



Title	Sign Language Translation Using Wearable Motion Capture System and Machine Learning Methods
Author(s)	Gu, Yutong
Citation	北海道大学. 博士(工学) 甲第15177号
Issue Date	2022-09-26
DOI	10.14943/doctoral.k15177
Doc URL	http://hdl.handle.net/2115/87183
Type	theses (doctoral)
File Information	Yutong_Gu.pdf



[Instructions for use](#)

Doctoral Dissertation

Academic Year 2022

Sign Language Translation Using Wearable
Motion Capture System and Machine Learning
Methods



HOKKAIDO
UNIVERSITY

GU Yutong

Division of Human Mechanical Systems and Design,

Graduate School of Engineering

Hokkaido University

Sign Language Translation Using Wearable Motion Capture System and Machine Learning Methods

Summary

Sign language is the main communication method among hearing-impaired people. As a kind of natural language, sign language has not become a mainstream research topic in natural language processing, although the machine translation of spoken or written language is highly accurate today. However, the research of machine translation with deep learning models provides development direction and innovative methods for sign language translation tasks. In order to further the research on end-to-end translation, it is necessary to consider the application of deep learning models. Previous works about sign language translation mainly falls to two categories: vision-based and wearable sensors-based. Vision-based methods exploit camera to capture features of hands. In wearable sensors-based research, devices like data glove, wristwatch or armband are the mainstream for data collection. In this dissertation, we will explore the sign language translation using wearable sensors.

Chapter 1 provides the background and necessity of sign language translation task. The literature review of translation methods and devices were introduced. The normally used datasets were also summarized.

Chapter 2 focuses on the isolated gesture recognition. Some widely use hand gesture recognition datasets were introduced. Then, the classification task of Ninapro DB5 dataset was tested with traditional machine learning models. Also, customized model for Ninapro DB5 was built to promote the

classification accuracy. Finally, dataset of 26 ASL alphabet gestures was collected by hand motion capture system. The quality of data from the device was tested.

Chapter 3 introduces the finger spell in American sign language. We collected inertial data of right hand during ASL letters performance, and did the classification tasks in both letter and word levels. The machine learning model contains convolutional neural network layers for feature extraction, long short-term memory layers for learning time series characteristics, and connectionist temporal classification layers to solve alignment problem between model output and ground truth.

Chapter 4 presents a wearable sensors-based sign language translation method considering both hands movements and facial expressions. Inertial measurement units and electromyography signals were preprocessed and segmented into a sequence of frames as the input of translation models. We classified facial expressions with EMG data only. Then we built encoder-decoder models to realize end-to-end sign language translation from signals to text sentences. Two kinds of end-to-end models based on LSTM and transformer were trained and evaluated by the collected dataset. WER and SER were used to compare the translation ability of models. Both models could translate 40 ASL sentences with high accuracy and the transformer-based model performed better than LSTM. The special role of EMG was verified with both facial expressions' classification and models' performance after removing EMG from the input. The translation accuracy in user-independent conditions was evaluated.

Chapter 5 summarizes the works in the dissertation and offers the prospective studies that can be investigated in this research field.

Finally, this study proposed the complete research process of sign language translation technology. The research was started from isolated gesture recognition and finally went to the end-to-end translation of full

sentences using wearable sensors. The facial expressions in sign language performances were also collected by EMG device. The combination of natural language processing and wearable sensors provides a new idea for sign language translation task. The datasets we collected will make it easier for more people to start research on sign language translation and machine learning. The works in this study are significant new and may contribute a huge impact for researchers in this field.

Acknowledgement

Special thanks to Prof. Todoh, Prof. Yamada, and Prof. Zha for the academic instruction on my research.

Special thanks to Hokkaido University DX Fellowship, f3 project, tuition fee exemption, MEXT honor scholarship, and COVID-19 emergency support for financial support. Special thanks to the biggest sponsor my mother.

The research about machine learning is interesting. I've been learning it for only two years. I would like keep on researching on it in the future, even now, my research is still very elementary. I watch the free online lecture of Pokemon master every year. I'm glad my friends took part in my boring experiments.

I like Hokkaido. I'm here to heal the sickness and stress I've accumulated over the years. Accidentally, I escaped catastrophe of covid here. In Hokkaido, I gained the only confidence in my life. Where my heart is at ease is my hometown.

Recently, I finally got up the courage to watch *Attack on Titan*. The world is cruel, and I have been running away. But now I am not afraid of this world. The first German I learned: wir sind der jäger.

Last year's November, I went to Kyoto to ask gods about what should I do in the future. The gods answered: Do what you really want to do. I was confused about future, but now, I know the answer. The gods here are very friendly.

I've always wanted to go to the beach to watch the sunrise. But I don't think I will do that.

Table of Contents

1. Introduction	1
1.1 Research Background	1
1.2 Related Works	3
1.2.1 Wearable Sensors-based Methods	3
1.2.2 Computer Vision-based Methods	6
1.2.3 Sign Language Dataset	7
1.2.4 Natural Language Processing	9
1.3 Research Purpose	12
2 Isolated Gesture Recognition Using Surface Electromyographic Data and Pattern Recognition Methods	14
2.1 Introduction	14
2.2 Materials and Methods	15
2.2.1 Benchmark Datasets Description	15
2.2.2 Data Preprocessing for Benchmark Dataset	19
2.2.3 Isolated Gestures for ASL Alphabet	21
2.2.4 Perception Neuron Motion Capture System	22
2.2.5 Dataset Collection	24
2.2.6 Data Preprocessing for Collected Dataset.	25
2.2.7 Model Design	27
2.3 Results	29
2.3.1 Gestures Classification Using Benchmark Dataset	29
2.3.2 Self-Collected Dataset of ASL Alphabet	32
2.4 Discussion	35

2.4.1	Features Selection	35
2.4.2	Early Fusion and Late Fusion Models	36
2.4.3	Features Selection Results	37
2.5	Conclusions	42
3	ASL Finger-spell Translation Using Hand Motion Capture System	43
3.1	Introduction	43
3.2	Materials and Methods	43
3.2.1	Words Dataset	43
3.2.2	Algorithm Introduction	44
3.2.3	Sequence to Sequence Model Design	49
3.3	Results	51
3.3.1	Words Level Classification	51
3.3.2	Sequence Recognition Results	52
3.4	Discussion	53
3.5	Conclusions	53
4	Sign Language Translation Using Wearable Inertial and Electromyography Sensors for Tracking Hand Movements and Facial Expressions	55
4.1	Introduction	55
4.2	Materials and Methods	56
4.2.1	ASL Specifics	56
4.2.2	Dataset Collection	57
4.2.3	Data Preprocessing	58
4.2.4	Facial Expressions Classifier	60
4.2.5	Sign Language Translation Models	61
4.3	Results	64
4.3.1	Facial Expressions Classification	64
4.3.2	Sign Language Translation	65

4.4	Discussion	70
4.4.1	Significance of EMG	70
4.4.2	User-independent Validation	72
4.4.3	Limitations	73
4.5	A Larger Dataset	75
4.6	Conclusions	75
5	Conclusions and Future Work	76
	References	79

List of Figures

1.1	CyberGlove data glove.	4
1.2	MYO armband for gesture recognition.	5
1.3	RWTH-PHOENIX-Weather example of original video frame. The sign language interpreter is shown in an overlay on the right of original video frame.	8
2.1	Placement of the electrodes: A. sEMG electrodes placed on finger extensor muscles (A.1 Equally spaced electrodes; A.2 Spare electrode); B. sEMG electrodes placed on finger flexor muscles (B.1 Equally spaced electrodes; B.2 Spare electrode); C. all the sensors positioned on the arm (C.1 Equally spaced electrodes; C.2 Spare electrode; C.3 Inclinometer; C.4 CyberGlove II).	16
2.2	Hand gestures of the Ninapro dataset.	17
2.3	Ninapro DB5 data acquisition setup.	18
2.4	HD-sEMG sensors for Capg-Myo dataset collection.	19
2.5	Sampling the dataset into small sEMG images.	20
2.6	Twenty-six signs of American sign language letters.	21
2.7	Dynamic process of gesture “j”.	22
2.8	Perception Neuron motion capture system.	22
2.9	Main interface of Axis Neuron software.	23
2.10	Fifty-nine bones of human skeleton model.	24
2.11	Right-hand motion capture device.	25
2.12	Useful information selection for right hand joints.	26
2.13	Signal segmentation with sliding window method.	27

2.14	The complete process of costume hand gesture classification system.	28
2.15	A custom machine learning model for hand gesture classification.	28
2.16	Hand gestures classification model with sliding window method for input segmentation.	29
2.17	Training and testing loss of the model.	32
2.18	Training and testing accuracy of the model.	32
2.19	Loss for model training and testing steps.	33
2.20	Accuracy for model training and testing steps.	33
2.21	User independent validation accuracy of 4 participants.	34
2.22	Confusion matrix of participant 3 (best among 4 participants).	34
2.23	Confusion matrix of participant 4 (accuracy 77.9%).. . . .	35
2.24	Selected right hand joints with original joint numbers.	36
2.25	Two kinds of multi-features as input.	37
2.26	Early and late fusion of classification models.	37
2.27	Classification accuracy of four kinds of cases.	38
2.28	Classification results with each selected feature.	39
2.29	Confusion matrix of all cases for participant 1.	40
2.30	Confusion matrix of all cases for participant 2.	40
2.31	Confusion matrix of all cases for participant 3.	41
2.32	Confusion matrix of all cases for participant 4.	41
3.1	Finger spelling of “world”.	44
3.2	How to form a “world”.	45
3.3	The proper path for π^1 (blue), π^2 (red), π^3 (green), π^4 (black).	46
3.4	All possible paths for correct alignments.	48
3.5	Sequence recognition model.	49
3.6	Detailed structure of LSTM unit.	49
3.7	Word level classification model.	51

3.8	Words classification results.	51
3.9	Sequence recognition result without considering individual. . . .	52
3.10	An example of alignment with largest probability.	52
3.11	Ten-fold cross validation of sequence recognition.	53
4.1	Devices for data collection: (A) Perception Neuron motion capture system; (B) EMG signal acquisition system.	57
4.2	Signal preprocessing flowchart.	59
4.3	An example of EMG data preprocessing: (A) Raw EMG data from sentence No. 21; (B) Corresponding preprocessed EMG data. . .	60
4.4	Facial expressions classification model.	61
4.5	Architecture of LSTM-based translation model.	62
4.6	Transformer-based translation model: (A) Architecture of the model; (B) Detailed structure of the self-attention layer.	64
4.7	Facial expressions classification results: (A) Accuracy of five cross-validation sets; (B) Total confusion matrix of cross-validation steps.	65
4.8	Training and validation losses of LSTM model for Participant one.	66
4.9	LSTM model evaluation result for Participant 1.	67
4.10	Training and validation losses of transformer model for Participant 1.	68
4.11	Transformer model evaluation result for Participant 1.	68
4.12	LSTM model evaluation result.	69
4.13	Transformer model evaluation result.	70
4.14	LSTM model evaluation without EMG as input for Participant one.	71
4.15	Transformer model evaluation without EMG as input for Participant 1.	71
4.16	Translation model for 300 ASL sentences.	75

List of Tables

1.1	Summary of sign language datasets.	10
2.1	Selected coordinates of right hand joints.	26
2.2	Results of mentioned classical machine learning models in hand gesture classification.	31
2.3	Differences in angle changes of R_Z between non-adjacent joint points.	36
2.4	Four cases for classification evaluation.	38
3.1	Sixty commonly used English words.	44
4.1	Forty commonly used American sign language sentences.	56
4.2	The vocabulary for 40 ASL sentences.	62
4.3	Word error rate comparison for Participant 1.	71
4.4	Sentence error rate comparison for Participant 1.	71
4.5	Word error rate comparison.	72
4.6	Sentence error rate comparison.	72
4.7	User-independent validation results.	73

Chapter 1

Introduction

1.1 Research Background

The world is a colorful place with many kinds of people living here. There is such a group of people that they cannot hear the diverse voices around them and cannot speak pleasant words. They are hearing-impaired people. Sign language is the main communication method among them. According to the World Federation of the Deaf, there are 70 million deaf people around the world using sign language in their daily life [1]. Although the deaf can use sign language to express their views in daily life, there are still many inconveniences. For example, most ordinary people do not understand sign language and cannot communicate with them. Therefore, a sign language translation system can build a bridge between the deaf and the hearing world. To overcome the communication barrier, sign language recognition systems have been specially developed for the hearing-impaired people around the world using various sign languages.

Sign language mainly convey information through the hand movements, hand shape, affiliating trunk movements, facial expressions and lip

movements. Sign language translation is a multidisciplinary field of study involving pattern recognition, signal processing, natural language processing, and linguistics. Recently, the development of sensors' technology have promoted research in human-computer interaction area, such as hand gesture recognition and speech recognition. While speech recognition technology is relatively mature, gesture pattern recognition is still a new research topic. Gestures can be considered as both an independent way of communication and a supplementary form of language. Gestures are consciously or unconsciously used in all aspects of human communication, thus they form the basis of sign language.

There are two main areas of research on sign language recognition: isolated words recognition and continuous sentences recognition. Two kinds of tasks are different from each other. The main task of isolated words recognition is to recognize a single gesture performed by signer. In continuous sentences recognition, the signer needs to perform a sequence of gestures one by one, and the goal is to recognize each gesture properly. There is usually no pause in sign language sentences, instead there is a natural transition between each word. Therefore, it is difficult to segment words accurately in a sentence. Also, it is difficult to judge the beginning and ending moment of each action.

Sign language recognition is a very challenging task. First of all, the number of words in sign language is very large. Secondly, there are certain differences in sign language grammar and rules in different countries and regions and it is difficult to build a unified framework to recognize multilingual sign language. Thirdly, there are great differences between the word order of sign language and spoken language and sign language also have obvious simplification. Finally, due to different body shapes and movement habits, different signers may have different sign language movements for the same contents. In order to complete the task under such

conditions, wearable sensors are selected as the recognition device. Smart wearable devices usually contain a variety of built-in sections, such as linear accelerometers and gyroscopes. In recent years, wearable sensors have been used more and more in researches related to human behavior recognition.

The study of sign language recognition technology can improve the living, learning and working conditions of deaf people by providing them with better service. Especially in some public places like hospitals and train stations, this technology can help the deaf better integrate into the society. Meanwhile, sign language recognition can also be applied to the broadcast of bilingual TV programs, sign language lectures, virtual human research, medical research and many other aspects. With the rise of deep learning methods, pattern recognition based on deep convolution neural network is introduced into sign language recognition task, and the performance is pushed to a new height.

With the rapid development of artificial intelligence technology and the urgent demand for intelligent application of sign language, this paper explores the sign language recognition method based on wearable sensors and natural language processing. Both isolated words recognition and continuous sentences translation are included in this research.

1.2 Related Works

The main purpose of this dissertation is to translate sign language into text sentences. In this section, both wearable-sensors based methods and computer vision-based methods are discussed.

1.2.1 Wearable Sensors-based Methods

The data glove is a kind of device that can record information about hands through the sensors inside, as shown in Figure 1.1. The research based on data glove usually encodes the signal of the bending sensor, i.e. the

movement of each finger joint, into a sign language movement feature vector. Sign language translation is realized by comparing the vector with the pre-set action coding table, or by classifying the feature vector. The glove can accurately capture the movements of all finger joints and the changes of palm shape, so it always shows a good recognition result.



Figure 1.1: CyberGlove data glove [2].

Takahashi and Kishino [3] completed a sign language recognition system based on VPL gloves. This recognition system used VPL data gloves to encode the direction and trajectory of hand movement and completed the recognition of 20 sign language letters. Lee et al. [4] used another kind of data glove as input to sign language recognition system. The system applied hidden Markov model to analyze and recognize the sign language information, and realized the recognition of 14 sign language gestures.

The advantage of data glove is that it can accurately locate the hand and obtain high-precision data through the sensors worn by the signer. Ibarguren et al. [5] collected various information about hand movements by data gloves and the recognition accuracy of hand gesture reached to 90%. Oz and Leu [6] collected the expression information and position changes of sign language gestures and completed the classification of 300 ASL words.

Kadous [7] built a sign language gesture dataset containing 95 words with independent meanings and the accuracy of recognition is 80%. Hemandez et al. [8] used data gloves to recognize 26 English letters. Even in the worst case, the recognition accuracy of the letter “U” can reach to 78%.

MYO armband is another normally used wearable device for sign language data collection. As shown in Figure 1.2, it contains 8-channel surface Electromyography (sEMG) sensors, 3-channel accelerometer, 3-channel gyroscope and 3-channel orientation.



Figure 1.2: MYO armband for gesture recognition [9].

Wu et al. [10] collected inertial data and sEMG data from forearms to detect hand/arm gestures, and 80 commonly used American sign language (ASL) signs were classified by support vector machine classifier. Zhang et al. [11] presented an ASL translation system named MyoSign using the MYO armband as data collection device. The end-to-end translation model consisted of convolutional neural network (CNN), long short-term memory (LSTM) and connectionist temporal classification (CTC) layers. 100 sentences comprised of 70 commonly used ASL words without considering sign language grammar were translated with more than 92% accuracy. Another work using MYO armband was proposed by Tateno et al. in 2020

[12]. They collected data of 20 ASL sentences and treated it as a classification task. The LSTM classifier could recognize these twenty motions with high accuracy among twenty participants.

1.2.2 Computer Vision-based Methods

The input data of sign language recognition system can be in the form of gesture images and a camera can easily capture it. Normally, people decode sign language actions through vision, and visual-based sign language recognition is the most intuitive simulation of human interpretation of sign language. The video can accurately capture rich finger and wrist movements and this is the incomparable advantage of other sign language acquisition methods. At the same time, since many news, weather forecast and other programs contain sign language translation, it provides many publicly available sign language materials with annotations. This facilitates the research of sign language based on camera / video.

Huang et al. [13] used RealSense to obtain the three-dimensional coordinates of 22 joint points of the human hand. With support vector machine method, the accuracy of letters' gesture recognition is 98.9%. Other related works [14, 15] have also studied sign language recognition with RealSense. As a non-contact controller, Leap Motion Controller (LMC) also has relevant applications in the acquisition of sign language data. LMC can run at about 200 frames per second and track hands, fingers and finger like objects. It can model the three-dimensional position coordinates of hands and fingers with 28 kinds of features including fingerprint, palm center, hand direction, etc. Mohandes et al. [16] used naive Bayes classifier to achieve 98.3% recognition accuracy on sign language data collected by LMC. Kinect by Microsoft can provide users with rich multi-modal data. It can obtain the bone information of the whole body, including hand joint points, facial landmarks, RGB color video information and depth video

information. Lee et al. [17] used Kinect to extract information from the bone data and collected X and Y positions of joints like wrist, spine, shoulder and hip.

In recent years, with the wide application of deep learning technology in computer vision, the performance of sign language recognition algorithm based on single-mode RGB video can be comparable to that of depth cameras. In order to easily obtain larger-scale sign language video data and carry out the development of miniaturized and integrated sign language recognition applications, ordinary RGB cameras have gradually become the mainstream tool for sign language data acquisition. Ordinary RGB cameras are cheap and highly integrated, which is very conducive to the practical application of large-scale sign language recognition. Zhang et al. [18] applied ordinary RGB cameras and color gloves to collect sign language video data in an unrestricted environment. Through the modeling of hand position and hand shape, a vision based Chinese sign language recognition system was constructed.

1.2.3 Sign Language Dataset

Previous works about sign language translation mainly falls to two categories: vision-based and wearable sensors-based works. Vision-based methods exploit camera to capture features of hands [19-21]. One most commonly used dataset is RWTH-PHOENIX-Weather [19] (as shown in Figure 1.3), which contains three years' sign language interpretation of daily news and weather forecast from German public tv-station. With this dataset, Pu et al. [22] built model with encoder-decoder structure considering both long short-term memory (LSTM) and connectionist temporal classification (CTC) as decoder. Camgoz et al. [23] applied transformer based architecture to make the model trainable in an end-to-end manner. Huang et al. [24] proposed a novel continuous sign recognition framework: Hierarchical

Attention Network with Latent Space (LS-HAN). Another well-known dataset is CSL [25]. This dataset containing 100 continuous Chinese sign language sentences was collected by Kinect device. Pu et al. [26] proposed a novel architecture with cross modality augmentation and reached state-of-the-art translation accuracy.



Figure 1.3: RWTH-PHOENIX-Weather example of original video frame [19]. The sign language interpreter is shown in an overlay on the right of original video frame.

