, the system searches for patterns with high similarity to the known sign language, thus realizing the recognition of the sign language. This approach based on the spatial distribution of nodes was reasonable and showed good results in experiments.

The second comparison was to observe the accuracy of the recognition system in recognizing partial gestures. This method reduces the number of recognition nodes to misrecognize the sign language. Although the system was unable to accurately recognize the sign language through the nodes, this method could be used to observe if the system is able to ambiguously recognize the meaning of the sign language as well as the recognition error. From Appendix Fig. 13, the recognition system can be seen. With 2/3 of the gestures exposed, the nodes were around 17 and the recognition error is low. 80% of the sign language will be recognized accurately. Among them, gestures that were obvious and special such as sign language meaning 'B' were more easily recognized. When 1/3 of the gestures are revealed, the recognition error was higher. As shown in Appendix Fig. 14, sign 'A' is recognized as 'B' and sign 'C' was recognized as 'E'. The fewer the nodes, the larger the recognition error. In this case, the accuracy was basically 20-30%. To summarize, the system was more accurate when recognizing revealing 2/3 gestures and above, i.e. it was better to show the full sign language in front of the camera.

The third comparison aims at recognizing similar sign languages and evaluating the accuracy of the recognition system. With this experiment, the system would be tested to see if it can accurately distinguish between the two differences and to determine how well it performs in recognizing gesture-like language. This comparison focused on the recognition of two similar sign languages, “E” and “he”, because these similar sign languages are more likely to trigger recognition errors and thus could provide insight into why the errors occur. Results are shown in Table 3.

Table 3. Table of Recognition Results for Similar Sign Languages

Sign Language Meaning of Show	Meaning of recognized	Detecting the gesture (T/F)
he	he	T
he	he	T
he	E	T
E	E	F
E	E	F

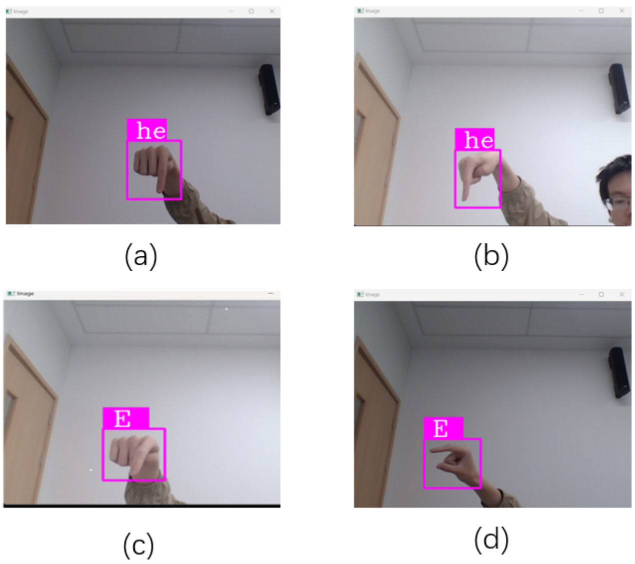

Fig. 6. *Detection results of similar sign language - Pictures (a), (b), (c) are all hand gestures meaning 'here' in Chinese. However, the gesture 'here' in (c) is incorrectly recognised as 'E'. The gesture in (d) is correctly recognised as 'E'.*

As shown in Fig. 6 when the sign language "he" was featuring an obliquely extended thumb is present, the system ends up recognizing "E" instead of "he." This observation suggests that in practice, the system may be affected by video fluency when processing dynamic gestures, which in turn affects the accuracy of sign language recognition. At rest, the system's recognition accuracy is at 90%. The accuracy is higher when the number of training sets is increased. The training sets in this project are around 18, as shown in Fig. 7. We created new videos of sign language recording as the training dataset for the current work. The number of training images for an action was between 10 and 20 (as in frames). Accuracy would increase as the number of training images increases.

