inventions

MDPI

Article

American Sign Language Alphabet Recognition Using Inertial Motion Capture System with Deep Learning

Yutong Gu [1,*], Sherrine [1], Weiyi Wei [1], Xinya Li [2], Jianan Yuan [3] and Masahiro Todoh [4]

[1] Graduate School of Engineering, Hokkaido University, Sapporo 0608628, Japan
[2] Graduate School of Economics and Business, Hokkaido University, Sapporo 0600809, Japan
[3] Graduate School of Environmental Science, Hokkaido University, Sapporo 0600810, Japan
[4] Faculty of Engineering, Hokkaido University, Sapporo 0608628, Japan
* Correspondence: gu_yutong@frontier.hokudai.ac.jp

Abstract: Sign language is designed as a natural communication method for the deaf community to convey messages and connect with society. In American sign language, twenty-six special sign gestures from the alphabet are used for the fingerspelling of proper words. The purpose of this research is to classify the hand gestures in the alphabet and recognize a sequence of gestures in the fingerspelling using an inertial hand motion capture system. In this work, time and time-frequency domain features and angle-based features are extracted from the raw data for classification with convolutional neural network-based classifiers. In fingerspelling recognition, we explore two kinds of models: connectionist temporal classification and encoder-decoder structured sequence recognition model. The study reveals that the classification model achieves an average accuracy of 74.8% for dynamic ASL gestures considering user independence. Moreover, the proposed two sequence recognition models achieve 55.1%, 93.4% accuracy in word-level evaluation, and 86.5%, 97.9% in the letter-level evaluation of fingerspelling. The proposed method has the potential to recognize more hand gestures of sign language with highly reliable inertial data from the device.

Keywords: American sign language alphabet; hand gesture classification; sequence recognition

Citation: Gu, Y.; S.; Wei, W.; Li, X.; Yuan, J.; Todoh, M. American Sign Language Alphabet Recognition Using Inertial Motion Capture System with Deep Learning. *Inventions* **2022**, *7*, 112. https://doi.org/10.3390/inventions7040112

Academic Editors: Edwin Lughofer and Anastasios Doulamis

Received: 30 September 2022
Accepted: 25 November 2022
Published: 1 December 2022

1. Introduction

Sign language is widely used by hearing-impaired people to communicate with each other. In daily life, we can see sign language interpreters in the live news or weather forecast, but most people with normal hearing could hardly understand the meaning of their expressions. Meanwhile, it is also difficult to start a conversation with real-life sign language users. With sign language recognition (SLR), the communication barriers could be alleviated. In American sign language (ASL), twenty-six special hand gestures representing the letters in the alphabet (A–Z) are normally used to spell proper nouns like names, technical terms, and abbreviations or unfamiliar words [1], which accounts for 12% to 35% of ASL [2]. Figure 1 shows the alphabet of ASL constituting designated fingers and hand-shape gestures [3].

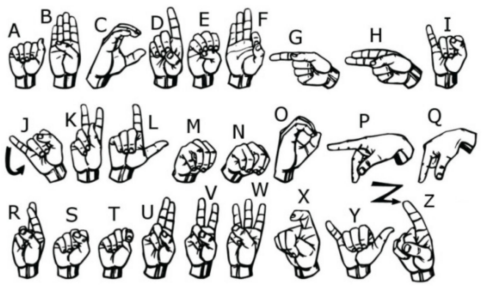


Figure 1. Twenty-six letters in ASL alphabet with different patterns of fingers and hand shapes.