The detailed information of other datasets published by the academic community is shown in Table 1.1. KSL [27] is a Korean sign language dataset. It is recorded by data gloves including optical fiber sensors, which can record bending angle, position, direction and other information. The HMU [28] is recorded by color gloves and ordinary cameras. The system can obtain the position of each joint point according to the color. Purdue RVL-SLLL [29] and RTWH-Boston 104 [30] are both ASL for continuous recognition. Purdue RVL-SLLL is designed to study prosody and sentence structure. The signer is asked to put his arm down before the beginning. RTWH-Boston 104 uses 3 black-and-white cameras and 1 RGB color camera for recording. 2 black-and-white cameras are facing the face of the signer. The remaining 1 black-and-white cameras shoot the side, and the

color cameras are only used to capture facial expressions. The data acquisition in SIGNUM [31] is carried out under the controlled environment of the laboratory, and the scene is formed by 6 fluorescent lamps illuminated from the front. Videos recorded by DGS Kinect 40 [32] are taken in a controlled environment. The advantage of this dataset is that all sign language actions contain data from multiple perspectives. Boston ASLLVD [33] labels sign language words, start and end time, hand type labels, etc. Unlike other datasets, PSL ToF [34] dataset uses Time-of-Flight (ToF) cameras to complete recording of 84 words. EVISIGN [35] is a large-scale Chinese isolated word sign language dataset, covering standard Chinese sign language vocabulary. Each vocabulary data is composed of RGB video, depth video and skeleton point information. Both MS-ASL [36] and WLASL2000 [37] are recently proposed large-scale isolated sign language recognition datasets, which only contain ordinary RGB videos. MS-ASL is downloaded from video websites and cut into isolated sign language words, including 222 signers. WLASL2000 is recorded by more than 100 presenters, including more than 2000 ASL isolated words. CSLD [38] is one of the largest Chinese continuous sign language datasets recorded by Kinect.

1.2.4 Natural Language Processing

As a kind of natural language, sign language has not become a mainstream research topic in natural language processing (NLP), although the machine translation of spoken or written language is highly accurate today. However, the research of machine translation with deep learning models provides development direction and innovative methods for sign language translation tasks. In order to further the research on end-to-end translation, it is necessary to consider the application of deep learning models.

Table 1.1: Summary of sign language datasets.

Name	Region	Type	Device	Vocabulary	Samples
KSL [27]	Korea	Isolated	Data glove	162	-
HMU [28]	Australia	Isolated	Color glove	22	44
Purdue RVL-SLLL [29]	America	Continuous	RGB camera	104	2576
RTWH-Boston 104 [30]	America	Continuous	RGB,CCD camera	104	201
SIGNUM [31]	Germany	Continuous	RGB camera	455	2340
DGS Kinect 40 [32]	Germany	Isolated	Kinect	40	3000
Boston ASLLVD [33]	America	Isolated	RGB camera	2742	9794
PSL ToF [34]	Poland	Isolated	ToF camera	84	1680
CSL-100 [25]	China	Continuous	Kinect	178	5000
RWTH-PHOENIX-Weather [19]	Germany	Continuous	RGB camera	1200	6841
MS-ASL [36]	America	Isolated	RGB camera	1000	25513
WLASL2000 [37]	America	Isolated	RGB camera	2000	21083
CSLD [38]	China	Continuous	Kinect	9107	49708

ELMo [39] is a new type of deep contextualized word representation, which can model the complex features of words (such as syntax and semantics) and the changes of words in language context. The word vector is the hidden state of bi-directional LSTM model pretrained in a large text corpus. Unlike word2vec, ELMo predicts a word from the context. So, the ELMo performs much better than word2vec in terms of multi-meaning words.

BERT [40] is a breakthrough NLP framework. It is a transformer-based

bidirectional encoder. It is designed to pretrain unlabeled text to obtain deep bidirectional representations by combining the left and right contexts. Therefore, with just one additional output layer, the pretrained BERT model can be fine-tuned to create state-of-the-art results for various NLP tasks. BERT is pretrained in a large amount of unlabeled text, including the entire Wikipedia (2.5 billion words) and Book Corpus (800million words). Bidirectional means that BERT learns information from the context on the left and right sides of the target word in the training steps.

Generative Pretrained Transformer (GPT) [41-43] series is a powerful pretrained language model proposed by OpenAI. This series of models can achieve amazing results in very complex NLP tasks, such as article generation, code generation, machine translation, Q&A, etc. For a new task, GPT only needs very little data to understand the requirements of the task and achieve the state-of-the-art result. The pretraining of a GPT model requires a large training corpus, a large number of model parameters and powerful computing resources. The model structure of GPT series adheres to the idea of continuously stacking transformers, and completes the iterative updating of models by continuously improving the scale and quality of training corpus and the number of network parameters. GPT has also proved that the ability of the model can be continuously improved by continuously improving the model capacity and corpus size.

The task of recognition is to recognize the corresponding gestures or words in complex signals. The input data of these methods are required to be labeled data. For spoken language, the label is usually the corresponding text data. Translation between spoken and sign language usually requires an intermediate language expression. Sign language is annotated with gloss, which is the literal translation of the current words. As a very famous sign language annotation method, HamNoSys [44] can describe the movement and limb position of the signer for computer modeling and analysis. In order

to facilitate structured storage, researchers have developed an XML based markup language. SIGML [45] is a gesture markup language which is widely used and compatible with HamNoSys. Sign language translation system can not only use the intermediate feature expression defined by the current specific language, but also learn its own feature expression. The use of pre-defined representations is usually highly compatible with translation grammar rules, and its representations are usually customized through existing grammar rules [46, 47]. Another representation method is through deep neural network, combining the characteristics of existing data, to learn the feature expression suitable for different tasks. This kind of method has been applied in some related tasks such as translation [48].

1.3 Research Purpose

This research focuses on the topic of sign language translation using wearable sensors. Current sign language translation device is still in its infancy and far away from commercial use. The progress about this topic is relatively slow due to the challenging problems in this area. In this paper, we provide a viable path for starting research into sign language translation with cutting-edge machine learning methods.

The research starts from isolated hand gestures recognition. A benchmark dataset is used to explore existing pattern recognition methods and become familiar with the experimental process of dataset collection. To match the sign language finger spelling, a dataset with hand gestures in alphabet is collected. The time and frequency domain features and position change features of hand joint angles are applied to promote the classification accuracy among multiple users. Finger spelling means signing a sequence of alphabet gestures continuously to form a word. The input inertial data have the same sequence order with the label (letters in a word). Since the input signal is hard to segmented by gestures, an alignment between the output of

machine learning model and the ground truth label is important. Connectionist temporal classification is used as the loss to optimize the model and solve the alignment problem.

The sign language is not always in the same order with spoken language. To solve the problem of difference in order, an end-to-end translation model is ideal for sign language. Like the translators of spoken language, the sign language translator is also expected to translate the input inertial signals into grammatical sentences directly. An encoder-decoder structured model is suitable in this research. As a special part of sign language, facial expressions are also considered. With EMG signals from facial muscles, some kinds of facial expressions can be classified accurately. The expressions information also promotes the translation results.

With the application of cutting-edge machine learning methods, the end-to-end sign language translation using wearable sensors becomes possible. Hope to have better ways to achieve real application in daily life in the future.

Chapter 2

Isolated Gestures Recognition Using Surface Electromyographic and Inertial Data with Pattern Recognition Methods

2.1 Introduction

American sign language contains about 1900 word gestures, 26 letter gestures, and 9 number gestures [49]. For a sign language recognition system, it is necessary to understand hand gestures. We will start the research from isolated gestures recognition. In this chapter, we introduce the commonly used hand gestures benchmark datasets. The IMU data of 26 gestures for ASL alphabet are collected by motion capture system. Some basic methods for feature extraction and pattern recognition are applied in classification tasks.

2.2 Materials and Methods

2.2.1 Benchmark Datasets Description

Non-Invasive Adaptive Prosthetics (Ninapro) is a publicly available resource that aims to support research on advanced myoelectric hand prosthetics. The first Ninapro dataset [50] includes 10 repetitions of 52 different movements of 27 intact subjects. The sEMG data are acquired using 10 Otto Bock MyoBock 13E200 electrodes, while kinematic data are acquired using a Cyberglove 2 data glove (shown in Figure 2.1). As shown in Figure 2.2, the experiment is divided in three exercises: basic movements of the fingers; isometric, isotonic hand configurations and basic wrist movements; grasping and functional movements. For each exercise, for each subject, the database contains one Matlab file with synchronized variables. The variables included in the Matlab files are:

- subject: subject number
- exercise: exercise number
- emg (10 columns): sEMG signal of the electrodes
- glove (22 columns): uncalibrated signal from the 22 sensors of the cyberglove
- stimulus (1 column): the movement repeated by the subject.
- restimulus (1 column): again the movement repeated by the subject. In this case the duration of the movement label is refined a-posteriori in order to correspond to the real movement
- repetition (1 column): repetition of the stimulus
- rerepetition (1 column): repetition of restimulus

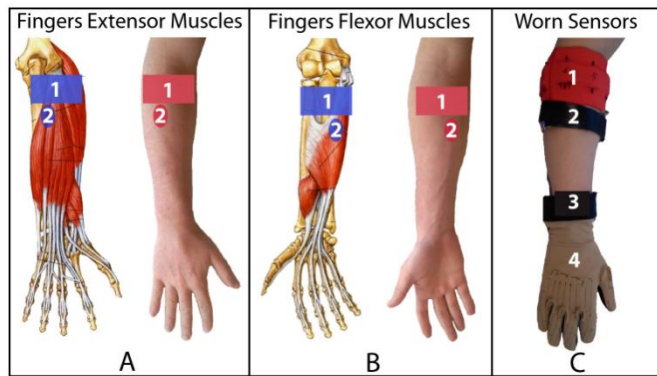


Figure 2.1: Placement of the electrodes [50]: A. sEMG electrodes placed on finger extensor muscles (A.1 Equally spaced electrodes; A.2 Spare electrode); B. sEMG electrodes placed on finger flexor muscles (B.1 Equally spaced electrodes; B.2 Spare electrode); C. all the sensors positioned on the arm (C.1 Equally spaced electrodes; C.2 Spare electrode; C.3 Inclinometer; C.4 CyberGlove II).

Ninapro DB2 [51] adds a special hand dynamics group measured by Finger-Force Linear Sensor. The experiment is divided in three exercises: basic movements of the fingers and of the wrist; grasping and functional movements; force patterns. The first two exercises are the same with that of Ninapro DB1. In the force patterns exercise, the subjects have to press combinations of fingers with an increasing force on a custom made device. The muscular activity is gathered using 12 active double differential wireless electrodes from a Delsys Trigno Wireless EMG system. Eight electrodes are equally spaced around the forearm in correspondence to the radio humeral joint; two electrodes are placed on the main activity spots of the flexor digitorum and of the extensor digitorum; two electrodes are placed on the main activity spots of the biceps and of the triceps. The described locations have been chosen in order to combine a dense sampling approach with a precise anatomical positioning strategy.










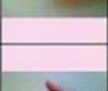

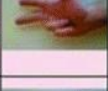























Exercise A			Exercise B			Exercise C					
1	Index flexion		13	Thumb up		30	Large diameter grasp		45	Quadpod grasp	
2	Index extension		14	Extension of index and middle, flexion of the others		31	Small diameter grasp (power grip)		46	Lateral grasp	
3	Middle flexion				15			Flexion of ring and little finger, extension of the others			
4	Middle extension		16	Thumb opposing base of little finger			33		Index finger extension grasp		48
5	Ring flexion				17	Abduction of all fingers				34	
6	Ring extension		18	Fingers flexed together in fist				35	Ring grasp		
7	Little finger flexion				19	Pointing index				36	Prismatic four fingers grasp
8	Little finger extension		20	Adduction of extended fingers				37	Stick grasp		
9	Thumb adduction				21	Wrist supination (axis: middle finger)				38	Writing tripod grasp
10	Thumb abduction		22	Wrist pronation (axis: middle finger)				39	Power sphere grasp		
11	Thumb flexion				23	Wrist supination (axis: little finger)				41	Precision sphere grasp
12	Thumb extension		24	Wrist pronation (axis: little finger)				43	Prismatic pinch grasp		
					25	Wrist flexion				44	Tip pinch grasp
			26	Wrist extension							
					27	Wrist radial deviation					
			28	Wrist ulnar deviation							
					29	Wrist extension with closed hand					

Figure 2.2: Hand gestures of the Ninapro dataset [51].

The electrodes were fixed on the forearm using their standard adhesive bands. Moreover, a hypoallergenic elastic latex free band was placed around the electrodes to keep them fixed during the acquisition. The sEMG signals are sampled at a rate of 2 kHz. During the acquisition, the subjects were asked to repeat the movements with the right hand. Each movement repetition lasted 5 seconds and was followed by 3 seconds of rest. The protocol includes 6 repetitions of 49 different movements (plus rest) performed by 40 intact subjects. The movements were selected from the hand taxonomy as well as from hand robotics literature.

Ninapro DB5 [52] is collected by two MYO armbands and a data glove, as shown in Figure 2.3. The subjects in this dataset wore two Myo armbands one next to the other, including 16 active differential wireless electrodes. The top MYO armband is placed closed to the elbow with the first sensor placed on the radio humeral joint, as in the standard Ninapro configuration for the equally spaced electrodes; the second MYO armband is placed just after the first, nearer to the hand, tilted of 22.5 degrees. This configuration provides an extended uniform muscle mapping at an extremely affordable cost. The MYO sensors do not require the arm to be shaved and after few minutes the armband tighten very firmly to the arm of the subject. The sEMG signals are sampled at a rate of 200 Hz.

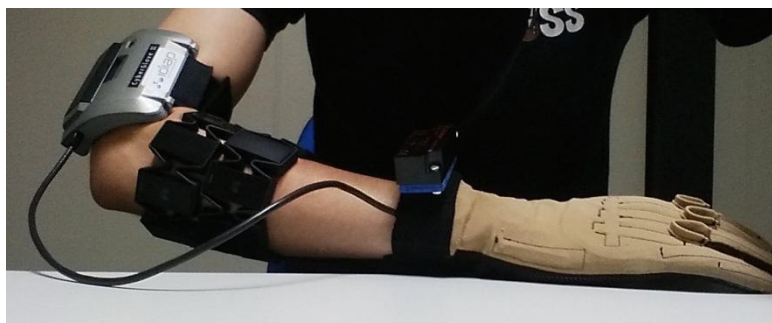


Figure 2.3: Ninapro DB5 data acquisition setup [52].

Besides Ninapro, another commonly use hand gesture recognition dataset

is Capg-Myo [53]. It is recorded by High-density surface electromyography (HD-sEMG) sensors. As shown in Figure 2.4, HD-sEMG is to record muscles' electrical activity from a restricted area of the skin by using two dimensional arrays of closely spaced electrodes. This dataset consists of 3 sub-sets (DB-a, DB-b and DB-c); 8 isometric and isotonic hand gestures were obtained from 18 of the 23 subjects in DB-a and from 10 of the 23 subjects in DB-b, and 12 basic movements of the fingers were obtained from 10 of the 23 subjects in DB-c. All gestures are chosen from Ninapro dataset. Gestures in DB-a and DB-b are equivalent to No. 13-20; gestures in DB-c are equivalent to No. 1-12; Max-force gestures are equivalent to gestures No. 5 and No. 6 in NinaPro.



Figure 2.4: HD-sEMG sensors for Capg-Myo dataset collection [53].

2.2.2 Data Preprocessing for Benchmark Dataset

sEMG signals which are generated by the electrical activity of the muscle fibers reflect the muscle activity and provide limb movement information. Those signals can be noninvasively detected by the surface electrodes. When designing Human Computer Interfaces (HCI), the objective is to create more natural and intuitive interfaces through which human users can interact with computers via speech, touch, or gesture. For this reason, the sEMG based gesture recognition with deep learning approach plays an increasingly important role in human-computer interaction. In recent years,

deep learning techniques achieve promising performance in various fields and provide a new perspective to analyze sEMG for hand gestures recognition. Inspired by the excellent performance of deep learning techniques, the CNN has been exploited for sEMG-based gesture recognition.

NinaPro DB5 dataset is applied for the model's train and evaluation in this section. The DB5 data set is 16 channels of sEMG signal data, and its value is between -128 and 128. The data in the dataset have already been filtered when collecting. The sEMG signals contain 16 channels of 52 gestures repeated 6 times by 10 subjects. Each gesture has a certain label (one of 1-52) attached with it. Since the signal is long and cannot be used as the input of machine learning model directly, windowing method is used to segment the long signal into small frames. The sampling rate is 200Hz and 300ms is chosen as one window, so the length of each segment of signal is 60. A novel method is processing the input data as an image, which can regarding the amplitude as grayscale (shown in Figure 2.5). In time domain, feature extraction is considered to compare the final result with raw data.

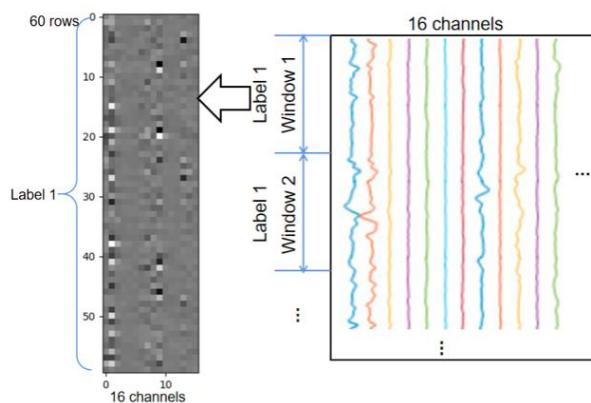


Figure 2.5: Sampling the dataset into small sEMG images.

Five features are selected for the EMG signal: Root Mean Square (RMS), Mean Absolute Value (MAV), Wave Length (WL), Zero Crossing (ZC) and Slope Sign Change (SSC).

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^n x^2(i)}$$

$$\text{MAV} = \frac{1}{N} \sum_{i=1}^N |x(i)|$$

$$\text{WL} = \sum_{i=1}^{N-1} |x(i+1) - x(i)|$$

$$\text{ZC} = \sum_{i=1}^{N-1} [\text{sgn}(x(i) \times x(i+1)) \cap |x(i) - x(i+1)| \geq \text{threshold}]$$

$$\text{SSC} = \sum_{i=2}^{N-1} [\text{sgn}[(x(i) - x(i-1)) \times (x(i) - x(i+1))]]$$

2.2.3 Isolated Gestures for ASL Alphabet

There are 26 signs for letters in ASL (shown in Figure 2.6). These 26 special signs are always used to spell proper nouns like names or spell a strange word that the signer forgot how to this word in sign language gesture. Some studies treat the hand gestures for letters as static gestures. However, the letter “j” and letter “z” are both dynamic process, and these studies have to ignore them to become 24 letters [54-56]. In this study, we will treat all gestures as dynamic process as most studies [57-60]. An example is shown in Figure 2.7, the sign starts from a rest state, and finally returns to the original rest state.

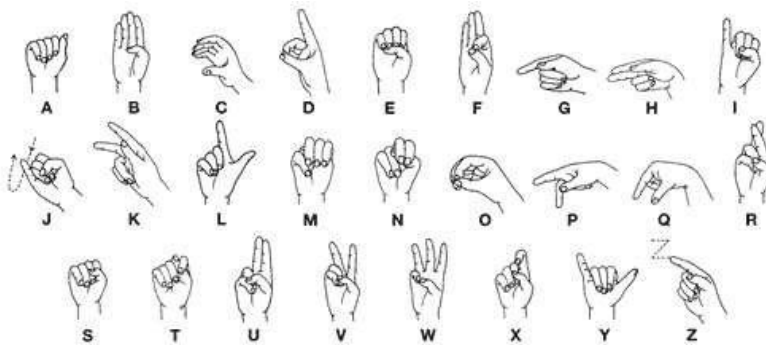


Figure 2.6: 26 signs of American sign language letters.



Figure 2.7: Dynamic process of gesture “j”.

2.2.4 Perception Neuron Motion Capture System

Perception Neuron (shown in Figure 2.8) is a wearable IMU sensors-based motion capture system. It can be used for different applications in the fields of VFX, game interaction, virtual reality, sports analysis, medical analysis and realtime stage performance among others. VFX: Motion recording and replaying; BVH and FBX format output supported for Maya, MotionBuilder, Blender, etc; VR / Game interaction: Seamless integration with HMD with open source game engine demos, gesture control library included; bring ‘yourself’ into the virtual world; Sports / Medical Analysis: Get all orientation, position, and raw acc and gyro data flow in real-time, free ‘Data-Visualizer’ for data plotting and recording / comparison; Stage performance: Real-time data stream output, wireless data transmission.

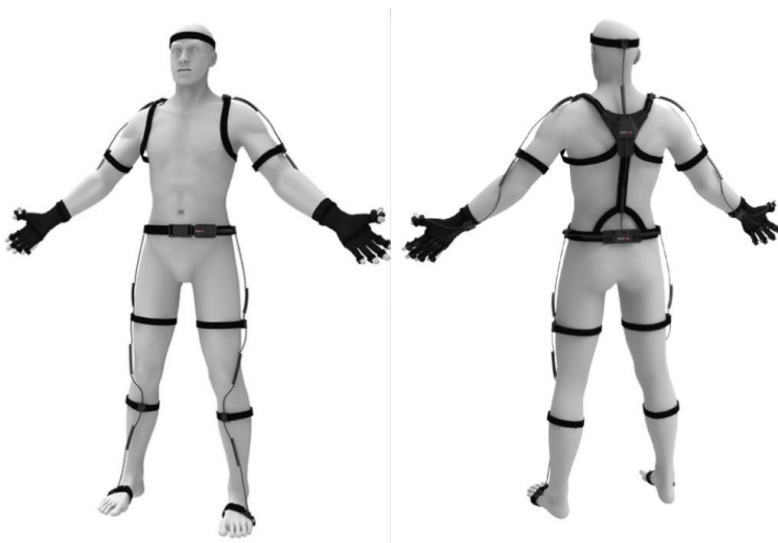


Figure 2.8: Perception Neuron motion capture system.