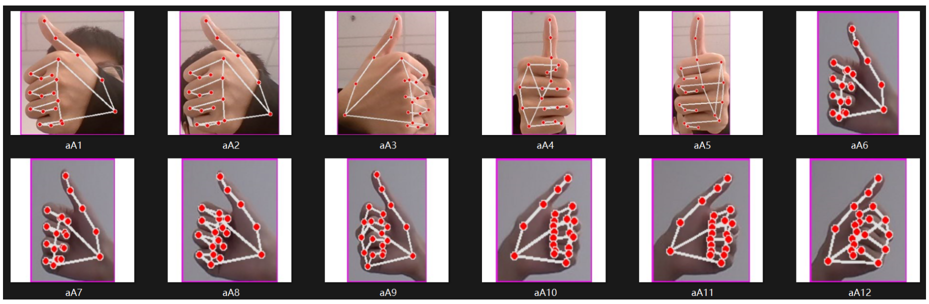

Fig. 7. *Training set for sign language 'A'*

3 Robot Control – dexterous hand movement production

In this section, we demonstrate and analyse the results and accuracy of the model and movement production. First, we present the work in simulating the bionic dexterous hand through the Simulink module and outputting sign language by implementing predefined inputs [24]. Next, modelling principles and observed gesture movements are analysed and discussed.

3.1 Methods

In MATLAB, the basic bionic hand needs to be modelled first (Fig.8). Control the motion of this model using forward kinematics and import it in various modules. This will allow us to show the trajectory of the bionic hand as it makes various gestures after running the program.

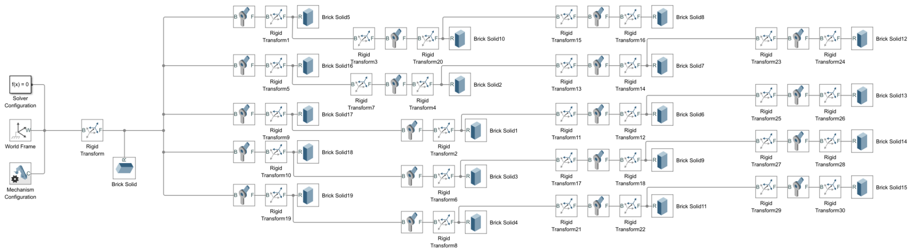

Fig. 8. *Bionic dexterous hand modelling. Each joint represents a finger joint which can be rotated in a plane. The 'solid' module is representing the connection between the finger joints of the hand. One degree of freedom was added to the joint between the finger and the palm. As a result, the thumb has three 'solid' modules the other has 4 'solid' modules.*

The following steps were proceeded to load the bionic dexterous hand in Simulink. The bionic hand data was first loaded into the workspace via the 'import robot' script, and then loaded into the workspace via the 'import robot' script. Next, the data was loaded into the 'Get Transform' module of Simulink. Through the function block, the required coordinate system transformation parameters can be obtained and then used to realize the corresponding transformation operations. The 'Coordinate Transformation Conversion' function block was then used to perform the actual coordinate system transformation.

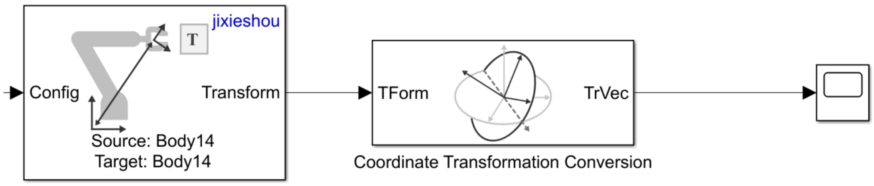

Fig. 9. *Motion control - Coordinate system parameter transformations are performed through the 'Get transform' and 'Coordinate Conversion' modules. With this you can get information such as the angular rotation coordinates of the joints.*

The movement production was realised by using forward kinematics (Fig.9). It first describes the position and attitude of an object in 3D space using homogeneous transform. Given a 3D coordinate system at the centre of each object, the motion of the object is represented by a change in the coordinate system. This method can represent the rotation and translation of an object in 3D space. Then the robot joint motions such as bionic hand were observed and parameterized by Denavit - Hartenberg (DH) table and the transfer matrix. Finally, the final coordinates can be derived.

3.2 Results

In addition to the importance of visual recognition, bionic hand simulation is a key step in the research. By simulating a hand model, the ability to acquire data related to hand movements and simulate the process of sign language interaction by a robot. Simple bionic hand models are constructed using joints and rectangles, and the displacement data of the joints or rectangles can be derived to observe their motion characteristics. In this research phase, a simulation experiment will be conducted using the sign language letter "A" as an example. This will provide basic data and theoretical support for us to further understand the motion characteristics of hand gestures and the simulation of robotic sign language interaction.

The program shown in Fig. 10 was developed based on a model created by forward kinematics and was used to simulate the motion of the hand model. In this subsystem, the basic structure diagram of the bionic hand was presented by "joint" and "solid" components. An external sinusoidal function fed the motion signals to the bionic hand, and its motion data was then passed through the modules 'Get Transform' and 'Coordinate Transformation Conversion'. These two modules were usually used in conjunction with other modules to ensure that coordinate transformations and conversions were handled correctly in the Simulink model. After running, the motion process can be viewed through the 'Mechanics explorers' window and the displacement images can be viewed in the 'scope' window to get a deeper understanding of the motion characteristics of the bionic hand and the simulation results.