The hand gesture recognition of this finite corpus with 26 possible classes has been conducted in prior work by the following two main classification mechanisms: vision-based and wearable sensors-based recognition. The vision-based approach utilizes RGB or RGB-D camera to catch the static gestures or dynamic movements of the hand. Most of the gestures can be regarded as static without involving any movement of the forearm except for the letters "j" and "z". Studies treating hand gestures as static always ignore these two letters to become 24 classification [4–6]. Jalal et al. [7] built a capsule-based Deep Neural Network (DNN) for the sign gestures recognition of the ASL Alphabet dataset [8] and achieved a relatively high classification accuracy of 99%. Ranga et al. [9] and Nguyen & Do [10] both did the classification on the Massey dataset [11]. Ranga et al. applied a hybrid discrete wavelet transform-Gabor filter for feature extraction from images. Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Convolutional Neural Network (CNN) models were evaluated, which produced the highest accuracy of 97.01% on signer dependent and 76.25% on signer independent evaluation. Nguyen & Do extracted Histogram of Oriented Gradients (HOG) and Local Binary Pattern (LBP) features from images and applied SVM and CNN architecture models to achieve the result of 98.36% without considering signer independence. With the fingerspelling A dataset [12], Rajan & Leo [13] applied the skin color-based YCbCr segmentation method to extract the hand shape. Besides, Shin et al. [14] estimated the coordinates of hand joints for classification and achieved 99.39%, 87.60%, and 98.45% on the above-mentioned three datasets, respectively. On dynamic gesture recognition, Thongtawee et al. [15] applied Webcam to collect 26 signs and achieved up to 95% recognition rate. Chong & Lee [16] used Leap Motion to recognize 26 letters and 10 digits. Dawod & Chakpitak [17] built a dataset of dynamic signs with Kinect and achieved high accuracy by Random Decision Forest (RDF) and Hidden Conditional Random Fields (HCRF) classifiers.

In wearable sensors-based recognition, commercial devices like the armband, smartwatch, and data glove provide convenient experimental applications. Due to the restricted arm movements of sign gestures in the alphabet, hand shapes are important distinguishing factors for different letters. Paudyal et al. [18] utilized the MYO armband to build a dataset from nine participants. The MYO armband returned Inertial Measurement Unit (IMU) signals indicating the forearm direction and electromyographic (EMG) signals indicating the hand shape. With the Dynamic Feature Selection and Voting (DyFAV) algorithm, the system with an independent multiple-agent voting approach could identify letters with high accuracy. Hou et al. [19] built a sign language translator based on the smartwatch. Instead of the ASL alphabet, hand gestures of 103 ASL words were collected due to the limited ability of the smartwatch in recognizing hand shapes. Saquib & Rahman [20] developed a system to detect the ASL alphabet and Bengali Sign Language (BdSL) alphabet with a data glove. The data glove returned highly reliable information on hand joints. Thus, the system was capable of accurately detecting both static and dynamic signs in the alphabet. A novel method for static hand gesture recognition is using the magnetic positioning system [21]. Additionally, some customized devices show better performance in gesture recognition. Lee et al. [22] customized a device with six IMU sensors to detect the orientation of the hand and fingers. Zhu et al. [23] presented a novel epidermal-intronic sensing (EIS) wearable device worn on finger joints for hand gesture recognition. Compared with introduced on-market devices, this device was lighter and more comfortable to wear.

In ASL fingerspelling, meaningful words are constructed by signing multiple letters in a sequence. Fingerspelling recognition is a challenging task with untrimmed sign language videos [24] because the boundaries of gestures in the sequence are relatively blurry. Shi et al. [25] built the first large dataset for the problem of finger spelling recognition with naturally occurring video data. With attention-based recurrent encoder-decoders and Connectionist Temporal Classification (CTC)-based approaches, the best recognition result was 42.8%. When using an end-to-end model with the iterative attention mechanism [26], the recognition accuracy finally reached 61.2%.

In general, both vision-based and wearable sensor-based approaches have their own merits and limitations. In this study, a wearable inertial motion capture system is utilized to collect a dynamic dataset of hand gestures in the ASL alphabet. Time and time-frequency domain features and angle-based features are extracted from the raw signals to promote classification accuracy. Cross-user classification results are evaluated to identify the general applicability of the method. Then, fifty commonly used English words are generated by the hand gesture data in the dataset. Two kinds of sequence recognition models are applied to the recognition of fingerspelling.