The Perception Neuron motion capture system needs to communicate with the Axis Neuron software. As shown in Figure 2.9, Axis Neuron can receive and process the motion data and export it to a third-party software. It is also possible to do synchronous broadcasting with a third-party software. In addition, high-quality motion data can be collected, which is compatible with most professional film effects and game development tools.

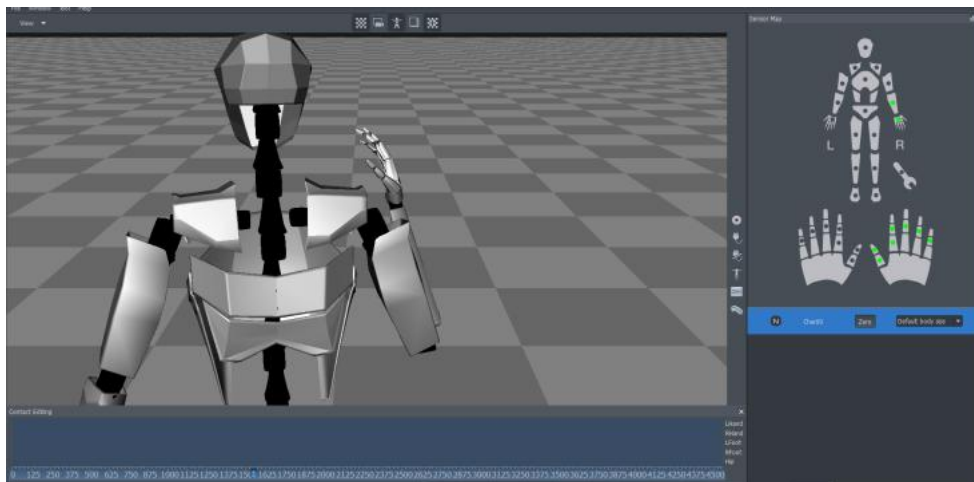


Figure 2.9: Main interface of Axis Neuron software.

The output of Axis Neuron software is .bvh file. The bvh data contains two parts: Hierarchy and Motion. Hierarchy describes the information of all joints in the human skeleton. Motion includes sampling rate and motion data of the whole recorded moving process. The motion data with displacement will include 6 floats for each and every 59 bones (shown in Figure 2.10): 3 for the displacements (X Y Z) and 3 for rotation data (Default rotation order is Y X Z). For the motion data without displacement, except the root node (Hip) having displacement and rotation, other bones will only stream rotation data.

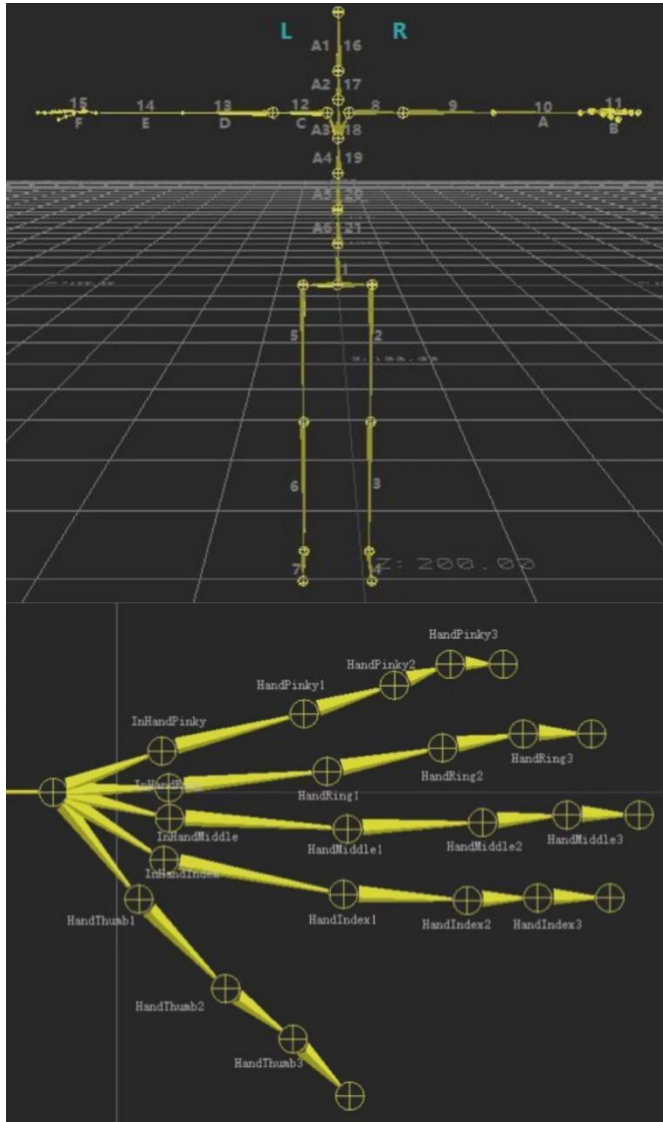


Figure 2.10: Fifty-nine bones of human skeleton model.

2.2.5 Dataset Collection

The sign language data is collected by Perception Neuron motion capture system. This device contains 31 IMU sensors equipped on whole body. Since the signs of letters are performed by the right hand, we only focus on the data of right-hand sensors (shown in Figure 2.11).

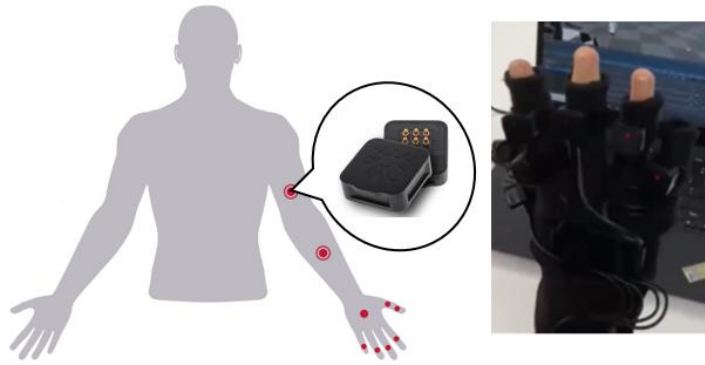


Figure 2.11: Right-hand motion capture device.

Axis Neuron is the official software of motion capture system. It records the data of each unit in real time during sign language performance. The position and rotation of each joint is calculated automatically according to sensors' data and the human-body model on the main interface can reproduce real human movement. After exporting .bvh file, we receive the joints' position and rotation data.

There are 26 letters in the dataset. Four volunteers participated in the experiment and each hand gesture was repeated 20 times. Finally, there are 2080 samples in total.

2.2.6 Data Preprocessing for Collected Dataset

For each joint in human body, the displacement and rotation data has already been calculated by the Axis Neuron software after exporting .bvh file. The displacement is actually a redundant feature, because it is calculated by rotation data and body length. For the exported coordinates (Rotation_Y, Rotation_X, Rotation_Z), they describe the hand movements of Radial deviation / Ulnar deviation, Pronation / Supination, and Extension / Flexion. Since all participants have a similar height of around 160cm, we consider them as sharing the same body size, and the length of each part of the body keeps the same. After observing the exported data, we manually remove redundant information. As a result, only the rotation data of selected

right-hand joints (listed in Table 2.1) is selected as the useful data of movement. The hand skeleton model can be simplified as Figure 2.12.

Table 2.1: Selected coordinates of right hand joints.

Joint Name	Coordinates Selection
Right Hand	R_Y, R_X, R_Z
Right Thumb 1	R_Y, R_Z
Right Thumb 2	R_Y, R_Z
Right Thumb 3	R_Y
Right Hand Index 1	R_Z
Right Hand Middle 1	R_Z
Right Hand Ring 1	R_Z
Right Hand Pinky 1	R_Z

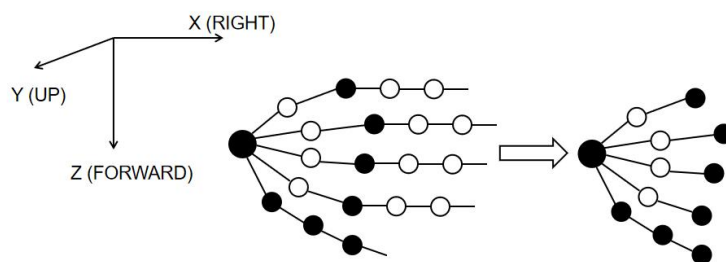


Figure 2.12: Useful information selection for right hand joints.

The sample rate of device is 120Hz, which is a little bit lower than normal IMU device. Each movement of sign lasts for around 2 seconds. So, we resample the data to 256 in the time axis direction, to guarantee all samples have the same length. We do not apply any filters to the collected data to avoid losing important features.

Since the signal is long in time axis, sliding window method is used to

segment the sequence into data frames (shown in Figure 2.13). The hanning window with window length of 300ms (32 points) is applied to data sequence, and the sliding window size is 150ms (16 points). Finally, we receive the data for each gesture with a shape of (15, 32, 12).

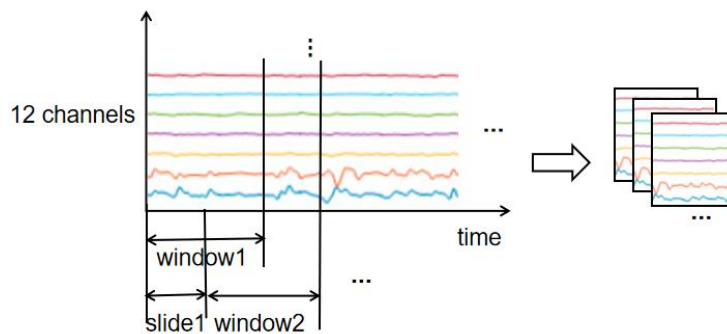


Figure 2.13: Signal segmentation with sliding window method.

2.2.7 Model Design

It is encouraging to build a custom model for hand gesture classification task with Ninapro DB5 dataset. In this section, we propose a custom hand movement classification system based on sEMG time-frequency features and deep learning method. The complete process is shown in the Figure 2.14.

The sEMG signal sequence is segmented into frames by Hanning window (window size 300ms, sliding size 150ms). Short-time Fourier transform (STFT) is applied on each frame to get advanced feature representation as input features to the deep learning model. As shown in Figure 2.15, the model structure contains CNN as feature extractor and LSTM as classifier. Domain adaptation is applied in the model to alleviate data distribution difference between training and testing dataset. The domain classifier is a linear classifier to judge whether the data from training set or testing set. We hope the domain classifier cannot judge where the data

comes from. Finally, the results of both hand gesture classification and domain classification are consider while training the model.

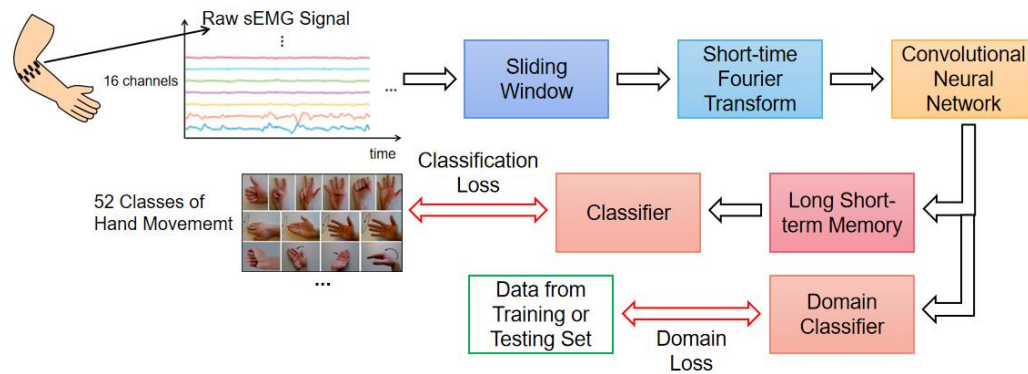


Figure 2.14: The complete process of costume hand gesture classification system.

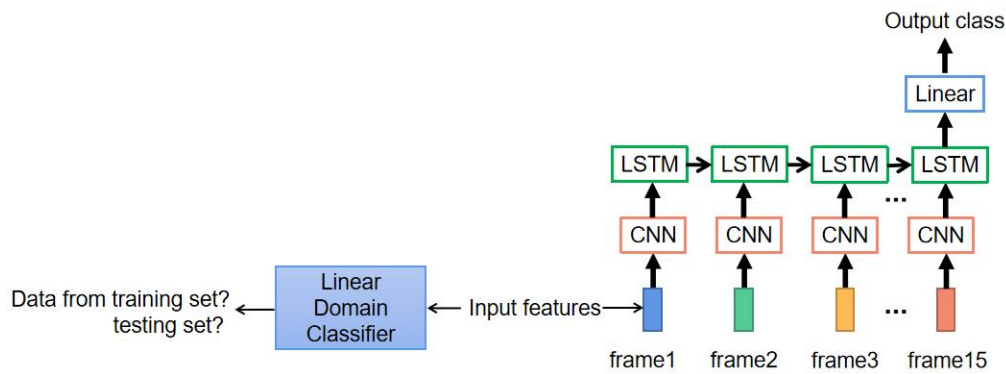


Figure 2.15: A custom machine learning model for hand gesture classification.

In self-collected dataset, we receive the raw data of right-hand joints information after data preprocessing. A classification task is needed here to check whether the data of letters has a high quality. Firstly, a CNN feature extractor is applied to extract useful features from raw data. CNN is a popular feature extractor that has been used in many areas, such as image, speech and text. Here, we will consider raw data after sliding window (shape (15, 32, 12)) as input. The classification model is shown in Figure 2.16. We employ 2D convolution filters with kernel size of 3×3 to learn high-level representations. Pooling is applied on each layer to reduce the dimension. We use ReLU as the activation function. In addition, we apply

batch normalization (BN) at each convolutional layer. BN guarantees the input distribution of each layer remains unchanged across different mini-batches, leading to much faster training speed for CNN. The classifiers consist of 2 fully connected layers. A softmax function finally calculates the probabilities of all classes and chooses the class with largest probability as the output (top 1 accuracy).

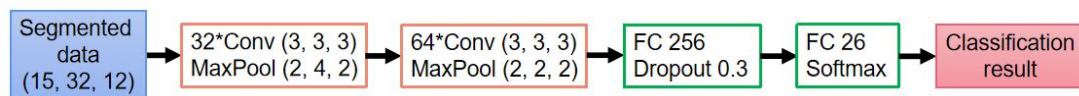


Figure 2.16: Hand gestures classification model with sliding window method for input segmentation.

2.3 Results

2.3.1 Gestures Classification Using Benchmark Dataset

Some classical machine learning models with CNN layers are often used as classifiers. LeNet [61] proposed by Lecun is the first work of CNN. It reduces parameters by sharing convolution kernels. The kernels are all square in shape, because this network was used for image processing at first. The input shape of sEMG image is $60 \times 16 \times 1$, so kernels with shape of rectangle are also considered. The convolution layers are normally used for feature extraction, and fully connected layers work as a classifier to output the result of classification. Cross entropy is used as the loss during backpropagation. 90% of all data is used for training the model and the remaining 10% is the testing set. The model gets a testing accuracy of 0.50 after adjustment, and the main reason for low accuracy is that the model is small with limited layers in depth. The highest accuracy is 60% for LetNet.

AlexNet [62] was proposed in 2012. It was the first time that network

training was accelerated with GPU. The Relu activation function is used to promote the speed of training and Dropout [63] method is used to alleviate overfitting. This model has a similar structure with LeNet, but it is much larger and deeper. After adjusting the parameters of the model, the accuracy of prediction in testing set only reaches to 30%. The result goes worse because the model is too large and dataset is not large enough to train the model.

InceptionNet (googleNet) [64] introduces a special Inception block to use different sizes of kernels in the same layer. The original design idea was to increase the network width and avoid the explosion of number of channels. In this way, different dimensions of features can be extracted and the perception of the model is improved. The famous batch normalization (BN) is firstly proposed and adopted by this work [65]. BN is a very useful regularization method, which can speed up the training speed of convolution network and improve the classification accuracy after convergence. BN standardizes the interior of each batch of data during training and normalizes the output to the normal distribution of $N(0,1)$. To some extent, BN also plays the role of regular, so dropout can be reduced or canceled to optimize the network structure. The model performs well in training process at an accuracy of 91%. But the testing accuracy is only 52%, which is still lower than expect. The mainly reason for lower testing accuracy is overfitting. The model contains too many parameters for limited amount of training data.

Another method of extracting features in different dimensions is using multi-channel CNN. The model applies 2 channels. The first channel uses small kernels and the second channel uses large kernels. The features extracted are mixed together to form the input to the fully connected layers. After training the model, the accuracy keeps nearly the same with InceptionNet at 50% accuracy in testing set, but the model structure is more simple.

A deeper network does not always means better, because it may lose the information of former layers. To solve this problem, inter layer residual connection is used in ResNet [66] to remember the former information. As a result, the gradient disappearance is alleviated and the network can be built much deeper. There is a clear improvement in accuracy using ResNet at 98% in training set and 70% accuracy in testing set, which illustrates that ResNet model is suitable for sEMG-based gesture classification.

The results of gesture classification accuracy by classical machine learning models are listed in Table 2.2. It is clear that ResNet has the highest accuracy among all models and this model is suitable for gesture classification. But the result is still lower than state-of-art accuracy for NinaPro DB5. To reach better result, further adjustment of parameters is needed.

Table 2.2: Results of mentioned classical machine learning models in hand gesture classification.

	Train	Test
LeNet (raw signal input)	0.50	0.50
LeNet (feature input)	0.58	0.60
AlexNet	0.35	0.27
InceptionNet	0.91	0.52
ResNet (raw signal input)	0.98	0.70
ResNet (feature input)	0.97	0.65

In the custom model for hand gesture classification task with Ninapro DB5, the total loss of the model is cross entropy loss of classification and binary cross entropy loss of domain adaptation. The training and testing steps are shown in Figure 2.17 and Figure 2.18. The training loss and testing loss drop dramatically in first 25 epochs and keep nearly flatten until the end.

The accuracy of training set is nearly 100% and the accuracy of testing set finally reaches to more than 80% after 100 epochs training of the model. The result indicates that this method achieves high accuracy in hand movement classification and it can be a useful tool in real-life neuroprosthetic controlling applications.



Figure 2.17: Training and testing loss of the model.

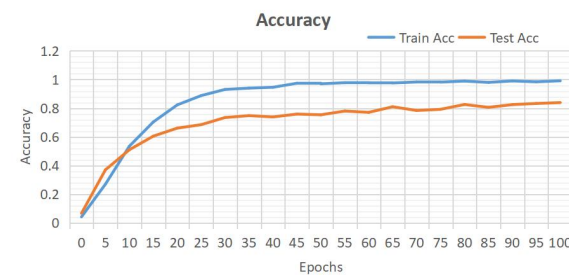


Figure 2.18: Training and testing accuracy of the model.

2.3.2 Self-Collected Dataset of ASL Alphabet

From the collected dataset, we randomly select 80% of all data as training set, and the remaining 20% as testing set. The training and testing processes are shown in Figure 2.19 and Figure 2.20. According to the training result, the model tends to converge after 7 steps, and the losses of both train set and test set drop to nearly 0 (the accuracy increase to nearly 1.0). The model can classify the 26 hand gestures perfectly without considering user independence.

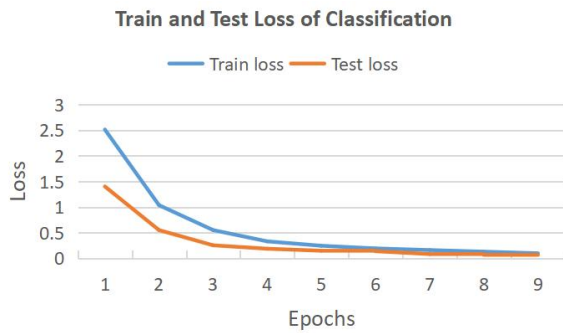


Figure 2.19: Loss for model training and testing steps.

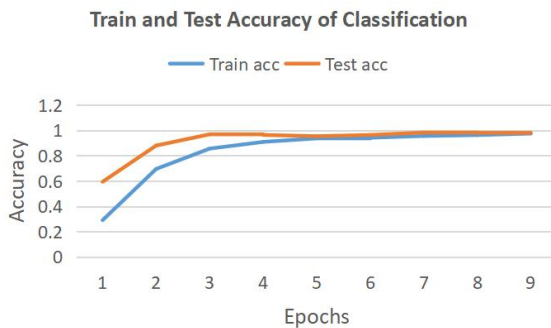


Figure 2.20: Accuracy for model training and testing steps.