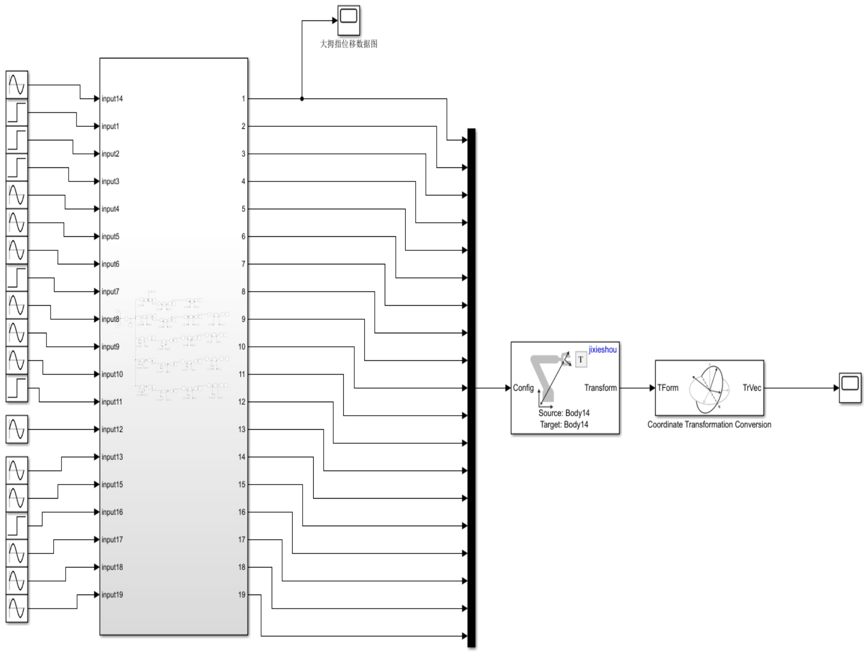

Fig. 10. *Structure of the gesture 'A' - The image shows the structure of a bionic hand. The input signals are passed into the robot and then the output signals are passed to the ‘Get transform’ and ‘Coordinate conversion’ module*

In Appendix Fig. 15, we illustrate the basic bionic hand model that was built, while in Appendix Fig. 16, we present the bionic hand model performing the gesture 'A' movement. In the movement of the bionic hand model, the movement is relatively smooth due to the use of sinusoidal function as the input signal. During the execution of gesture 'A', the thumb moves downward while the other four fingers show a bent state. The entire gesture takes 2 seconds to complete. This type of motion with sinusoidal function input is common in simulation and its smoothness makes the bionic hand movement natural and smooth.

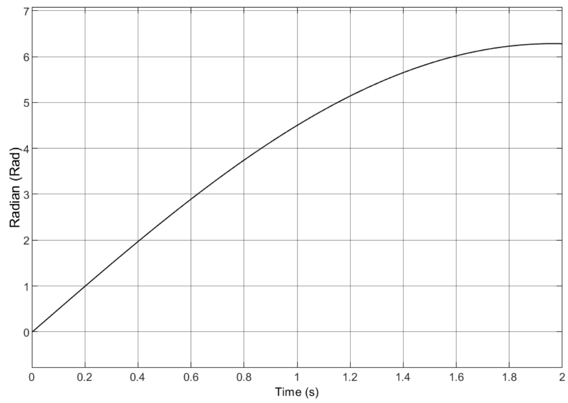

Fig. 11. *Thumb movement data - Angular rotation data*

Fig. 11 shows the finger displacement data extracted from the bionic hand model, which demonstrates that the motion data of the bionic hand model can be directly applied to practical application scenarios. By extracting the motion data from the bionic hand model, it can be used in real HCI systems, thus providing an important foundation and support for the development of HCI technology. This simulation result verifies the effectiveness of the bionic hand model in simulating human hand movements and provides practical guidance for applying the bionic hand technology to real-world scenarios. In Table 4 we show the final pose of each joint for the four gestures 'A', 'B', 'C', and 'here '. A robot hand could be simply controlled for corresponding sign language by controlling different arc lengths or angles of the nineteen joints. These results could be directly applied to future robots prototyping.