The rest of this paper is organized as follows: Section 2 introduces the experimental data collection device and signal preprocessing methods. The machine learning models for hand gesture classification and sequence recognition are also presented. Section 3 provides the recognition results by using the designed models. Then, the differences in easily confusing hand gestures are discussed. Finally, the conclusion is drawn for this research.

2. Materials and Methods

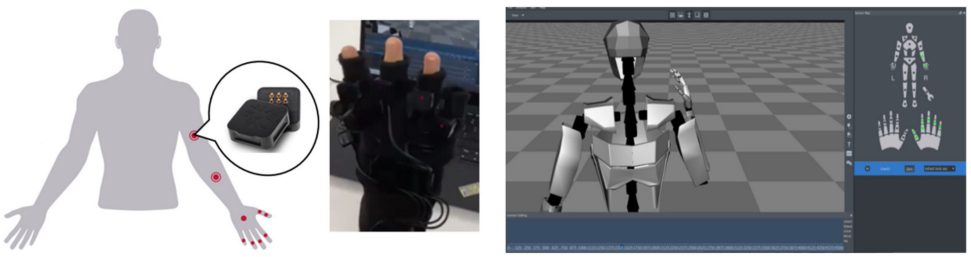
2.1. Isolated Hand Gestures Recognition of Twenty-Six Letters in the Alphabet

2.1.1. Dataset Collection

In the previous works of sign language recognition, the signs in the alphabet are regarded as static (24 signs without "j" and "z") or dynamic (all 26 signs) processes. In this study, all the signs are regarded as dynamic processes, because the next task conducted is the fingerspelling recognition, containing a sequence of dynamic gestures in the data. As shown in Figure 2, we set a rest state as the start and end of each sign. The sign starts from a rest state and finally returns to the original rest state.

Figure 2. Dynamic process of sign "j".

Perception Neuron (Noitom Ltd., Beijing, China) is a wearable IMU sensors-based motion capture system. Figure 3 shows the right-hand mode of the system with nine inertial sensors (named "Neuron") distributed on the right hand and arm. The red points show the positions of Neurons fixed by fabric and straps. Each Neuron is composed of an accelerometer, gyroscope, and magnetometer. Like all other IMU sensors, it can return the yaw, pitch, and roll of the attached position to detect the bone posture. The sampling rate is fixed at 120 Hz. Axis Neuron (Noitom Ltd., Beijing, China) is the official software of the device. It can receive and process the motion data and export files in *bvh* format.



(**a**) (**b**)

Figure 3. Perception Neuron motion capture system for right hand [27]: (**a**) distribution of sensors; (**b**) main interface of Axis Neuron software.

A *bvh* file contains the rotation information of all twenty right-hand joints, as illustrated in Figure 4. The coordinate values (Rotation_Y, Rotation_X, Rotation_Z) record the angles

rotated by the coordinate system under the movement compared with the initial state. Before the experiment, device calibration is conducted to determine the initial orientation of the coordinate system. The user stands still with arms stretched and palms down, and this state is regarded as the initial state with all coordinate values to be 0. Four participants (height: 157.6–162.3 cm, weight: 43.5–57.8 kg) were involved in the experiment. Each of the twenty-six gestures was repeated 20 times. Finally, 2080 samples were collected in the dataset.

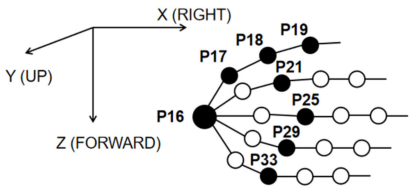


Figure 4. Twenty right-hand joints recorded by the device.

2.1.2. Data Preprocessing

Due to the limitation of the device, some channels of coordinates (the hollow points in Figure 4) keep the same value during the whole process. We manually remove the unchanged channels to select only useful information. The remaining coordinates are listed in Table 1. Most coordinates are Rotation_Z describing the hand extension/flexion.