Four participants were included in the experiment. To do the user independent validation, we use the data of 3 participants to train the model and leave the remaining 1 participant as testing set. As shown in Figure 2.21, the total average accuracy of these four groups drops to nearly 70%. Participant 3 and 4 show higher accuracy than participant 1 and 2. The influence factors for the drop of accuracy include the difference of body size, range of motion, and different understanding of gestures among participants which leads to different hand movements.

To observe the classification results more intuitively, we draw the confusion matrices of participant 3 (85.5% accuracy) and participant 4 (77.9% accuracy) in Figure 2.22 and Figure 2.23.

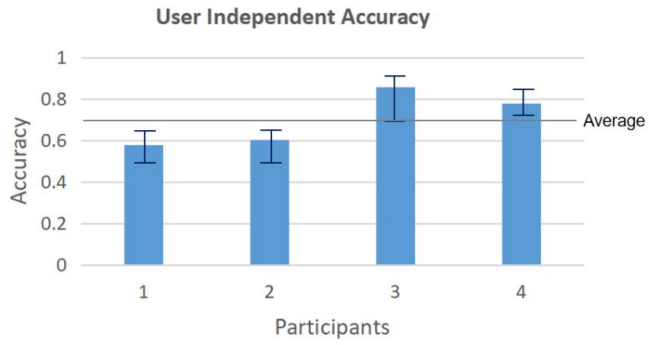


Figure 2.21: User independent validation accuracy of 4 participants.

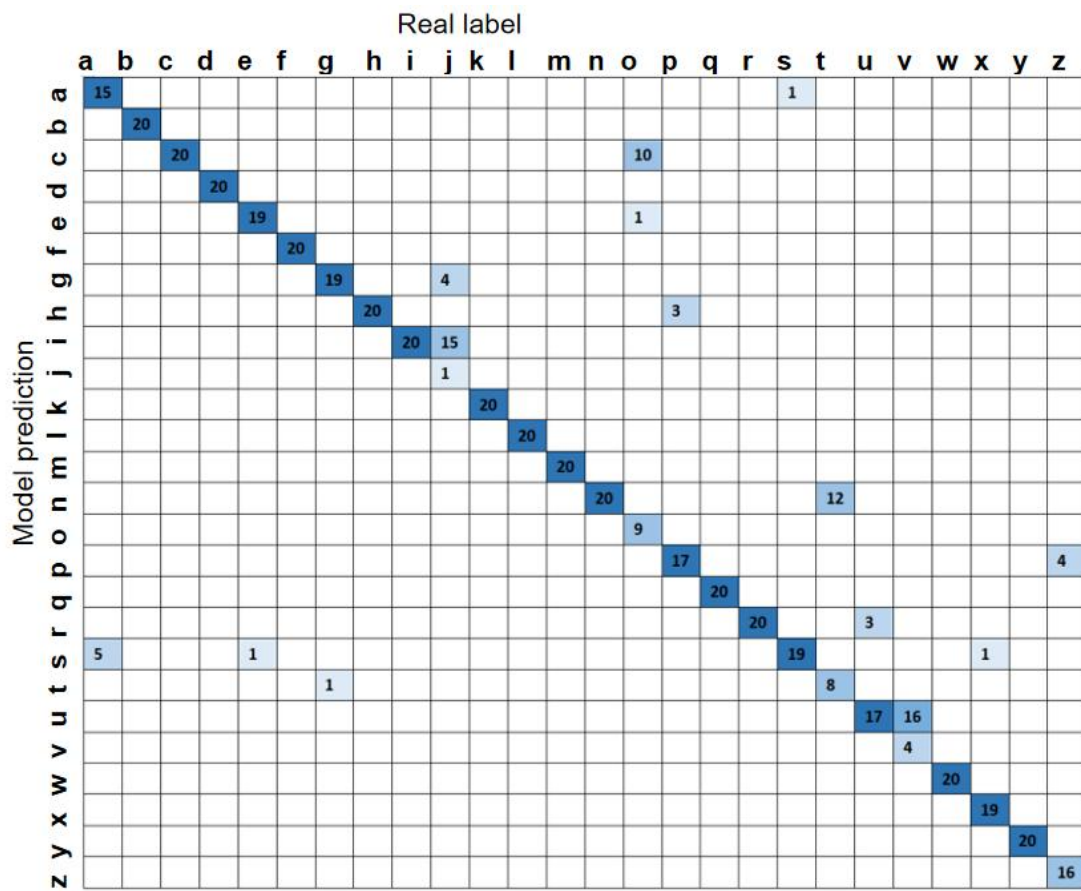


Figure 2.22: Confusion matrix of participant 3 (best among 4 participants).

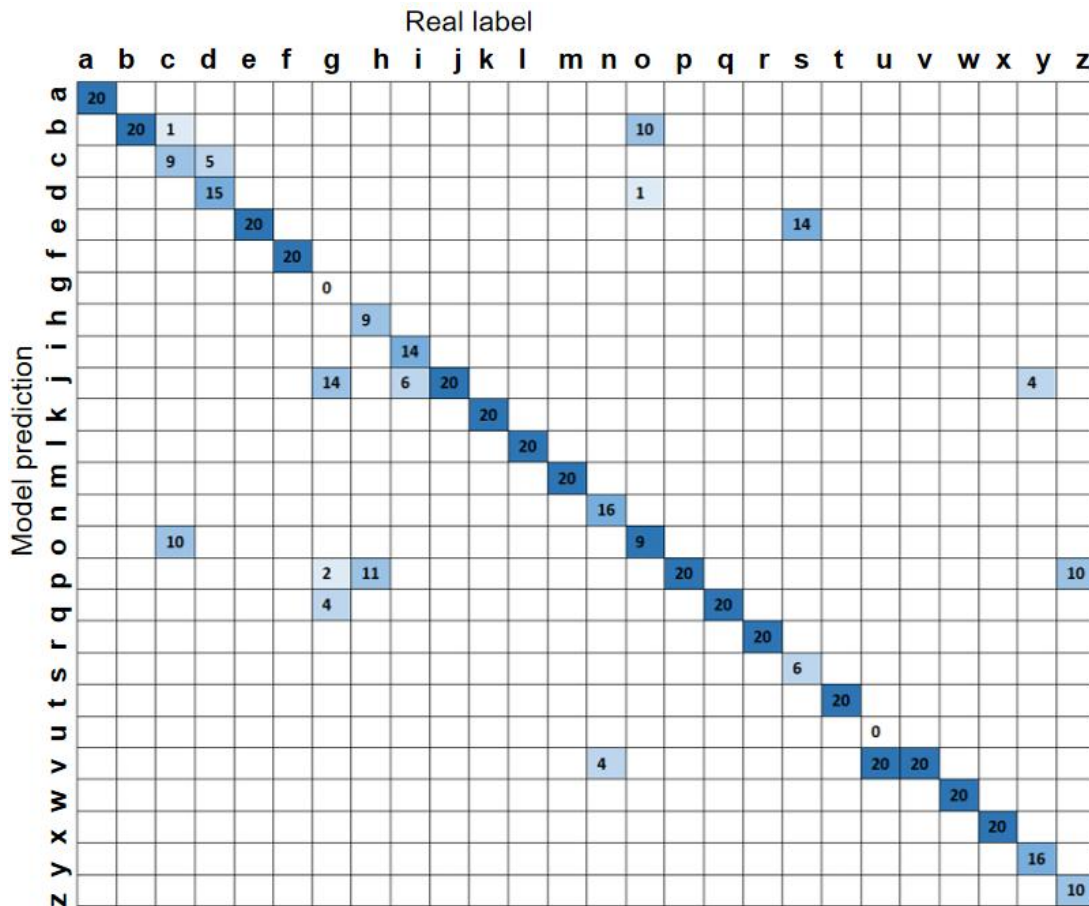


Figure 2.23: Confusion matrix of participant 4 (accuracy 77.9%).

For participant 3, the model tends to misread “j” to “i”, and “v” to “u”. The gesture for “j” is using the pinky finger to write a “j”, and this contains the hand gesture of “i” with only stretching the pinky finger. The difference of gesture “v” and “u” is the distance between the index finger and middle finger. For participant 4, besides misreading “u” to “v”, the model also makes more mistakes between “g” and “j”. The reason is that both gestures contain the movements of pointing to the left.

2.4 Discussion

2.4.1 Features Selection

In order to promote the classification accuracy, time domain features (RMS,

MAV, WL, ZC, SSC) and time-frequency domain features (STFT, DWT) are selected as one feature input.

Another feature we selected is the differences in angle changes of R_Z between non-adjacent joint points [67]. The original names of selected hand joints are marked in Figure 2.24. According to the positional relationship between the joint points, we divide the angle changes feature into four groups as listed in Table 2.3.

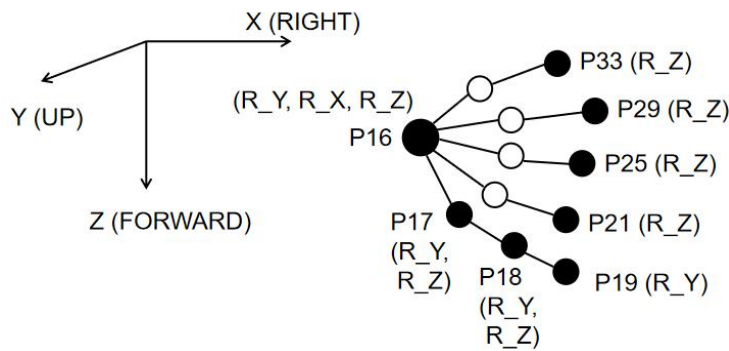


Figure 2.24: Selected right hand joints with original joint numbers.

Table 2.3: Differences in angle changes of R_Z between non-adjacent joint points.

Group name	Coordinates
Group 1	P33-P16, P29-P16, P25-P16, P21-P16, P19-P16, P18-P16
Group 2	P33-P17, P29-P17, P25-P17, P21-P17
Group 3	P33-P18, P29-P18, P25-P18, P21-P18
Group 4	P33-P29, P33-P25, P33-P21, P29-P25, P29-P21, P25-P21

2.4.2 Early Fusion and Late Fusion Models

Since we have two kinds of multi-features (time and frequency domain features and angle changes of non-adjacent joint points, as shown in Figure 2.25) as input, early fusion and late fusion models are both considered as

classification model (shown in Figure 2.26). The early fusion model concatenate all input data together from the start. The late fusion model concatenate features together after convolutional layers.

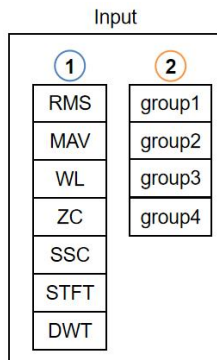


Figure 2.25: Two kinds of multi-features as input.

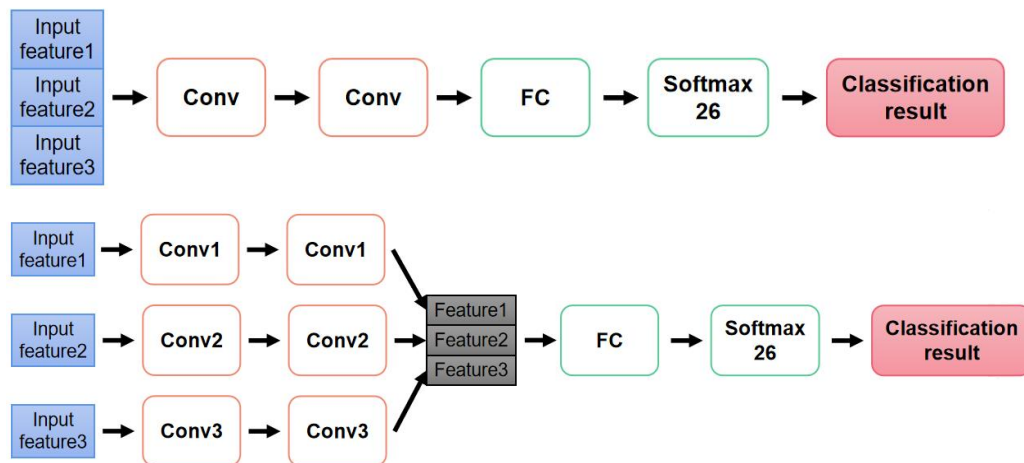


Figure 2.26: Early and late fusion of classification models.

2.4.3 Features Selection Results

We use the following four cases listed in Table 2.4 to verify the average classification accuracy. Selected features of participant 1, 2, 3, 4 are regarded as testing set respectively, and the remaining three participants are used to train the model. The classification results of four cases are shown in Figure 2.27. The average accuracy reaches (69.0%, 54.0%, 86.8%, 80.6%) for each participant, which is (11.4%, -6.1%, 1.3%, 2.7%) higher than the

raw data classification results of (57.6%, 60.1%, 85.5%, 77.9%). The total average accuracy of the whole dataset increases by 2.3%. The selected features show a better performance than the raw data. Case 4 (angle changes group features with late fusion model) shows the highest accuracy among the selected four cases at 74.8%, which is 4.5% higher than the raw data, and 2.2% higher than the average accuracy of four cases.

Table 2.4: Four cases for classification evaluation.

Case 1	Time & frequency features, Early fusion model
Case 2	Time & frequency features, Late fusion model
Case 3	Angle changes features, Early fusion model
Case 4	Angle changes features, Late fusion model

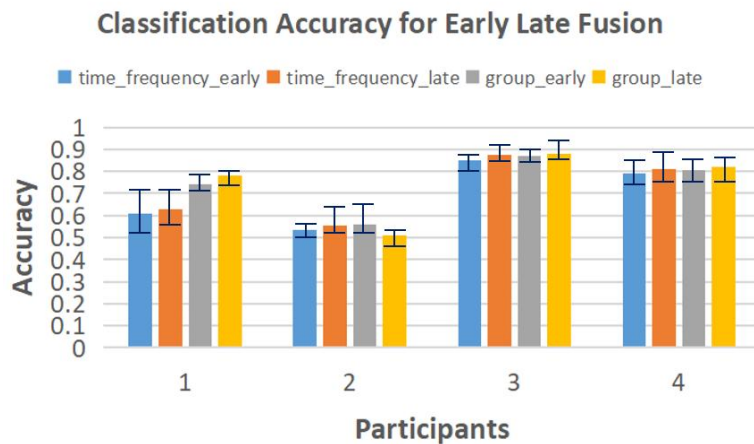


Figure 2.27: Classification accuracy of four kinds of cases.

The classification results with only one feature as input is shown in Figure 2.28. Compared with the raw data, STFT feature achieves a higher accuracy. Other features appear relatively lower results when applying only one feature as input, because the amount of known information is too small. The result illustrates that it is encouraging to combine multiple features

together.

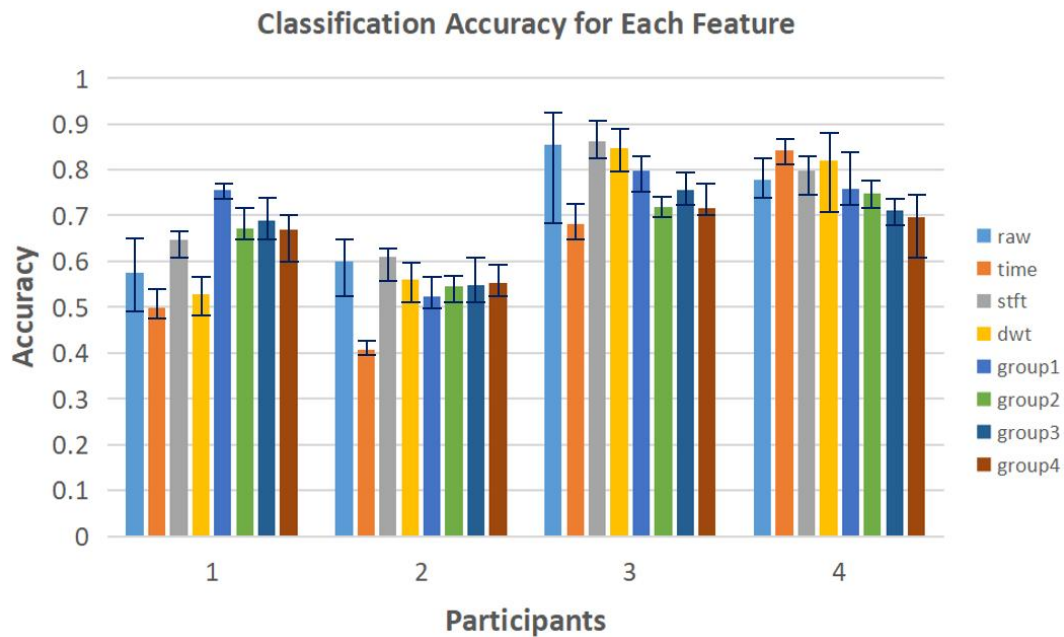


Figure 2.28: Classification results with each selected feature.

The confusion matrices of all participants under four cases (time frequency features early fusion, time frequency features late fusion, selected group features early fusion, selected group features late fusion) is shown in Figure 2.29 to Figure 2.32. In general, participant 3 gives the best performance, and participant 2 gives the worst results. Compared with the raw data confusion matrices, the results of participant 1, 3, and 4 are improved, but the average accuracy of participant 2 drops a little. In reality, participant 2 has relatively specific body shape among all participants. If more participants are included in this experiment, this problem will be alleviated.

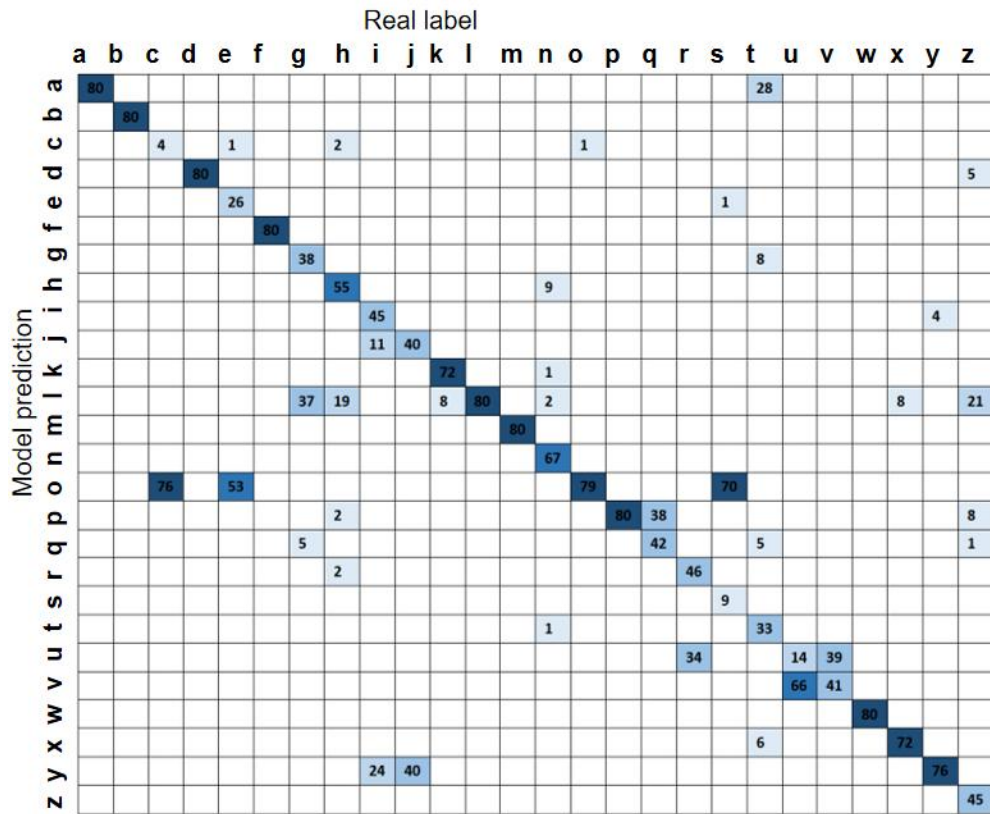


Figure 2.29: Confusion matrix of all cases for participant 1.