Table 4. The final pose of each joint for the four gestures 'A', 'B', 'C', and 'here '. Ninteen joint's rotation arc length for different gestures in simulation are recorded in radians.

	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19
A	0.5	0	0	0	1.8	2.0	1.5	0	1.8	2.0	1.5	0	1.8	2.0	1.5	0	1.8	2.0	1.5
B	-0.2	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	1	-0.5	0	0.5	1	0.5	0	0.5	1	0.5	0	0.5	1	0.5	0	0.5	1	0.5
he	0	0	0	0	1.8	0	0	0	1.8	2.0	1.5	0	1.8	2.0	1.5	0	1.8	2.0	1.5

4 Discussion

The project commenced with the ambitious objective of designing and implementing a sophisticated sign language robotic system to improve communication for those with hearing impairments. This conclusion encapsulates the principal findings and insights derived from the research and experiments conducted throughout the project.

An extensive literature review and background study on key components crucial for the project—sign language, computer vision, and convolutional neural networks (CNN)—was carried out in the initial phase [25]. This foundational research furnished a detailed comprehension of the complexities and unique characteristics of sign language, setting it apart from spoken languages with its visual-gestural modality. The exploration of computer vision techniques and CNNs enhanced our understanding of the technological dimensions essential for the success of the project.

The real-time sign language recognition system developed in the first section of the project has yielded promising results. The system effectively used colour recognition and feature recognition to recognize gestures. The use of OpenCV library and cvzone library proved to be helpful in capturing and processing gestures accurately [26]. The implementation of a deep learning model for sign language recognition pre-trained using the VGG16 model improved the accuracy and efficiency of the system. However, the experiments also revealed some limitations. Although the system recognizes gestures with 95% accuracy, it faces challenges in recognizing sign languages with similar gestures and fuzzy recognition. These findings indicate areas of improvement for the system in future iterations, especially in enhancing the fuzzy recognition algorithm and improving the ability to recognize similar gestures.

The second section of the project involves MATLAB Simulink to build a bionic hand, which has great potential. It will not only complement the real-time translation system but will also pave the way for more interactive and dynamic sign language communication. It simulates motion in real-world environments and derives displacement data for each joint of the bionic hand. The simulated data can be directly rationalized for the real environment and the bionic hand can be created quickly.

In conclusion, the project marks a significant step forward in bridging the communication gap for the hard-of-hearing disabled people [27]. While there are areas for improvement, the research and development conducted have laid a solid foundation for future advancements in this field [28]. The project underscores the importance of technology in enhancing inclusivity and accessibility, paving the way for a more integrated society where communication barriers are progressively diminished.

Acknowledgement

This project is supported by XJTLU Internal funding PGRS/TW5A2312003 to FZ.

Open practices statement

Materials such as Chinese sign language recognition, MATLAB simulation of the corresponding software and some simulation results, could be found available in the OSF repository (https://osf.io/j2s9p/?view_only=4783ae77224248a397c3c9cbe6a0ab25) upon publication. None of the experiment reported in the present study was preregistered.

References

[1] Rastgoo R, Kiani K, Escalera S. Sign language recognition: A deep survey[J]. Expert Systems with Applications, 2021, 164: 113794.

[2] Cheok M J, Omar Z, Jaward M H. A review of hand gesture and sign language recognition techniques[J]. International Journal of Machine Learning and Cybernetics, 2019, 10: 131-153.

[3] Nandy A, Mondal S, Prasad J S, et al. Recognizing & interpreting Indian sign language gesture for human robot interaction[C]//2010 international conference on computer and communication technology (ICCCT). IEEE, 2010: 712-717.

[4] Calinon S, Billard A. Stochastic gesture production and recognition model for a humanoid robot[C]//2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566). IEEE, 2004, 3: 2769-2774.

[5] Nihal R A, Broti N M, Deowan S A, et al. Design and development of a humanoid robot for sign language interpretation[J]. SN Computer Science, 2021, 2: 1-17.

[6] Lim J Y, Sa I, MacDonald B, et al. A Sign Language Recognition System with Pepper, Lightweight-Transformer, and LLM[J]. arXiv preprint arXiv:2309.16898, 2023.

[7] Li J, Zhong J, Wang N. A multimodal human-robot sign language interaction framework applied in social robots[J]. Frontiers in Neuroscience, 2023, 17: 1168888.

[8] Zhi D, de Oliveira T E A, da Fonseca V P, et al. Teaching a robot sign language using vision-based hand gesture recognition[C]//2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA). IEEE, 2018: 1-6.