Table 1. Selected coordinates of right-hand joints.

Joint Name	Coordinates Selection
P16: Right Hand	R_Y, R_X, R_Z
P17: Right Thumb 1	R_Y, R_Z
P18: Right Thumb 2	R_Y, R_Z
P19: Right Thumb 3	R_Y
P21: Right Hand Index 1	R_Z
P25: Right Hand Middle 1	R_Z
P29: Right Hand Ring 1	R_Z
P33: Right Hand Pinky 1	R_Z

Each gesture lasts for around 2 s. According to the sampling rate of 120 Hz, all the movement data are resampled to the same length of 256. A median filter is added to make data smooth. The sliding window method is applied to segment the long data into frames along the time axis direction. The window size is selected as 32 points (around 250 ms) and the sliding size is 16 points (around 125 ms).

To promote classification accuracy, five time domain features (Root Mean Square (RMS), Mean Average Value (MAV), Wave Length (WL), Zero Crossing (ZC), Slope Sign Changes (SSC)) and two time-frequency domain features (Short-Time Fourier Transform (STFT), Discrete Wavelet Transform (DWT)) are calculated from the raw data. Another selected feature is the differences in coordinates Rotation_Z between non-adjacent joints. By choosing different joints as references, the feature is divided into four groups as illustrated in Table 2.

Table 2. Differences of R_Z between non-adjacent joints.

Group Name	Reference	Coordinates
Group 1	P16: Right Hand	P33–P16, P29–P16, P25–P16, P21–P16, P19–P16, P18–P16
Group 2	P17: Right Thumb 1	P33–P17, P29–P17, P25–P17, P21–P17
Group 3	P18: Right Thumb 2	P33–P18, P29–P18, P25–P18, P21–P18
Group 4	Other non-adjacent joints	P33–P29, P33–P25, P33–P21, P29–P25, P29–P21, P25–P21

2.1.3. Classification Model Design

When using the raw data as input, the model mainly contains two layers: CNN and the Fully Connected (FC) network (shown in Figure 5). CNN is used as a feature extractor, and the fully connected network is used as a classifier [28]. A softmax function finally calculates the probabilities of all classes and chooses the class with the largest probability as the model output (top 1 accuracy).

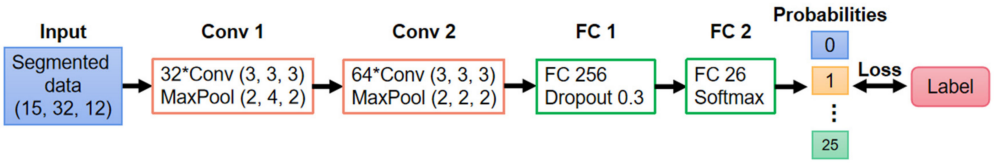


Figure 5. Classification model for raw data as input.

Since two kinds of multi-features (time and time-frequency domain features; four groups of differences of R_Z between non-adjacent joints) are also selected as inputs, early fusion and late fusion models are both considered as classification models. As shown in Figure 6, the early fusion model concatenates all input data together from the start. The late fusion model concatenates features together after convolutional layers.