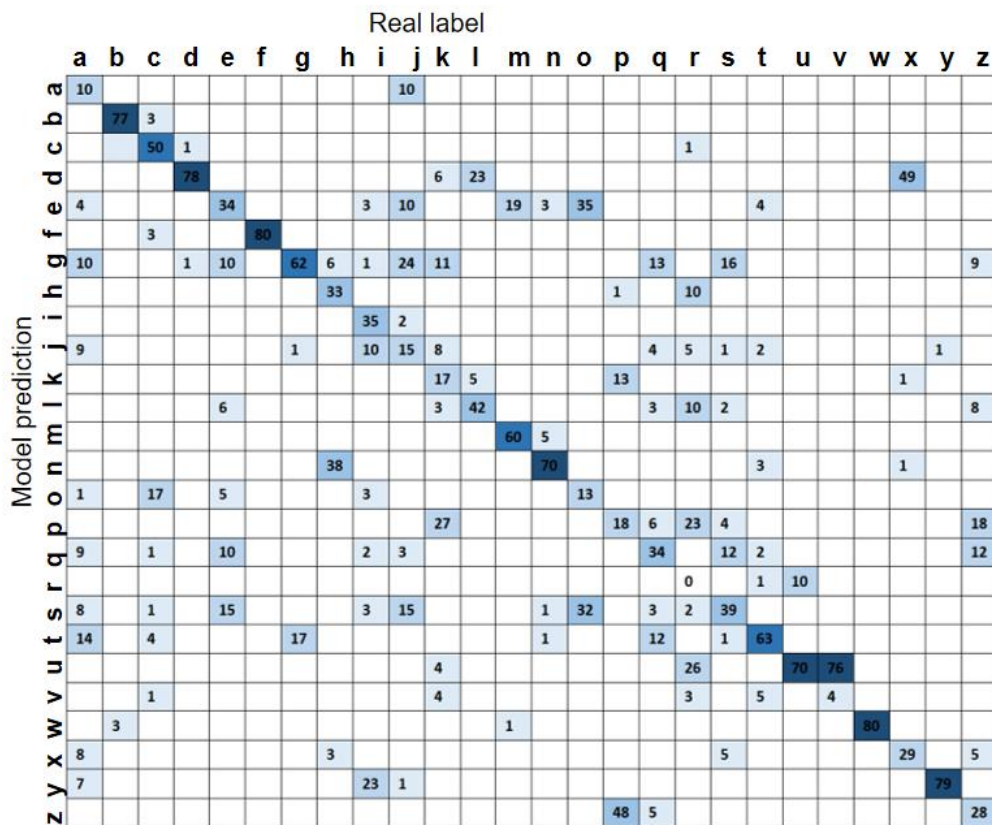


Figure 2.30: Confusion matrix of all cases for participant 2.

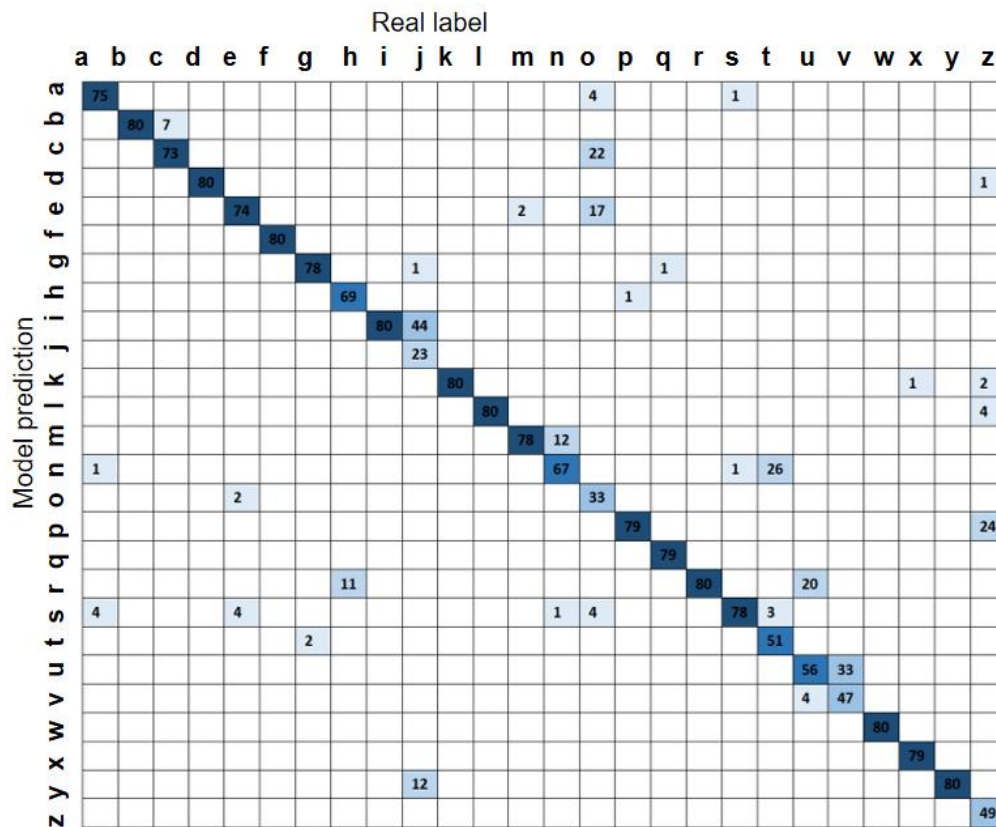


Figure 2.31: Confusion matrix of all cases for participant 3.

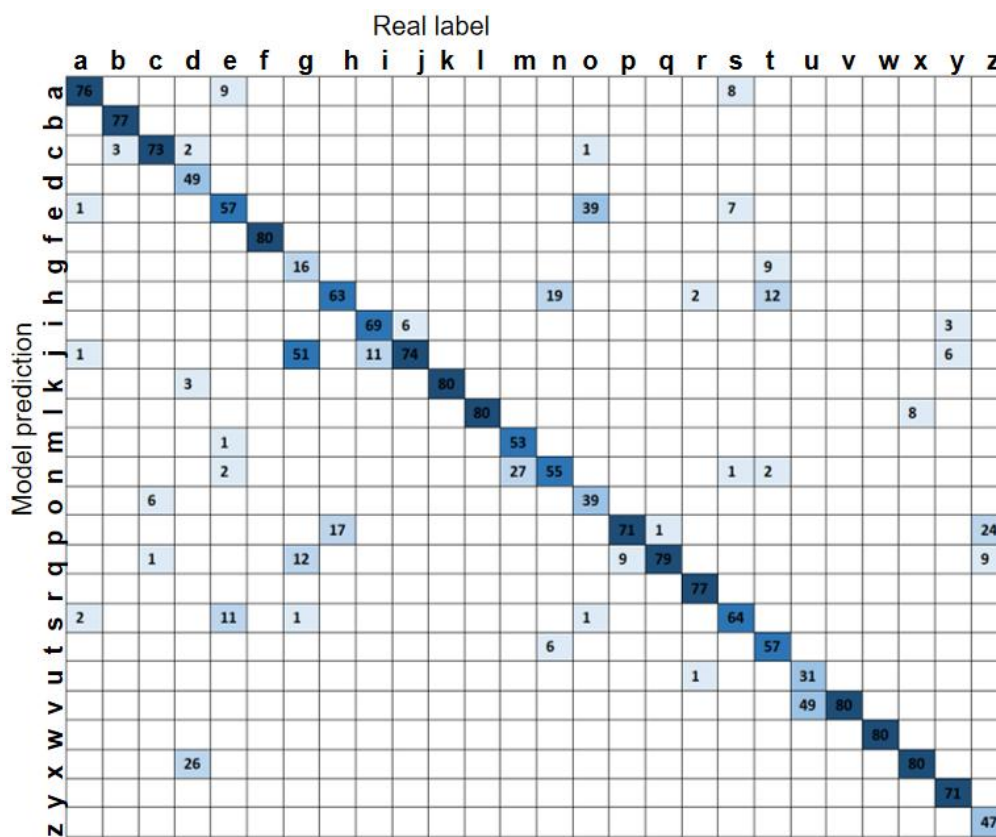


Figure 2.32: Confusion matrix of all cases for participant 4.

The classification results of letter “g” and letter “u” still show low prediction accurate rate by misread to letter “j” and letter “v”. Although the accuracy increases a little by features selection, we still hope for a much better method to solve this problem. Intuitively, letter “g” and “j” both include moving to the left side, but the hand shape is different. We use the raw data of P33 (pinky joint 1), P21 (index joint 1), and P17, P18, P19 (right thumb joint 1, 2, 3) to do the binary classification. The binary accuracy can reach 97.9%. The letter “u” and “v” have different angles between index finger and middle finger. Use differences in angle changes between P25 (middle joint 1) and P21 (index joint 1) as input, the binary classification accuracy for “u” and “v” is 100%.

2.5 Conclusions

In this chapter, we firstly introduced the widely use hand gesture recognition datasets of Ninapro. Then, we did the classification task on Ninapro DB5 dataset with customized machine learning model. After that, we collected an isolated hand gestures dataset according to ASL alphabet. The device is IMU motion capture system. Some commonly used features were selected for promoting the classification accuracy.

Chapter 3

ASL Finger-Spell Translation Using Hand Motion Capture System

3.1 Introduction

In American sign language, people sometimes need to sign a proper noun like name or brand. There are no corresponding words in ASL dictionary for these special words. So it is necessary to spell these words with a sequence of letters from ASL alphabet. In this chapter, we use the above mentioned IMU motion capture device to recognize a sequence of continuous hand gestures.

3.2 Materials and Methods

3.2.1 Words Dataset

The finger spelling means signing a sequence of letters continuously to form a word (an example is shown in Figure 3.1). We collect dataset of a sequence of letters to do the sequence recognition task. Sixty commonly

used words are selected to do the word-recognition task. These words are shown in Table 3.1. Three participants finished the finger spelling experiment. Each word was repeated for 20 times. Finally, there are 3600 samples in total in the words dataset.



Figure 3.1: Finger spelling of “world”.

Table 3.1: Sixty commonly used English words.

time	person	year	way	day	thing
man	world	life	hand	part	child
eye	woman	place	work	case	point
company	number	group	problem	fact	be
have	do	say	get	make	go
know	take	come	think	want	give
use	find	ask	try	leave	new
first	last	long	great	own	other
old	right	week	see	look	tell
seem	feel	call	good	little	dazzle

3.2.2 Algorithm Introduction

Connectionist temporal classification (CTC) has been widely used in speech and handwriting recognition [68] as it eliminates the need to know the alignment between input and output. We extend the letter classes with an

extra label “blank”. The blank label accounts for “do not belong to any class”. For example, the alignments $(t,-,i,i,-,-,m,m,-,e,-,-,-)$, $(t,-,-,i,-,-,m,-,-,e,-,-,-)$ and $(t,t,i,i,i,m,m,e,e,-,-,-,-)$ are all correspond to the word “time”. Another example “world” is shown in Figure 3.2.

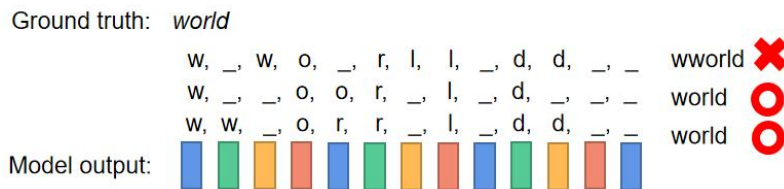


Figure 3.2: How to form a “world”.

All predicted sequences that can be converted into real sequences by the mapping function are correct prediction results. That is, the prediction sequence can be obtained without data alignment processing. The object is to maximize the probability sum of all correct prediction sequences. A forward-backward algorithm is used to find all correctly predicted sequences. The forward process calculates the probability of predicting the correct prefix from time 1 to t ; the backward process calculates the probability of predicting the correct suffix from time t to T .

Define a transform B , that reduce consecutive identical letters to 1 and remove “blank”. In this way, the model output is transformed into real letters sequence.

$$B(\pi^1) = B(-- stta - t ---- e) = \text{state}$$

$$B(\pi^2) = B(sst - aaa - tee) = \text{state}$$

$$B(\pi^3) = B(-- sttaa - tee -) = \text{state}$$

$$B(\pi^4) = B(sst - aa - t ---- e) = \text{state}$$

The following probability needs to be maximized:

$$p(L|x) = \sum_{B(\pi)=L} p(\pi|x)$$

Where the ground truth L is a sequence. Assuming that the outputs between time steps are independent, the probability calculation formula for any output sequence π is as follows:

$$p(\pi|x) = \prod_{t=1}^T y_{\pi_t}^t$$

π_t is the index corresponding to the element selected by the output sequence at time step t . For example, the element selected in the first time step of the sequence is “a”, then the obtained value is 1. If “z” is chosen, the resulting value is 26. If the “blank” is selected, the resulting value is 27.

$$\pi = (\text{--- stta - t --- e})$$

$$p(\pi|x) = y_{-}^1 \cdot y_{-}^2 \cdot y_s^3 \cdot y_t^4 \cdot y_t^5 \cdot y_a^6 \cdot y_{-}^7 \cdot y_t^8 \cdot y_{-}^9 \cdot y_{-}^{10} \cdot y_{-}^{11} \cdot y_e^{12}$$

Derivation of a letter at a time step happens to be the path associated with probability y_k^t .

$$\frac{\partial p(L|x)}{\partial y_k^t} = \frac{\partial \sum_{B(\pi)=L, \pi_t=k} p(\pi|x)}{\partial y_k^t}$$

Take the above mentioned $\pi^1, \pi^2, \pi^3, \pi^4$ as examples, the two paths are shown in Figure 3.3.

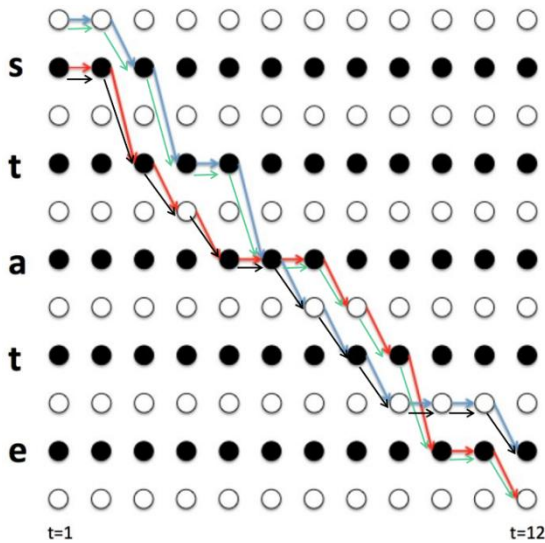


Figure 3.3: The proper path for π^1 (blue), π^2 (red), π^3 (green), π^4 (black).

The four paths all pass through the “a” at t=6. Observe the four paths, and the following formula can be obtained.

$$p(\pi^1, \pi^2, \pi^3, \pi^4|x) = \text{forward} \cdot y_a^t \cdot \text{backward}$$

The above forward and backward only contain 4 paths. If the meaning of forward and backward is generalized, considering all paths, it can be expressed as follows:

$$\sum_{B(\pi)=L, \pi_6=a} p(\pi|x) = \text{forward} \cdot y_a^t \cdot \text{backward}$$

Define forward as $\alpha_t(L'_k)$, representing the sum of the probabilities of 1 to t in the path probabilities passing through L'_k at time t.

$$\alpha_t(L'_k) = \sum_{B(\pi)=L, \pi_t=L'_k} \prod_{t'=1}^t y_{\pi_{t'}}^{t'}$$

Initial conditions:

$$\alpha_1(-) = y_-^1$$

$$\alpha_1(L_1) = y_{L_1}^1$$

$$\alpha_1(L_t) = 0, \forall t > 1$$

More generally, the following recurrence relation can be obtained according to Figure 3.4.

$$\alpha_t(L'_k) = (\alpha_{t-1}(L'_k) + \alpha_{t-1}(L'_{k-1}) + \alpha_{t-1}(-)) \cdot y_{L'_k}^t$$

The backward represents the sum of the probabilities of t to T in the path probabilities of passing L'_k at time t.

$$\beta_t(L'_k) = \sum_{B(\pi)=L, \pi_t=L'_k} \prod_{t'=t}^T y_{\pi_{t'}}^{t'}$$

Initial conditions:

$$\beta_T(-) = y_-^T$$

$$\beta_T(L'_{|L'|-1}) = y_{|L'|-1}^T$$

$$\beta_T(L'_{|L'|-i}) = 0, \forall i > 1$$

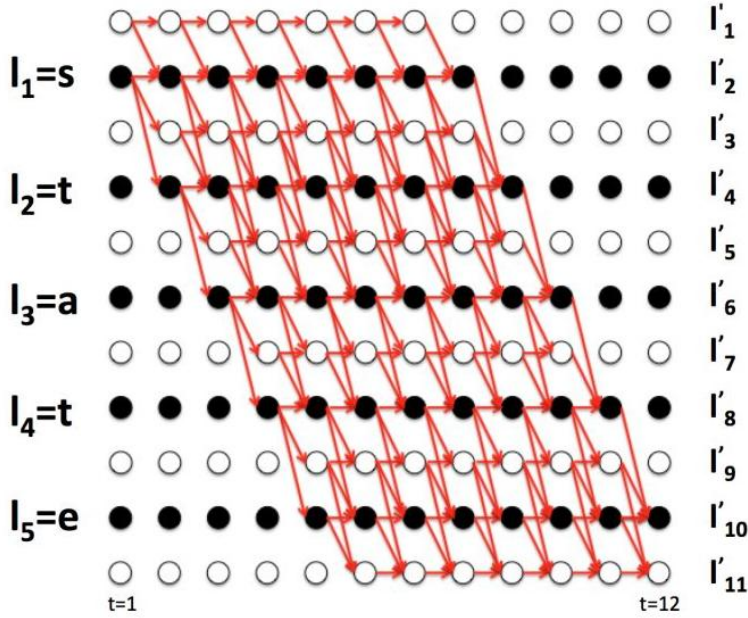


Figure 3.4: All possible paths for correct alignments.

Similarly, the following recurrence relation can be obtained:

$$\beta_t(L'_k) = (\beta_{t+1}(L'_k) + \beta_{t+1}(L'_{k+1}) + \beta_{t+1}(-)) \cdot y_{L'_k}^t$$

According to the definition of forward and backward, they can be multiplied to get:

$$\alpha_t(L'_k)\beta_t(L'_k) = \sum_{B(\pi)=L, \pi_t=L'_k} y_{L'_k}^t \prod_{t=1}^T y_{\pi_t}^t$$

When $p(L|x)$ taking the derivative of L'_k , only the paths related to $\pi_t = L'_k$ are used. Finally, the derivative can be obtained.

$$\frac{\partial p(L|x)}{\partial y_k^t} = \frac{\partial \sum_{B(\pi)=L, \pi_t=k} \frac{\alpha_t(k)\beta_t(k)}{y_k^t}}{\partial y_k^t}$$

3.2.3 Sequence to Sequence Model Design

The machine learning model is illustrated in Figure 3.5. The first layer is CNN that extracting features for each frame of raw input data. The second layer is LSTM. LSTM is widely used in speech recognition, language modeling, and translation to model temporal dependence. As an extended model of Recurrent Neural Network (RNN), LSTM can preserve the long-term dependencies by controlling the percentage of previous information dropping, current information inputting, and current information outputting [69]. Figure 3.6 shows the LSTM expanded by time step and the detailed structure of the LSTM unit.

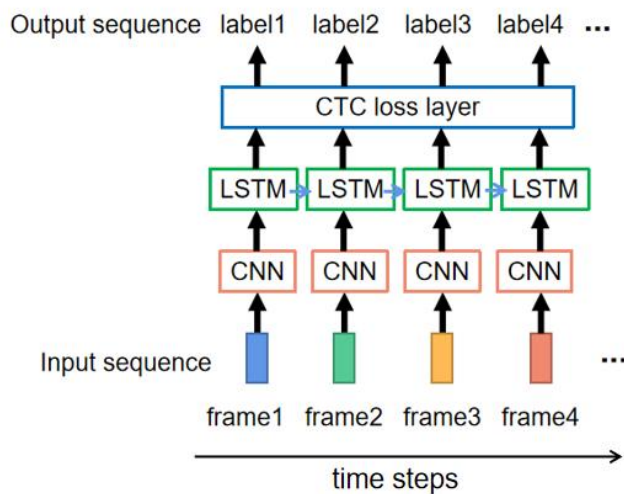


Figure 3.5: Sequence recognition model.

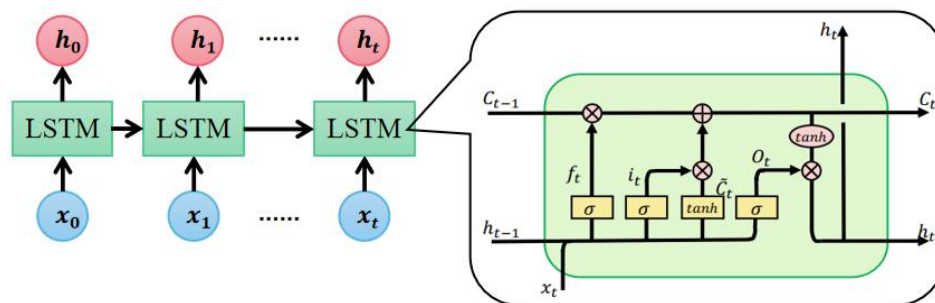


Figure 3.6: Detailed structure of LSTM unit.

The cell state C_{t-1} and hidden state h_{t-1} from the previous time step along with the current input x_t are the inputs to the current LSTM unit. The forget gate f_t , input gate i_t , update gate \tilde{C}_t , and output o_t are calculated as follows:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_c) \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \end{aligned}$$

Where σ is the sigmoid function, and W, b are weights and bias respectively. With these results, C_t and h_t are updated:

$$\begin{aligned} C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\ h_t &= o_t * \tanh(C_t) \end{aligned}$$

The hidden vectors C_t and h_t passed to the decoder are used as the initial hidden state of decoder LSTM.