[9] Sabyrov A, Mukushev M, Kimmelman V. Towards Real-time Sign Language Interpreting Robot: Evaluation of Non-manual Components on Recognition Accuracy[C]//CVPR Workshops. 2019.

[10] Núñez-Marcos A, Perez-de-Viñaspre O, Labaka G. A survey on Sign Language machine translation[J]. Expert Systems with Applications, 2023, 213: 118993.

[11] Patil M, Heinke D, Zhang F. Reveal the Distractions of the Irrelevant Features using a Neurobiologically Plausible Cognitive Robotics Model. In Rita 2023-The 11th International Conference on Robot Intelligence Technology and Applications 2023 Oct 31.

[12] Makwana M, Zhang F, Heinke D, Song JH. Continuous action with a neurobiologically inspired computational approach reveals the dynamics of selection history. PLoS Computational Biology. 2023 Jul 17;19(7):e1011283.

[13] Leek EC, Leonardis A, Heinke D. Deep neural networks and image classification in biological vision. Vision Research. 2022 Aug 1;197:108058.

[14] Bradski G, Kaehler A. Learning OpenCV: Computer vision with the OpenCV library[M]. " O'Reilly Media, Inc.", 2008.

[15] Chua L O, Roska T. The CNN paradigm[J]. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 1993, 40(3): 147-156.

[16] Voulodimos A, Doulamis N, Doulamis A, et al. Deep learning for computer vision: A brief review[J]. Computational intelligence and neuroscience, 2018, 2018(1): 7068349.

[17] J cofresi, "汉语手指语图", Pinterest,
URL: https://www.pinterest.co.uk/pin/390194755199902627/ (accessed Dec. 4, 2023)

[18] "American Sign Language Alphabet" [Online image], Wikipedia,
URL:
https://zh.wikipedia.org/zh-cn/%E7%BE%8E%E5%9C%8B%E6%89%8B%E8%AA%9E%E5%AD%97%E6%AF%8D, Dec 18, 2004 (accessed Dec. 4, 2023)

[19] Kamal S M, Chen Y, Li S, et al. Technical approaches to Chinese sign language processing: A review[J]. IEEE Access, 2019, 7: 96926-96935.

[20] Qassim H, Verma A, Feinzimer D. Compressed residual-VGG16 CNN model for big data places image recognition[C]//2018 IEEE 8th annual computing and communication workshop and conference (CCWC). IEEE, 2018: 169-175.

[21] Pulli K, Baksheev A, Kornyakov K, et al. Real-time computer vision with OpenCV[J]. Communications of the ACM, 2012, 55(6): 61-69.

[22] Chua L O. CNN: A vision of complexity[J]. International Journal of Bifurcation and Chaos, 1997, 7(10): 2219-2425.

[23] Tammina S. Transfer learning using vgg-16 with deep convolutional neural network for classifying images[J]. International Journal of Scientific and Research Publications (IJSRP), 2019, 9(10): 143-150.

[24] Sridevi P, Islam T, Debnath U, et al. Sign Language recognition for Speech and Hearing Impaired by Image processing in matlab[C]//2018 IEEE Region 10 Humanitarian Technology Conference (R10-HTC). IEEE, 2018: 1-4.

[25] Ragib Amin Nihal, Nawara Mahmood Broti, Shamim Ahmed Deowan, and S. Rahman, "Design and Development of a Humanoid Robot for Sign Language Interpretation," SN Computer Science, vol. 2, no. 3, Apr. 2021, doi: https://doi.org/10.1007/s42979-021-00627-3.

[26] Zhi D, de Oliveira T E A, da Fonseca V P, et al. Teaching a robot sign language using vision-based hand gesture recognition[C]//2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA). IEEE, 2018: 1-6.

[27] Nihal R A, Broti N M, Deowan S A, et al. Design and development of a humanoid robot for sign language interpretation[J]. SN Computer Science, 2021, 2: 1-17.

[28] Akhtar N, Ragavendran U. Interpretation of intelligence in CNN-pooling processes: a methodological survey[J]. Neural computing and applications, 2020, 32(3): 879-898.