CTC is used to solve the alignment problem between network output and ground truth label. After feature extraction and classification of each frame, the sequence of results is compared with the real label sequence by CTC loss, and determine whether the output sequence is correct.

$$\text{loss} = -\log \sum p(\text{align}|\text{input})$$

The input to the model is 13 frames of raw data and the output from the model is 13 frames of probability distribution (26 letters and blank). Since the ground truth only contains 2-7 letters, the CTC loss can merge the same adjacent letters and then remove the blank “_”.

3.3 Results

3.3.1 Words Level Classification

There are 60 words in the dataset, each of which is composed by 2 to 7 letters. Since 60 is not a large number for classification task, we can still use classification method to recognize each word after adding class labels to samples. The length of each sample can be different because of different word length, so we pad the data to the same length with 0. Since the data is segmented into frames, CNN is used here and the depth direction of input data is the time step direction. The machine learning model is shown in Figure 3.7.

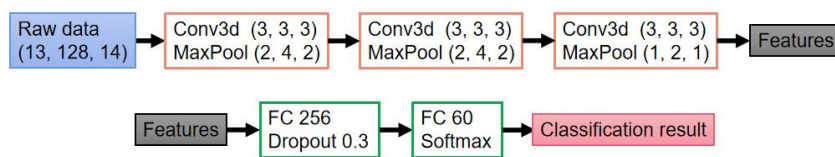


Figure 3.7: Word level classification model.

The training result shows a similar trend with letters classification task, but the model converges much faster at 3 steps. The losses of both train set and test set drop to nearly 0 at 5 steps. The classification result is shown in Figure 3.8. It is clear that the model can classify 60 classes of different words.

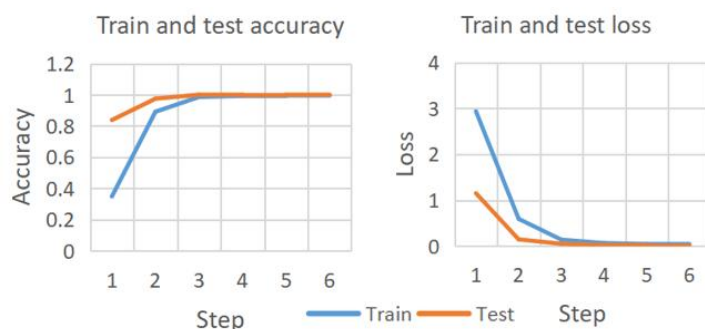


Figure 3.8: Word level classification results.

3.3.2 Sequence Recognition Results

While training the model without considering individual, the loss of train set drops evidently from the first steps, as shown in Figure 3.9. In the decoding of test steps, we only choose the label with largest probability for each frame (beam search, beam=1). An example is shown in Figure 3.10. The test-loss curve fluctuates and reaches the lowest point. Use early-stopping method to stop training the model in step 20 and receive a test accuracy of 0.976. The sequence recognition model also performs well.



Figure 3.9: Sequence recognition result without considering individual.



Figure 3.10: An example of alignment with largest probability.

With ten-fold cross validation, the dataset is randomly divided into ten subsets. We leave each subset as testing set and use the remaining nine subsets to train the model. The completely correct words accuracy of ten subsets are shown in Figure 3.11. the average accuracy of ten subsets is

86.4%. No subset shows big deviation from the average value. This is lower than the isolated gestures classification result, because the word consists of several letters.

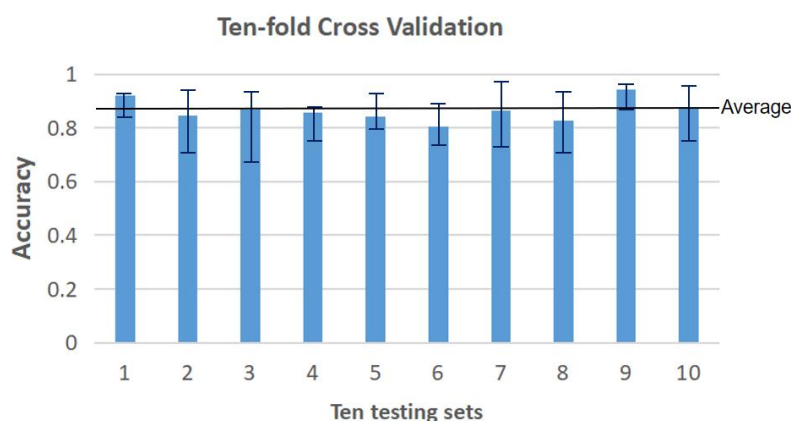


Figure 3.11: Ten-fold cross validation of sequence recognition.

3.4 Discussion

In words level classification, the testing accuracy is relatively high. The reason is that the words are different in length, and the letters that make up the words are different. In sequence recognition, the accuracy drops to 86.4%. When treating the word as a sequence of letters, the model needs to predict all the letters correctly to form a correct prediction. So, the sequence recognition is more challenging. But it could tell the meaning of output from the model for each frame of input.

3.5 Conclusions

In this chapter, we collected IMU data of right hand during ASL finger spelling words performance, and did the classification tasks in word level in advance. Then, sixty words were recognized with a sequence recognition machine learning model. The sequence to sequence model showed higher

accuracy, and that was suitable for the finger spell of sign language translation. In the future, more data including real sign language performance in daily life would be collected.

Chapter 4

Sign Language Translation Using Wearable Inertial and Electromyography Sensors for Tracking Hand Movements and Facial Expressions

4.1 Introduction

Sign languages are not exactly expressed with hands. It is also critical to catch facial expressions. In this chapter, a novel American sign language translation method based on wearable sensors is proposed. We leverage inertial sensors to capture signs and surface electromyography sensors to detect facial expressions. We apply a convolutional neural network to extract features from input signals. Then, long short-term memory and transformer models are exploited to achieve end-to-end translation from input signals to text sentences. We evaluate two models on 40 American sign language sentences strictly following the rules of grammar. Word error rate and sentence error rate are utilized as the evaluation standard.

4.2 Materials and Methods

4.2.1 ASL Specifics

American sign language is a kind of visual language expressed via a sequence of sign gestures. A sign consists of four main components, i.e. hand shape, movement, palm orientation, and location. Besides, facial expression can also be critical to expressing the signer’s current mood. For example, raised eyebrow always indicates asking a question and a neutral face conveys a statement of fact. In addition to neutrality and questioning, positive and negative emotions are also considered in this research. 40 commonly used sentences (listed in Table 4.1) with emotions positive, negative, questioning, and neutral are selected for recognition. These 40 sentences come from popular sign language videos on the Internet. The signers perform these sentences with obvious facial expressions.

Table 4.1: Forty commonly used American sign language sentences.

Positive	Negative	Questioning	Neutral
1. I’m happy!	11. today I feel sad.	21. are you deaf?	31. I’m fine.
2. wow the steak is delicious!	12. I don’t like cat.	22. are you finish?	32. I’m busy.
3. happy new year!	13. why you are sad.	23. are you alright?	33. I need help.
4. merry Christmas!	14. I’m afraid of spider.	24. do you want milk and cookies?	34. you like him.
5. wow the dessert is delicious!	15. running, growing up, I hate it.	25. do you like ice-cream?	35. I go to church on Sunday.
6. haha the commercial is funny!	16. I don’t know where, sad.	26. are you happy with studying history?	36. I’m a broke college student.
7. with you I’m happy!	17. my friend dislikes wrestling.	27. do you come to church on Sunday?	37. I go to beach this summer.

8. happy thanksgiving!	18. his wife dislikes cooking.	28. do you also want fries?	38. we are hungry.
9. happy mother's day!	19. I'm worried. they are angry.	29. did you finish eating vegetable?	39. I go back home.
10. this year we are happy!	20. I feel annoyed.	30. does this food have strawberry?	40. they enjoy eating hamburgers.

4.2.2 Dataset Collection

The movements of forearms and hands are obtained by the Perception Neuron motion capture system. As shown in Figure 4.1A, this system is based on wearable IMU sensors named “Neuron”. Each Neuron is composed of an accelerometer, gyroscope, and magnetometer. There are 25 Neurons for capturing upper body movements. The motion capture system needs to communicate with the Axis Neuron software. Axis Neuron can receive and process the data from all IMU sensors and export it into a .bvh format file. In this file, skeleton information and movement information of the whole process is recorded. We only use the motion data which records the rotation information of all joints of the human body. We only focus on the data from hands and forearms’ joints. The sampling rate is 60Hz.

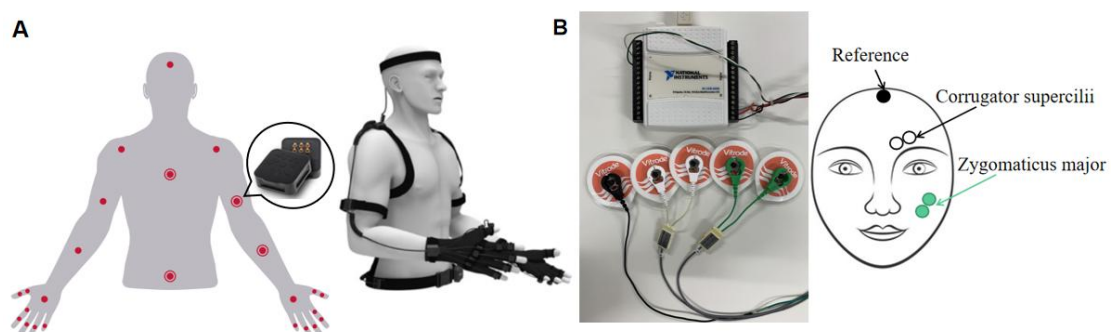


Figure 4.1: Devices for data collection: (A) Perception Neuron motion capture system; (B) EMG signal acquisition system.

EMG measures the electrical activity generated by the muscle. Figure

4.1B shows a 2-channel EMG signal acquisition system. The system mainly includes an NI data collector and differential electrodes. The NI USB-6008 provides eight single-ended analog inputs. Four single-ended analog inputs are used to form two differential channels. Another grounded channel is used as a reference. The electrode applied in this system is wet silver/silver chloride (Ag/AgCl) surface electrode. The useful information of EMG signals is mainly distributed in the frequency range of 0-500 Hz [70]. To meet the Nyquist sampling theorem, the sampling rate is chosen as 1 kHz.

In the experiment, EMG signals from zygomaticus major and corrugator supercilii areas and IMU signals from forearms and hands were collected. Three participants with the right hand as the dominant hand participated in data collection. The signers performed each sign language sentence with both hand movements and facial expressions. Participant 1 contributed the largest amount of data (1600 samples). Participants 2 and 3 each contributed 400 samples. Finally, there are 60 samples for each sentence and 2400 samples in total in the dataset.

4.2.3 Data Preprocessing

The .bvh data from the IMU motion capture system includes all the motion data of 59 bones. We only focus on the data from hands and forearms. Finger spacing is fixed in Axis Neuron software, as a result, some channels maintain the same values throughout the experiment. We manually remove these channels containing no useful information. Finally, only 38 channels remain for the inertial data of forearms and hands. Since the IMU signals are sampled with a much lower sampling rate, we only use a median filter with a kernel size of 5 to make data smooth. The signal preprocessing flow is shown in Figure 4.2.

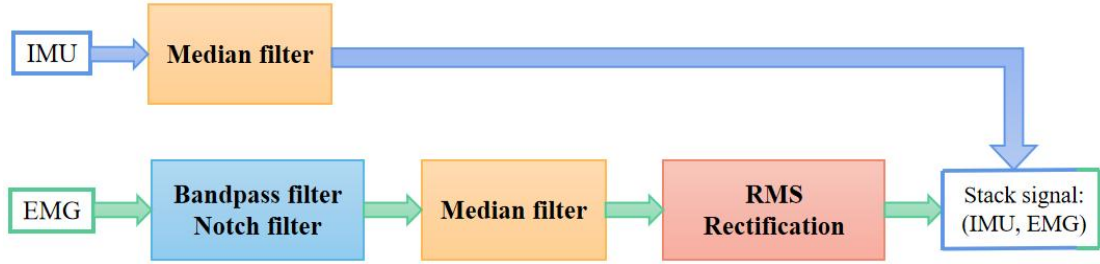


Figure 4.2: Signal preprocessing flowchart.

Compared to IMU, the EMG signal is much noisier and unstable. To maintain EMG features' performance, the signal is band-pass and notch filtered to remove power-line interference and motion artifacts [71]. Then, a median filter is used to smooth the data. Rectification is a commonly applied approach to magnify the EMG features [72]. The Root-Mean-Square rectification of signal $x(t)$, is defined as

$$EMG_{rect}(t) = \sqrt{\frac{1}{T} \int_{t-T}^T x^2(\tau) d\tau}$$

Where T is the window size that controls the trade-off between smooth envelopes against transient variations of EMG signal. We set this value to be 0.02 seconds to avoid signal distortion and to keep approximately consistent in length with the IMU signal according to the sampling rates of the two devices. The lengths between EMG and the corresponding IMU signal may be different, so we resample the EMG to the same length as IMU in the final step of preprocessing. An example of the EMG data from sentence No. 21 before and after preprocessing is shown in Figure 4.3.

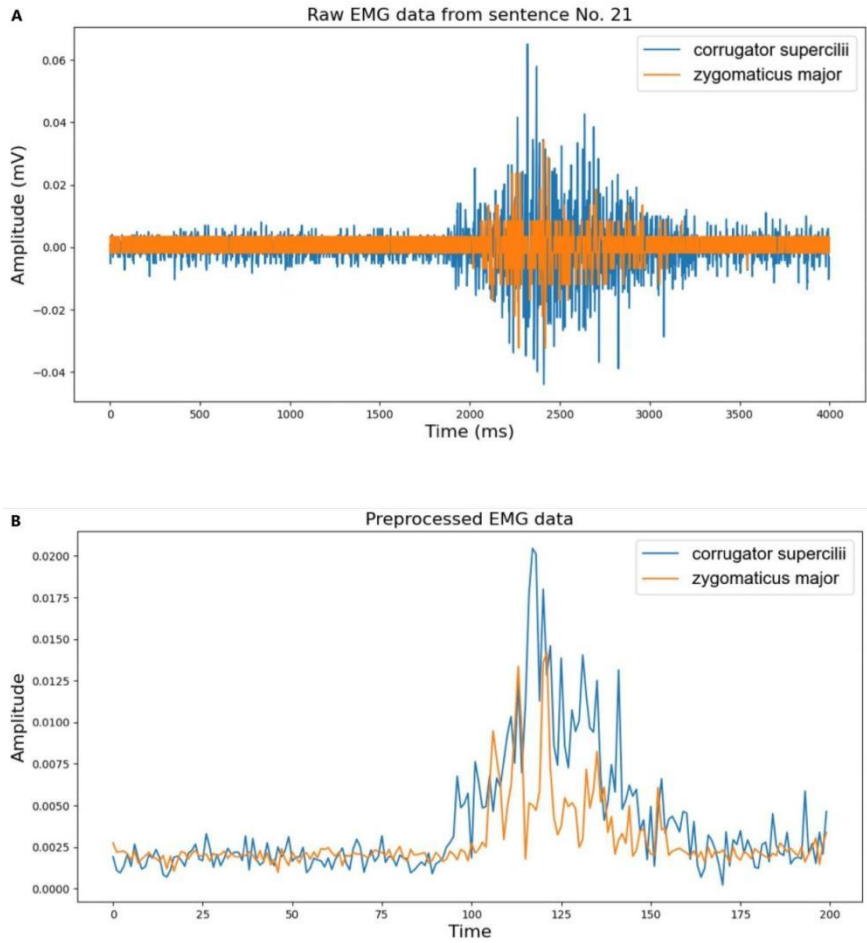


Figure 4.3: An example of EMG data preprocessing: (A) Raw EMG data from sentence No. 21; (B) Corresponding preprocessed EMG data.

4.2.4 Facial Expressions Classifier

CNN is an effective technique to solve signal and image classification problems. Based on shared-weights architecture, CNN eliminates effects from motion differences in amplitude and trajectory. An emotional classifier using CNN as a feature extractor is proposed in this research.

The CNN classifier mainly consists of four layers as shown in Figure 4.4. The first two layers are convolutional layers with 9×1 and 5×1 kernels, respectively. Since the input EMG signal contains two independent channels, to avoid any confusion, the convolutional kernels are both 1-D kernels.

Batch normalization is used for reducing internal covariate shift, and rectified linear unit (ReLU) is selected as the activation function. Max pooling is set to reduce the computational burden. The following layer is a fully-connected layer with a dropout strategy to prevent overfitting. Finally, there is a fully-connected layer with G-way softmax. G is the number of facial expressions to be recognized.

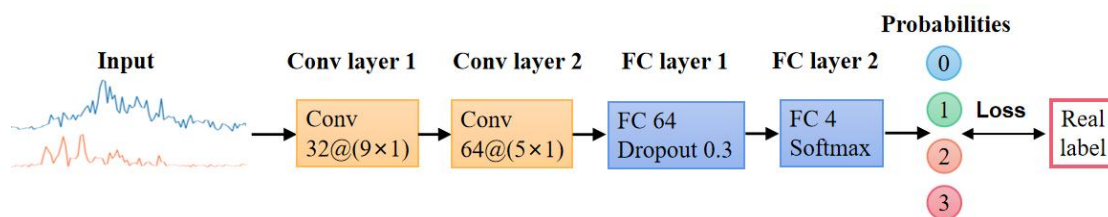


Figure 4.4: Facial expressions classification model.

4.2.5 Sign Language Translation Models

In the sign language dataset, the continuous signal stream for each sentence lasts for around 3-10 seconds. With the sliding window method, the long signal stream is segmented into a sequence of frames. Since the sampling rate of the motion capture system is 60Hz, the window size we set is 600ms (36 sample points) and the sliding size is 300ms (18 sample points).

The label for collected EMG & IMU data is the corresponding text sentence. There are 40 sentences in the dataset consisting of words and punctuation. We build a vocabulary at the word level and use the index of the word as the label. The vocabulary is shown in Table 2. Three kinds of special words are added to vocabulary: <BOS>, <EOS>, and <PAD> (indicating “begin of sentence”, “end of sentence”, and “padding”). We add <BOS> and <EOS> to the beginning and end of each sentence in the dataset and then pad the sentence to the same length with <PAD>. Finally, text sentences are changed into sequences of words’ indices.

Table 4.2: The vocabulary for 40 ASL sentences.

!	,	.	?	Christmas	I	I'm	Sunday
a	afraid	alright	also	and	angry	annoyed	are
back	beach	broke	busy	cat	church	college	come
commercial	cooking	cookies	day	deaf	delicious	dessert	did
dislikes	do	does	don't	eating	enjoy	feel	fine
finish	food	friend	fries	funny	go	growing	haha
hamburgers	happy	hate	have	help	him	his	history
home	hungry	ice-cream	is	it	know	like	merry
milk	mother's	my	need	new	of	on	running
sad	spider	steak	strawberry	student	studying	summer	thanksgiving
the	they	this	to	today	up	vegetable	want
we	where	why	wife	with	worried	wow	wrestling
year	you	<BOS>	<EOS>	<PAD>			

The first model is based on LSTM. As illustrated in Figure 4.5, the first layer of the encoder is CNN. The CNN layer extracts superior representations of features from input data frames as introduced in section 4.2.4. The input signal of stacked IMU and EMG has 40 channels, so the convolutional kernels we use here are 2-D kernels with the shape of 3×3 .

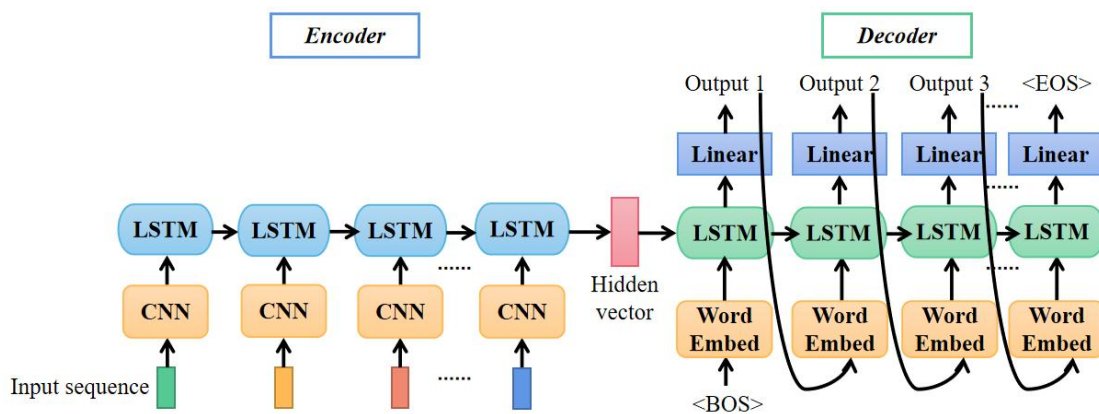


Figure 4.5: Architecture of LSTM-based translation model.

The second layer of the encoder is LSTM. Given the special word <BOS>, the decoder starts to output predicting results step by step. If the output of a time step turns to <EOS>, the whole predicting procedure should be finished.

The transformer model has been used successfully in a variety of tasks including reading comprehension, textual entailment, and learning task independent sentence representations [73]. With the self-attention mechanism, the model can draw global dependencies between input and output without considering the distance. The architecture of the transformer-based translation model is shown in Figure 4.6A.

In the encoder section, the input to the self-attention layer consists of two parts: features sequence extracted from the CNN layer and positional encoding recording the sequence order. The detailed structure of the self-attention layer is shown in Figure 4.6B. Query, key, and value all come from the same input by performing different linear transformations:

$$Q = W_Q \cdot x$$

$$K = W_K \cdot x$$

$$V = W_V \cdot x$$

The attention score is calculated as:

$$scores = softmax\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right)$$

where d_k is the dimension of K . The output of the self-attention layer is matrix multiplication between score and value matrix V :

$$Attention(Q, K, V) = scores \cdot V$$

After going through layer normalization and feed-forward module, the input is finally encoded into a hidden vector.

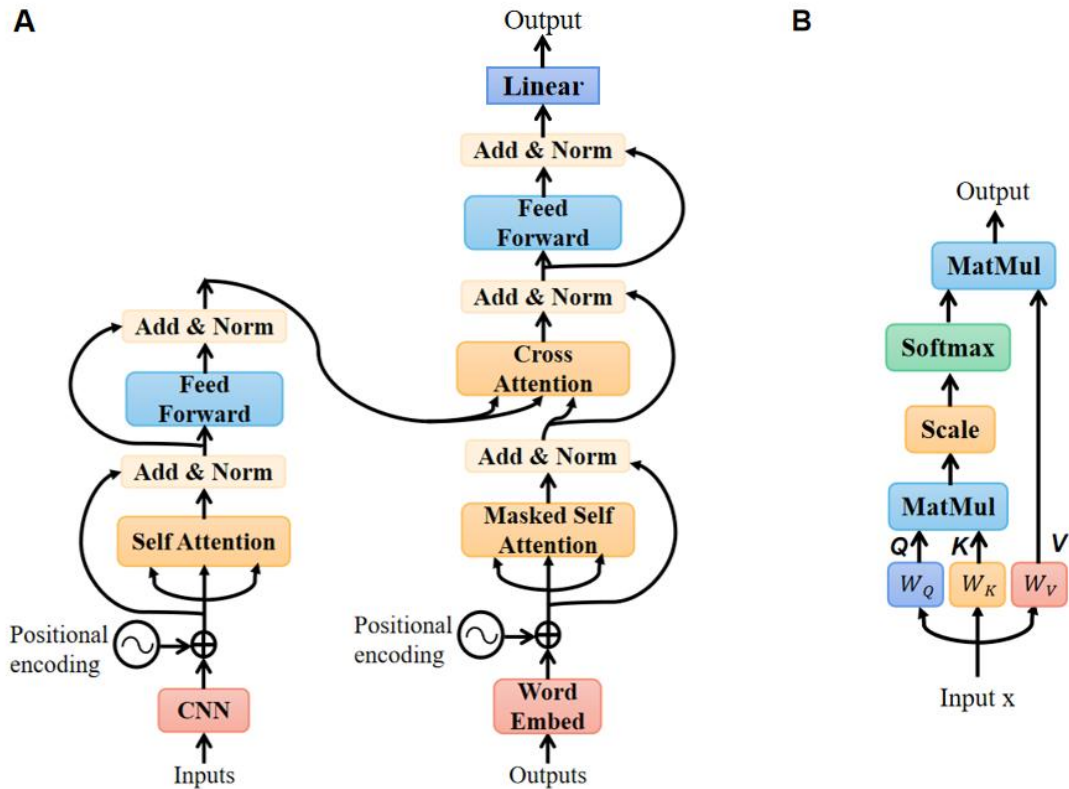


Figure 4.6: Transformer-based translation model: (A) Architecture of the model; (B) Detailed structure of the self-attention layer.

In the model’s training step, the input of the decoder is a text sentence. In the masked self-attention layer, the model can only attend to the output words that have been predicted before. The encoder-decoder cross attention layer includes K and V from encoder output and Q from decoder input. The calculation method is the same as self-attention. The output of the decoder is the probabilities of all possible words in the vocabulary. With the greedy search decoding method, we choose the word with the largest probability as the model prediction.

4.3 Results

4.3.1 Facial Expressions Classification

We validate the CNN classifier with five-fold cross-validation. The dataset

of EMG signals containing 2400 samples is randomly divided into five subsets. We leave each subset as the validation set and train the model with the remaining four subsets. This process is repeated five times. The loss function of the model is cross-entropy loss and the optimizer is Adam with a learning rate of 0.001. According to the recognition results of validation sets, the classification accuracy is calculated as given below:

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of validation samples}}$$

After training the model, the classification results of all cross-validation sets are shown in Figure 4.7A. The accuracy of more than 99% illustrates that EMG features are significantly different in four kinds of facial expressions. The confusion matrix accumulated from all cross-validation steps is shown in Figure 4.7B.

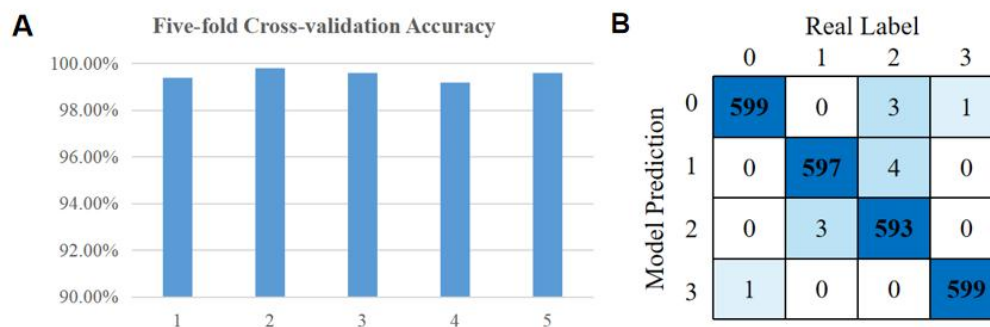


Figure 4.7: Facial expressions classification results: (A) Accuracy of five cross-validation sets; (B) Total confusion matrix of cross-validation steps.

4.3.2 Sign Language Translation

There are three participants contributing in data collection. We first validate the translation results on Participant 1. We randomly divide 1600 samples from Participant 1 into training set (70%, 1120 samples), validation set (15%, 240 samples) and testing set (15%, 240 samples). We use data in the training set to train the model and then adjust parameters with validation set to select

the model with best performance. The training loss is cross entropy between model predictions and real words in the label sentence. The optimizer is Adam with learning rate of 0.0003.

On the testing set, we employ Word Error Rate (WER) and Sentence Error Rate (SER) as the evaluation of the model. WER measures the least operations of substitution, deletion, and insertion to transform the predicted sentence into the ground truth sentence:

$$WER = \frac{N_{sub} + N_{del} + N_{ins}}{N_{ground\ truth\ words}}$$

where N_{sub} , N_{del} , and N_{ins} are numbers of required substitutions, deletions, and insertions, respectively. SER measures the percentage of not completely correct sentences of the model's testing prediction results:

$$SER = \frac{N_{error\ sentences}}{N_{ground\ truth\ sentences}}$$

In training step of LSTM based translation model, the losses of training and validation sets both drop dramatically in first few epochs, as illustrated in Figure 4.8. After 15 epochs' training, the model tends to converge with loss of nearly 0. We stop training the model at epoch 20 and evaluate it with testing set.

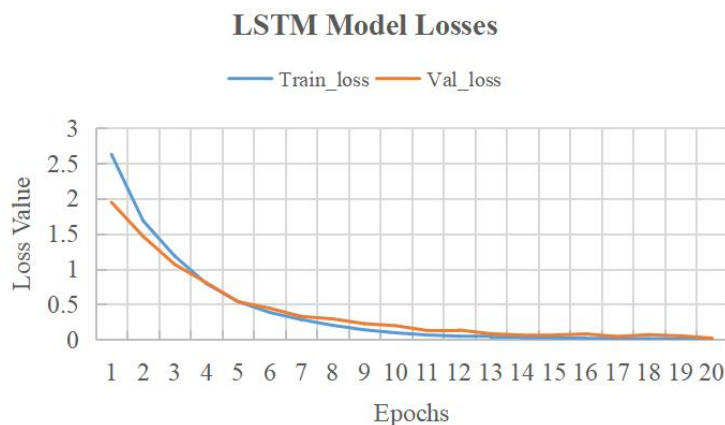


Figure 4.8: Training and validation losses of LSTM model for Participant 1.

Figure 4.9 shows the evaluation result of LSTM based translation model

on testing dataset. The blue bars are the sentences amount distribution of 40 sign language sentences in testing set, and the orange bars show the error sentences amount. Most sentences are predicted correctly by the model. The SER we calculated is 3.333% (8 error sentences of 240 samples) and the WER is 2.595% (14 del errors and 18 sub errors of 1233 words). The sentence No. 32 (“I’m busy.”) gives the worst prediction performance at 6 incorrect predictions of 8 samples.

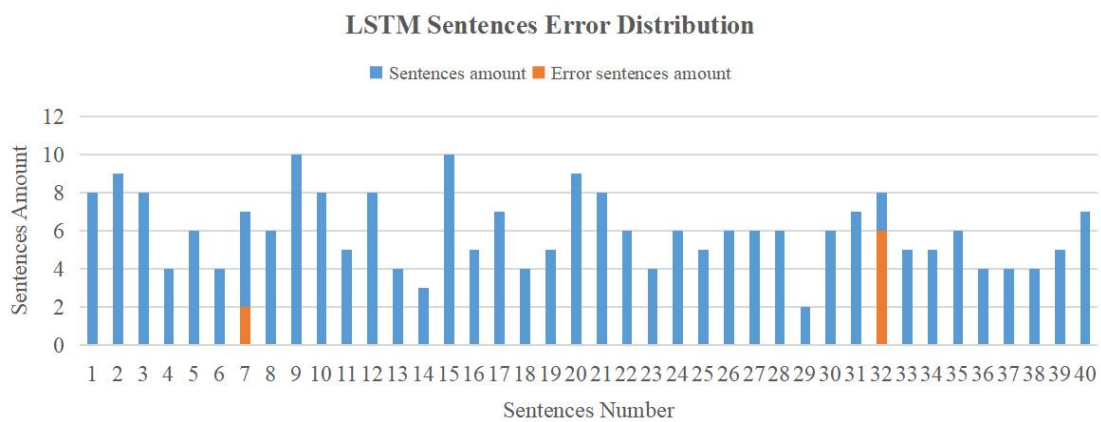


Figure 4.9: LSTM model evaluation result for Participant 1.

Transformer based translation model converges much faster, so we train the model for only 15 epochs. The losses of training and validation steps are shown in Figure 4.10, and the evaluation result is shown in Figure 4.11. It is clear that this model performs much better than LSTM model in testing dataset. There are only 2 error sentences from 240 sentences in the dataset, and thus the SER is 0.833%. The WER is calculated to be 0.649% (4 del errors, 3 sub errors and 1 ins error of 1233 words).



Figure 4.10: Training and validation losses of transformer model for Participant 1.

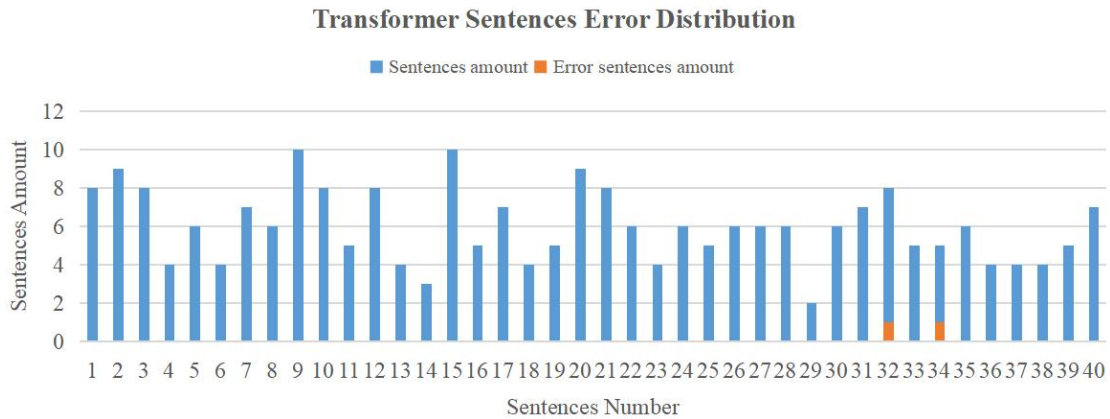


Figure 4.11: Transformer model evaluation result for Participant 1.

The translation result using only data from Participant 1 shows a high accuracy. For the full dataset, we randomly divide 2400 samples in the dataset into a training set (70%, 1680 samples), validation set (15%, 360 samples), and testing set (15%, 360 samples). We use data in the training set to train the model and then adjust parameters with the validation set to select the model with the best performance. The training loss is cross-entropy between model prediction and real labeled sentences. The optimizer is Adam with a learning rate of 0.0003.

In the training step of the LSTM translation model, the losses of training and validation set both drop dramatically in the first few epochs. After 15 epochs of training, the model tends to converge with a loss of nearly 0. We

stop training the model at epoch 20 and evaluate it with the testing set. Figure 4.12 shows the evaluation result of the LSTM translation model on the testing dataset. The blue bars are the sentence amount distribution of 40 sign language sentences in the testing set and the orange bars show the error sentences amount. Most sentences are predicted correctly by the model. The SER we calculated is 9.17% (33 error sentences of 360 samples) and the WER is 7.74% (43 del errors, 17 ins errors, and 87 sub errors of 1898 words).

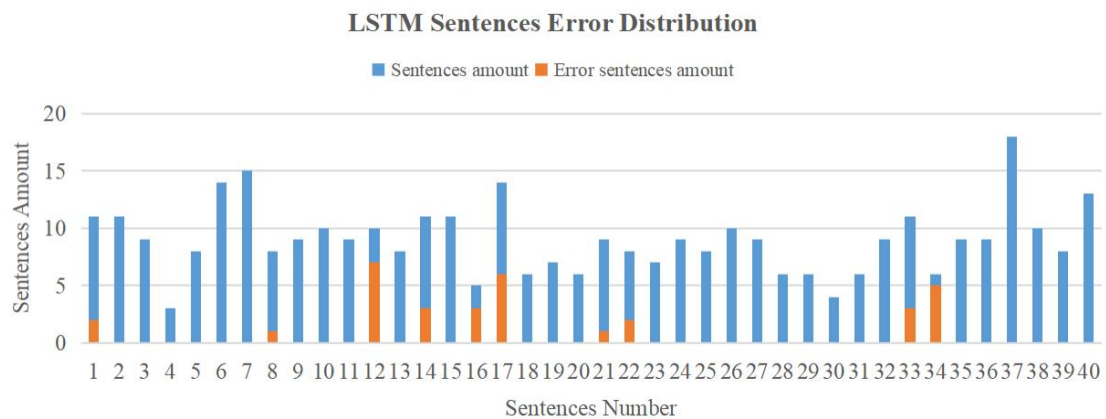


Figure 4.12: LSTM model evaluation result.

The transformer translation model converges much faster, so we train the model for only 15 epochs. The evaluation result is shown in Figure 4.13. This model performs much better than the LSTM model in the testing dataset. There are only 17 error sentences from 360 sentences in the dataset and thus the SER is 4.72%. The WER is calculated to be 4.21% (33 del errors and 47 sub errors of 1898 words).

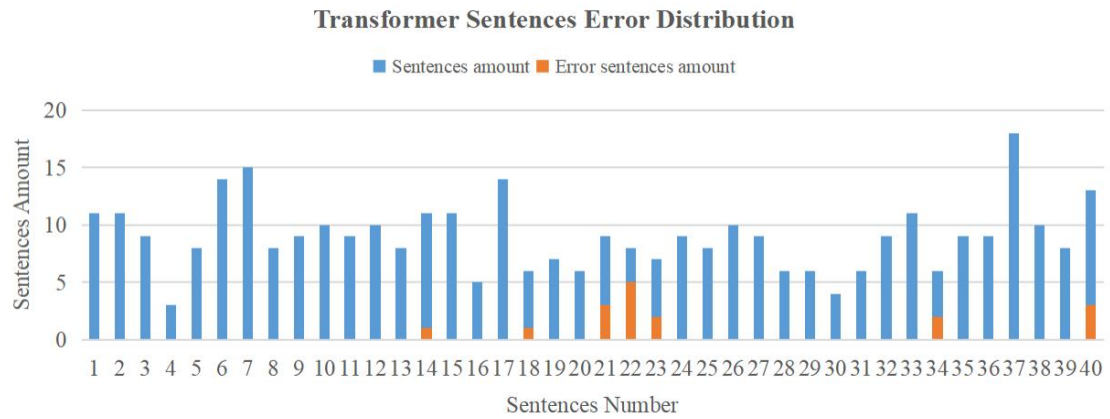


Figure 4.13: Transformer model evaluation result.

4.4 Discussion

4.4.1 Significance of EMG

In this work, EMG signals from facial areas provide four kinds of emotional information during sign language performance. Combining EMG and IMU data as input provides the model with more information to achieve better prediction results. To evaluate the significance of EMG, we remove the EMG data from the input and then train the translation models again with only IMU data.

We first evaluate the significance of EMG using the data from Participant 1. The comparisons between input with or without EMG are shown in Table 4.3 and Table 4.4. WER of two kinds of models increases by 2.190% and 2.433% without EMG data as input, and SER also increases by 3.334% and 2.500% respectively. The detailed sentences error results predicted by two models are shown in Figure 4.14 and Figure 4.15. Both models give more wrong predictions, but the transformer based model still performs much better than LSTM model at 1.703% lower error rate in word level and 3.334% lower error rate in sentence level.

Table 4.3: Word error rate comparison for Participant 1.

	LSTM	Transformer
Input with EMG	2.60%	0.65%
Input without EMG	4.79%	3.08%

Table 4.4: Sentence error rate comparison for Participant 1.

	LSTM	Transformer
Input with EMG	3.33%	0.83%
Input without EMG	6.67%	3.33%

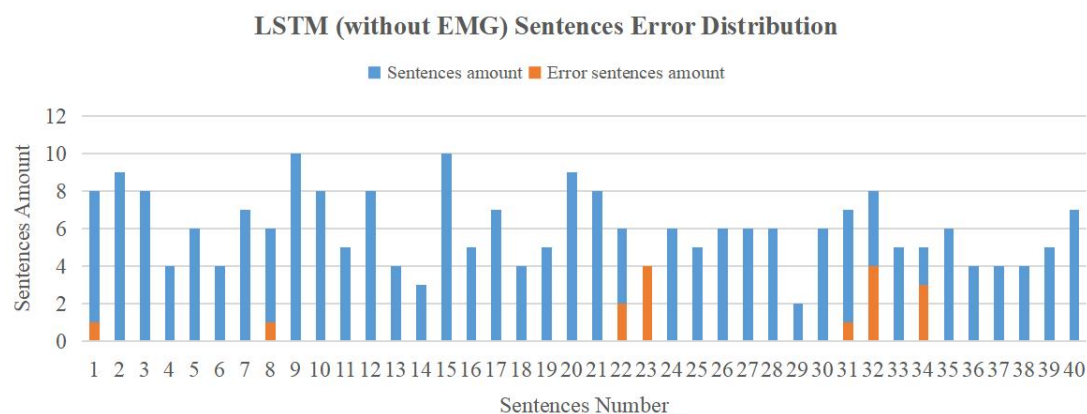


Figure 4.14: LSTM model evaluation without EMG as input for Participant 1.

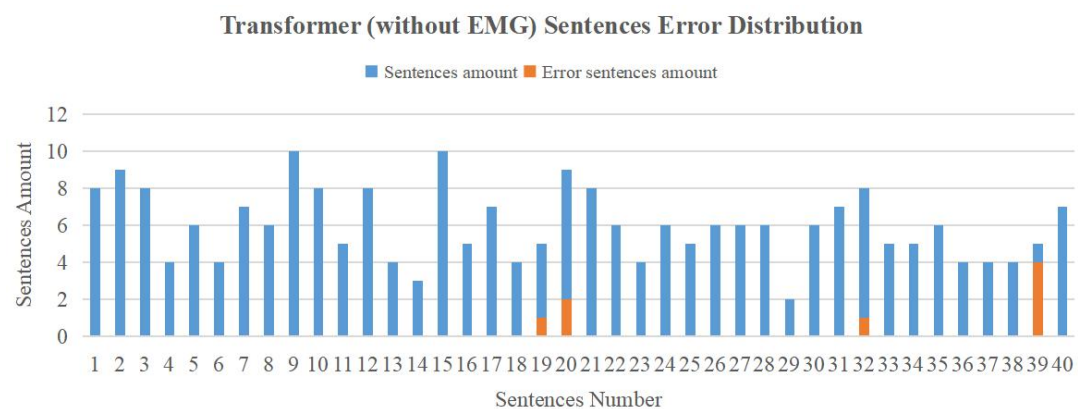


Figure 4.15: Transformer model evaluation without EMG as input for Participant 1.