Appendix

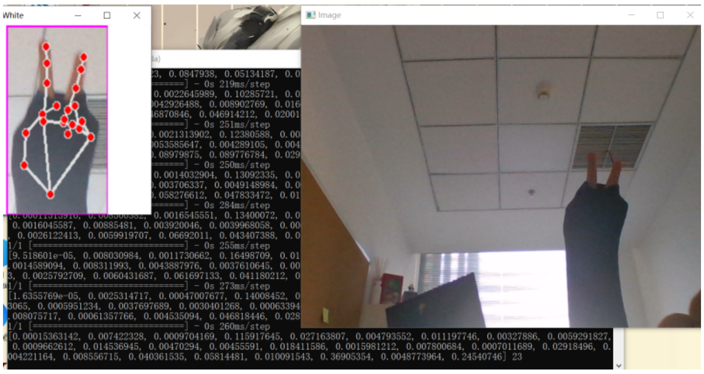

Fig. 12. *Detect when finger is exposed - Observe whether the system recognises the obscured hand.*

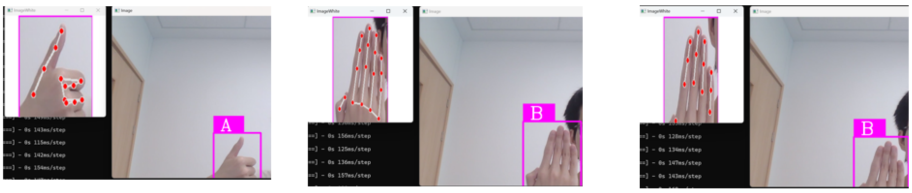

Fig. 13. *True Recognition - These three figure identifications are correct.*

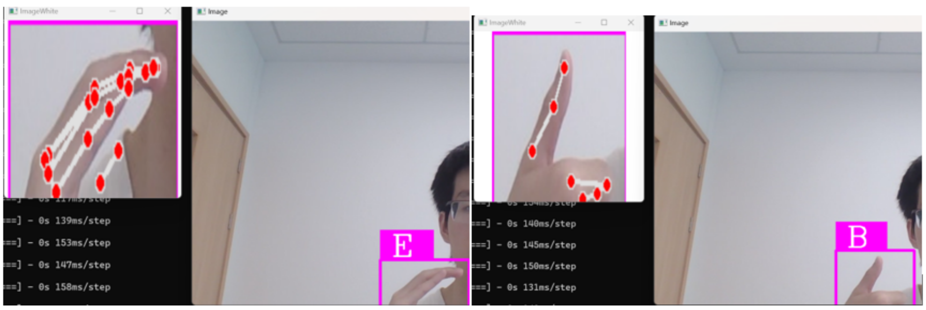

Fig. 14. *False Recognition - When 1/3 of the gesture was exposed, the system incorrectly recognised gesture 'C' as gesture 'E'. Gesture 'A' was incorrectly recognised as gesture 'B'.*

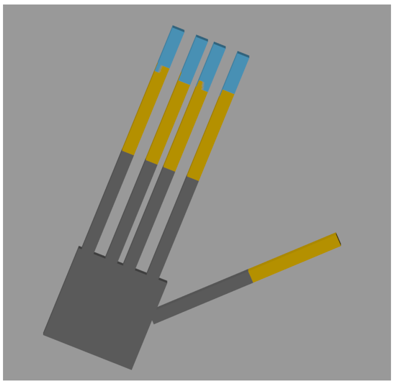

Fig. 15. *Bionic Hand's Basic Model*

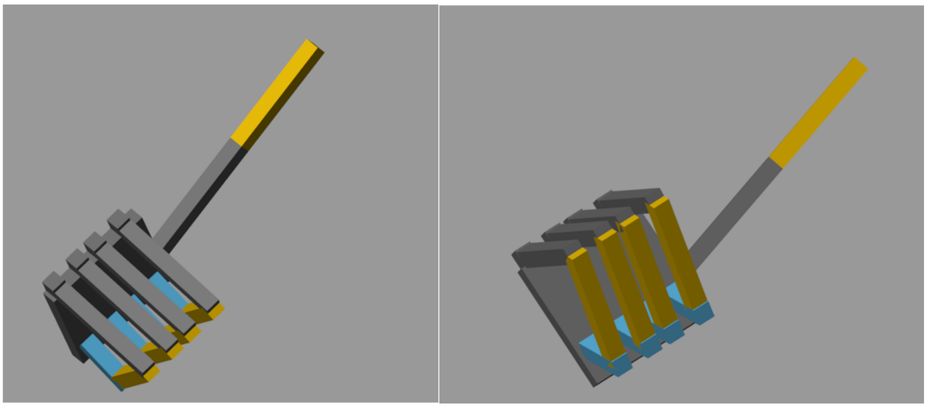

Fig. 16. *Gesture 'A' – The movement result for gesture 'A'*