The significance of EMG using the full dataset is then evaluated. The

comparisons between input with or without EMG are shown in Table 4.5 and Table 4.6. WERs of the two models increase by 4.12% and 4.21% without EMG data as input, and SERs also increase by 5.55% and 4.17%, respectively. Both models give more wrong predictions, but the transformer model still performs much better than the LSTM model at a 3.43% lower error rate at the word level and 5.83% lower error rate at the sentence level.

Table 4.5: Word error rate comparison.

	LSTM	Transformer
Input with EMG	7.74%	4.22%
Input without EMG	11.86%	8.43%

Table 4.6: Sentence error rate comparison.

	LSTM	Transformer
Input with EMG	9.17%	4.72%
Input without EMG	14.72%	8.89%

4.4.2 User-independent Validation

We evaluate the performance of models in user-independent conditions. Three participants participated in this experiment. Participant 1 who contributed the largest amount of data (1600 samples) is always used as a part of the training set. Participants 2 (400 samples) and 3 (400 samples) are regarded as testing sets respectively. The results are shown in Table 5. In the sign language translation task, both WER and SER increase dramatically to more than 40%. Due to different habits and amplitudes of each person’s sign language performances, there are great differences between the movement data in user-independent validation. The method we proposed can still

translate more than half of the sentences in the testing set accurately. In the user-independent validation of facial expression classification with EMG, the accuracy remains at a high level of more than 93%. The result illustrates that the EMG signals of four different expressions have distinguishable features.

Table 4.7: User-independent validation results.

	WER	SER	Facial expression classification accuracy
Participant 2	41.95%	44.50%	93.25%
Participant 3	41.12%	46.00%	95.00%

4.4.3 Limitations

The dataset contains limited sentences and participants. Only four kinds of facial expressions are considered, as a result, the CNN classifier gives high-accurate results on this four-category classification task. LSTM and transformer are two commonly used models in NLP research. Instead of text or speech, the input of sign language is signals from the human body. The transformer model outperforms the LSTM model. The transformer is originally proposed to solve the sequential order problem of RNN. The LSTM model can only read input from left to right or from right to left, but the transformer considers the overall input content at the same time. With EMG as a part of the input, the accuracy of the model prediction improves. EMG can enhance the model’s translation ability. In user-independent validation, the translation accuracy drops dramatically due to the significant inter-individual differences in movement. More participants should be involved in the experiment and the model should learn knowledge from more data.

Compared with visual methods of sign language translation, a camera is more portable but will encounter background and perspective problems. Even the most popular Kinect camera with skeleton tracking function cannot extract the detailed skeleton structure of hands. To some extent, wearable IMU sensors are more reliable. The IMU-based motion capture device for the upper body contains 25 sensors. It is a unitary device and cannot be disassembled. This motion capture system is bulky for a translation system with only 40 sentences, but it has the potential to recognize more sentences. A larger dataset using this device is in preparation and machine learning algorithms more suitable for wearables are being developed.

4.5 A Larger Dataset

A larger ASL dataset is also collected without considering user independence. This dataset contains 300 commonly used ASL sentences from online sign language lectures. One participant did 40 repetitions for each sentence. There are 300×40 samples in the dataset. The vocabulary contains totally 526 words. The model we use here is a bi-directional LSTM encoder and LSTM decoder, as shown in Figure 4.16. We randomly choose 70% of data to train the model and use 15% data to adjust parameters. The remaining 15% data is the testing set. The sentence error rate of testing dataset is calculated to be 25.8%. The model can translate the sentence correctly with accuracy of 74.8%.

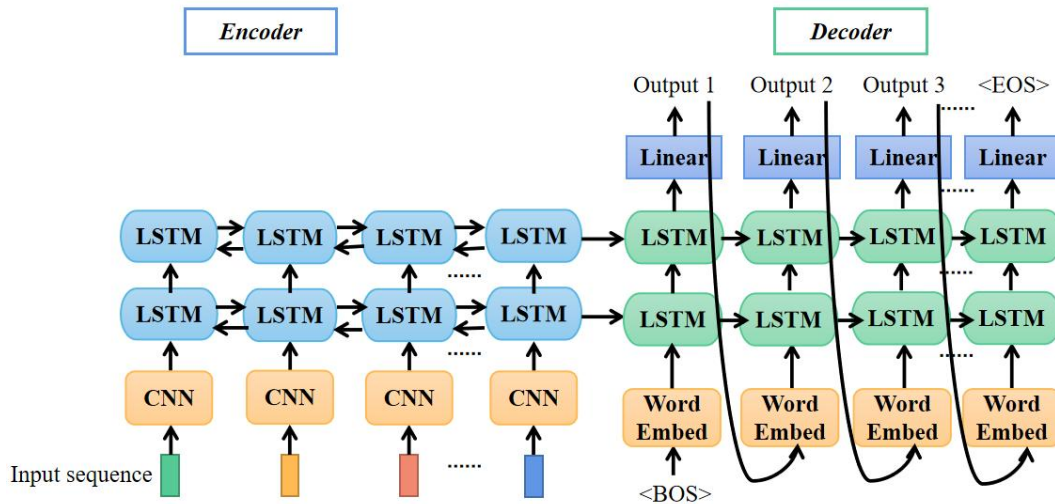


Figure 4.16: Translation model for 300 ASL sentences.

4.6 Conclusions

In this Chapter, we presented a wearable sensors-based sign language translation method considering both hands' movements and facial expressions. IMU and EMG signals were preprocessed and segmented into a sequence of frames as the input of translation models. We classified facial expressions with EMG data only. Then we built encoder-decoder models to realize end-to-end sign language translation from signals to text sentences. Two kinds of end-to-end models based on LSTM and transformer were trained and evaluated by the collected dataset. WER and SER were used to compare the translation ability of models. Both models could translate 40 ASL sentences with high accuracy and the transformer-based model performed better than LSTM. The special role of EMG was verified with both facial expressions classification and models' performance after removing EMG from the input. The translation accuracy in user-independent conditions was evaluated

Chapter 5

Conclusions and Future Work

Sign language is the main communication method among hearing-impaired people. As a kind of natural language, sign language has not become a mainstream research topic in natural language processing, although the machine translation of spoken or written language is highly accurate today. However, the research of machine translation with deep learning models provides development direction and innovative methods for sign language translation tasks. In order to further the research on end-to-end translation, it is necessary to consider the application of deep learning models. Previous works about sign language translation mainly falls to two categories: vision-based and wearable sensors-based. Vision-based methods exploit camera to capture features of hands. In wearable sensors-based research, devices like data glove, wristwatch or armband are the mainstream for data collection. In this work, we explored the sign language translation using wearable sensors.

This study proposed the complete research process of sign language translation technology. The research was started from isolated gesture recognition and finally went to the end-to-end translation of full sentences using wearable sensors. The facial expressions in sign language performances were also collected by EMG device. The combination of

natural language processing and wearable sensors provides a new idea for sign language translation task. The datasets we collected will make it easier for more people to start research on sign language translation and machine learning. The works in this study are significant new and may contribute a huge impact for researchers in this field.

In isolated hand gestures recognition, the result is improved by feature selection. The difference between non-adjacent joints is a special feature that is seldom used in other studies, but it provides the best result in this research. With multiple features as input, if the weights are added for features, the result could be much better. There are some studies using EMG signals to detect facial expression, but this method has never been used in the sign language. So, it is creative to add facial expressions into sign language research using wearable sensors.

There are some limitations for this research. The biggest problem is the lack of participants in the data collection experiments. With limited participants, the model could only learn a limited number of patterns from hand gestures. That's why the results of user independent validation are always dropped. If the model could learn more patterns from different people, for each individual, there should be some similar information in the training set from other people. As for the EMG signal, studies using it to do facial expressions classification always contain only limited number of classes. Also, no improvement is made in this study.

In the future work, a larger dataset with more sentences is in preparation. The 300 ASL sentences dataset we mentioned is a part of it. The suitable machine learning model will be built for this customized dataset and more advanced algorithms will be considered. Since the dataset is more complex, we may also need some special methods like pretraining the model using related available data. Vision-based sign language translation is also a popular topic in this research area. Research on visual methods will be

launched soon in the near future. Then, the comparison could be made between wearable sensors-based method and vision-based method.

References

- [1] World Federation of the Deaf. Available online: <http://wfdeaf.org/our-work/> (accessed on 15 April 2022).
- [2] CyberGlove data glove. <http://www.cyberglovesystems.com/cyberglove-iii/>
- [3] Takahashi, T., & Kishino, F. (1991). Hand gesture coding based on experiments using a hand gesture interface device. *Acm Sigchi Bulletin*, 23(2), 67-74.
- [4] Bergerman, M., Lee, C., & Xu, Y. (1995, September). Dynamic coupling of underactuated manipulators. In *Proceedings of International Conference on Control Applications* (pp. 500-505). IEEE.
- [5] Ibarguren, A., Maurtua, I., & Sierra, B. (2010). Layered architecture for real time sign recognition: Hand gesture and movement. *Engineering Applications of Artificial Intelligence*, 23(7), 1216-1228.
- [6] Oz, C., & Leu, M. C. (2011). American sign language word recognition with a sensory glove using artificial neural networks. *Engineering Applications of Artificial Intelligence*, 24(7), 1204-1213.
- [7] Kadous, M. W. (1996, October). Machine recognition of Auslan signs using PowerGloves: Towards large-lexicon recognition of sign language. In *Proceedings of the Workshop on the Integration of Gesture in Language and Speech* (Vol. 165, pp. 165-174). Wilmington: DE.
- [8] Hernandez-Rebollar, J. L., Lindeman, R. W., & Kyriakopoulos, N. (2002, October). A multi-class pattern recognition system for practical finger spelling translation. In *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces* (pp. 185-190). IEEE.
- [9] Tatarian, K., Couceiro, M. S., Ribeiro, E. P., & Faria, D. R. (2018, September). Stepping-stones to transhumanism: An emg-controlled low-cost prosthetic hand for academia. In *2018 International Conference on Intelligent Systems (IS)* (pp. 807-812). IEEE.

- [10] Wu, J., Sun, L., & Jafari, R. (2016). A wearable system for recognizing American sign language in real-time using IMU and surface EMG sensors. *IEEE journal of biomedical and health informatics*, 20(5), 1281-1290.
- [11] Zhang, Q., Wang, D., Zhao, R., & Yu, Y. (2019, March). MyoSign: enabling end-to-end sign language recognition with wearables. In *Proceedings of the 24th international conference on intelligent user interfaces* (pp. 650-660).
- [12] Tateno, S., Liu, H., & Ou, J. (2020). Development of sign language motion recognition system for hearing-impaired people using electromyography signal. *Sensors*, 20(20), 5807.
- [13] Huang, J., Zhou, W., Li, H., & Li, W. (2015, July). Sign language recognition using real-sense. In *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)* (pp. 166-170). IEEE.
- [14] Mistry, J., & Inden, B. (2018, September). An approach to sign language translation using the intel realsense camera. In *2018 10th Computer Science and Electronic Engineering (CEECE)* (pp. 219-224). IEEE.
- [15] Liao, B., Li, J., Ju, Z., & Ouyang, G. (2018, June). Hand gesture recognition with generalized hough transform and DC-CNN using realsense. In *2018 Eighth International Conference on Information Science and Technology (ICIST)* (pp. 84-90). IEEE.
- [16] Mohandes, M., Aliyu, S., & Deriche, M. (2014, June). Arabic sign language recognition using the leap motion controller. In *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)* (pp. 960-965). IEEE.
- [17] Lee, G. C., Yeh, F. H., & Hsiao, Y. H. (2016). Kinect-based Taiwanese sign-language recognition system. *Multimedia Tools and Applications*, 75(1), 261-279.
- [18] Zhang, L. G., Chen, Y., Fang, G., Chen, X., & Gao, W. (2004, October).

A vision-based sign language recognition system using tied-mixture density HMM. In Proceedings of the 6th international conference on Multimodal interfaces (pp. 198-204).

[19] Koller, O., Forster, J., & Ney, H. (2015). Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141, 108-125.

[20] Sun, C., Zhang, T., & Xu, C. (2015). Latent support vector machine modeling for sign language recognition with Kinect. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(2), 1-20.

[21] Fang, B., Co, J., & Zhang, M. (2017, November). Deepasl: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation. In Proceedings of the 15th ACM conference on embedded network sensor systems (pp. 1-13).

[22] Pu, J., Zhou, W., & Li, H. (2019). Iterative alignment network for continuous sign language recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4165-4174).

[23] Camgoz, N. C., Koller, O., Hadfield, S., & Bowden, R. (2020). Sign language transformers: Joint end-to-end sign language recognition and translation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10023-10033).

[24] Huang, J., Zhou, W., Zhang, Q., Li, H., & Li, W. (2018, April). Video-based sign language recognition without temporal segmentation. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).

[25] Zhou, H., Zhou, W., & Li, H. (2019, July). Dynamic pseudo label decoding for continuous sign language recognition. In 2019 IEEE International Conference on Multimedia and Expo (ICME) (pp. 1282-1287). IEEE.

- [26] Pu, J., Zhou, W., Hu, H., & Li, H. (2020, October). Boosting continuous sign language recognition via cross modality augmentation. In Proceedings of the 28th ACM International Conference on Multimedia (pp. 1497-1505).
- [27] Kim, J. S., Jang, W., & Bien, Z. (1996). A dynamic gesture recognition system for the Korean sign language (KSL). *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(2), 354-359.
- [28] Holden, E. J., & Owens, R. (2001). Visual sign language recognition. In *Multi-Image Analysis* (pp. 270-287). Springer, Berlin, Heidelberg.
- [29] Martínez, A. M., Wilbur, R. B., Shay, R., & Kak, A. C. (2002, October). Purdue RVL-SLLL ASL database for automatic recognition of American Sign Language. In Proceedings. Fourth IEEE International Conference on Multimodal Interfaces (pp. 167-172). IEEE.
- [30] Dreuw, P., Rybach, D., Deselaers, T., Zahedi, M., & Ney, H. (2007). Speech recognition techniques for a sign language recognition system. *hand*, 60, 80.
- [31] Von Agris, U., Knorr, M., & Kraiss, K. F. (2008, September). The significance of facial features for automatic sign language recognition. In 2008 8th IEEE International Conference on Automatic Face & Gesture Recognition (pp. 1-6). IEEE.
- [32] Cooper, H. M., Ong, E. J., Pugeault, N., & Bowden, R. (2012). Sign language recognition using sub-units. *Journal of Machine Learning Research*, 13, 2205-2231.
- [33] Neidle, C., Thangali, A., & Sclaroff, S. (2012). Challenges in development of the american sign language lexicon video dataset (asllvd) corpus. In 5th workshop on the representation and processing of sign languages: interactions between corpus and Lexicon, LREC.
- [34] Kapuscinski, T., Oszust, M., Wysocki, M., & Warchol, D. (2015). Recognition of hand gestures observed by depth cameras. *International*

- Journal of Advanced Robotic Systems, 12(4), 36.
- [35] Chai, X., Wang, H., & Chen, X. (2014). The devisign large vocabulary of chinese sign language database and baseline evaluations. In Technical report VIPL-TR-14-SLR-001. Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS). Institute of Computing Technology.
- [36] Joze, H. R. V., & Koller, O. (2018). Ms-asl: A large-scale data set and benchmark for understanding american sign language. arXiv preprint arXiv:1812.01053.
- [37] Li, D., Rodriguez, C., Yu, X., & Li, H. (2020). Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In Proceedings of the IEEE/CVF winter conference on applications of computer vision (pp. 1459-1469).
- [38] Yuan, T., Sah, S., Ananthanarayana, T., Zhang, C., Bhat, A., Gandhi, S., & Ptucha, R. (2019, May). Large scale sign language interpretation. In 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019) (pp. 1-5). IEEE.
- [39] Sarzynska-Wawer, J., Wawer, A., Pawlak, A., Szymanowska, J., Stefaniak, I., Jarkiewicz, M., & Okruszek, L. (2021). Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, 304, 114135.
- [40] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
- [41] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.
- [42] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 9.

- [43] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877-1901.
- [44] Hanke, T. (2004, May). HamNoSys-representing sign language data in language resources and language processing contexts. In *LREC* (Vol. 4, pp. 1-6).
- [45] Elliott, R., Glauert, J. R., Kennaway, J. R., & Marshall, I. (2000, November). The development of language processing support for the ViSiCAST project. In *Proceedings of the fourth international ACM conference on Assistive technologies* (pp. 101-108).
- [46] Davydov, M., & Lozynska, O. (2017, September). Information system for translation into Ukrainian sign language on mobile devices. In *2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT)* (Vol. 1, pp. 48-51). IEEE.
- [47] Zhao, L., Kipper, K., Schuler, W., Vogler, C., Badler, N., & Palmer, M. (2000, October). A machine translation system from English to American Sign Language. In *Conference of the Association for Machine Translation in the Americas* (pp. 54-67). Springer, Berlin, Heidelberg.
- [48] Pavlovic, V. I., Sharma, R., & Huang, T. S. (1997). Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7), 677-695.
- [49] Tennant, R. A., Gluszak, M., & Brown, M. G. (1998). *The American sign language handshape dictionary*. Gallaudet University Press.
- [50] Atzori, M., Gijsberts, A., Kuzborskij, I., Heynen, S., Hager, A. G. M., Deriaz, O., ... & Caputo, B. A Benchmark Database for Myoelectric Movement Classification.
- [51] Atzori, M., Gijsberts, A., Castellini, C., Caputo, B., Hager, A. G. M., Elsig, S., ... & Müller, H. (2014). Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Scientific data*, 1(1), 1-13.

- [52] Pizzolato, S., Tagliapietra, L., Cognolato, M., Reggiani, M., Müller, H., & Atzori, M. (2017). Comparison of six electromyography acquisition setups on hand movement classification tasks. *PloS one*, 12(10), e0186132.
- [53] Du, Y., Jin, W., Wei, W., Hu, Y., & Geng, W. (2017). Surface EMG-based inter-session gesture recognition enhanced by deep domain adaptation. *Sensors*, 17(3), 458.
- [54] Daroya, R., Peralta, D., & Naval, P. (2018, October). Alphabet sign language image classification using deep learning. In *TENCON 2018-2018 IEEE Region 10 Conference* (pp. 0646-0650). IEEE.
- [55] Ranga, V., Yadav, N., & Garg, P. (2018). American sign language fingerspelling using hybrid discrete wavelet transform-gabor filter and convolutional neural network. *Journal of Engineering Science and Technology*, 13(9), 2655-2669.
- [56] Jalal, M. A., Chen, R., Moore, R. K., & Mihaylova, L. (2018, July). American sign language posture understanding with deep neural networks. In *2018 21st International Conference on Information Fusion (FUSION)* (pp. 573-579). IEEE.
- [57] Paudyal, P., Lee, J., Banerjee, A., & Gupta, S. K. (2019). A comparison of techniques for sign language alphabet recognition using armband wearables. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 9(2-3), 1-26.
- [58] Shi, B., Brentari, D., Shakhnarovich, G., & Livescu, K. (2021). Fingerspelling Detection in American Sign Language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4166-4175).
- [59] Shi, B., Del Rio, A. M., Keane, J., Michaux, J., Brentari, D., Shakhnarovich, G., & Livescu, K. (2018, December). American sign language fingerspelling recognition in the wild. In *2018 IEEE Spoken Language Technology Workshop (SLT)* (pp. 145-152). IEEE.

- [60] Hernandez, V., Suzuki, T., & Venture, G. (2020). Convolutional and recurrent neural network for human activity recognition: Application on American sign language. *PloS one*, 15(2), e0228869.
- [61] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- [62] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- [63] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929-1958.
- [64] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
- [65] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456). PMLR.
- [66] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- [67] Shin, J., Matsuoka, A., Hasan, M. A. M., & Srizon, A. Y. (2021). American sign language alphabet recognition by extracting feature from hand pose estimation. *Sensors*, 21(17), 5856.
- [68] Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006, June). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international*

conference on Machine learning (pp. 369-376).

[69] Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28.

[70] De Luca, C. J., Gilmore, L. D., Kuznetsov, M., & Roy, S. H. (2010). Filtering the surface EMG signal: Movement artifact and baseline noise contamination. *Journal of biomechanics*, 43(8), 1573-1579.

[71] Phinyomark, A., Limsakul, C., & Phukpattaranont, P. (2009, April). EMG feature extraction for tolerance of 50 Hz interference. In *Proc. of PSU-UNS Inter. Conf. on Engineering Technologies, ICET* (pp. 289-293).

[72] Yang, L., Wang, W., & Zhang, Q. (2016, November). Secret from muscle: Enabling secure pairing with electromyography. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM* (pp. 28-41).

[73] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.