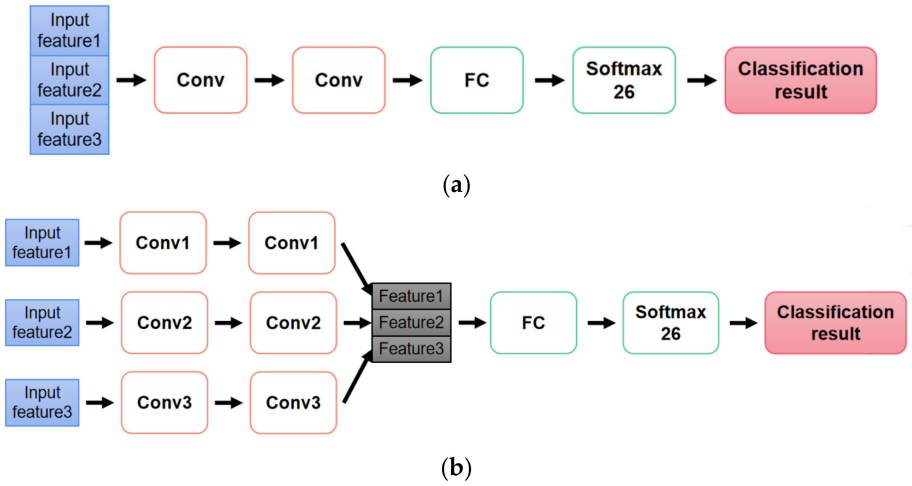


Figure 6. Classification models with multi-features as input. (**a**) Early fusion model. (**b**) Late fusion model.

2.2. American Sign Language Fingerspelling Recognition

2.2.1. Dataset of Fifty Commonly Used English Words

The fingerspelling of ASL means signing a sequence of letters continuously to form a word. These signs in the sequence do not have pre-marked boundaries between each other, so it is a sequence recognition task instead of isolated hand gesture recognition. Since we have already collected the hand gestures for twenty-six letters in the alphabet, we use the collected samples to generate words. Fifty commonly used English words listed in Table 3 are selected to do the sequence recognition task.

Table 3. Fifty 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				

2.2.2. Sequence Recognition Model Design

The input of a word formed by letters is a long signal containing a sequence of hand gestures. Using the sliding window method, the long input is segmented into frames along the direction of time. As shown in Figure 7, the model of sequence recognition mainly contains three layers. The first layer is CNN which extracts features from each frame of input data. The second layer is long short-term memory (LSTM). LSTM is widely used in modeling temporal dependence. As an extended model of Recurrent Neural Network (RNN), LSTM can preserve long-term dependence by controlling the percentage of previous information dropping, current information inputting, and current information outputting [29]. The final layer is CTC, which eliminates the need to know the alignment between input and output [30].

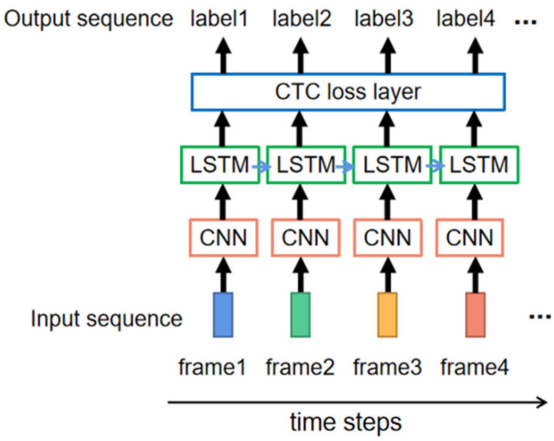


Figure 7. CTC-based sequence recognition model.

The input of the model is N frames of preprocessed signal, and the output from the LSTM layer is N frames of features with time dependence. However, the label is a word with n letters. CTC adds a special token "-", accounting for not belonging to any class. For example, the outputs from LSTM layer (t,-,i,i,-,-,m,m,-,e,-,-,-), (t,-,-,i,-,-,m,-,-,e,-,-,-), and (t,t,i,i,i,,m,m,e,e,-,-,-,-) all correspond to the word "time" after merging the same adjacent letters and deleting the "-". When using CTC as the loss to train the model, it calculates the sum of probabilities of all possible alignments.

$$loss = -log \sum p(alignment|input) \quad (1)$$

In the decoding step, we only choose the label with the largest probability of each frame (beam search, beam = 1) as the final result.

The lengths of input N and output n are different, so it is also critical to use an encoder-decoder structured model, as shown in Figure 8. When using the LSTM as an encoder, it transforms the input sequence into a hidden vector and passes it to the decoder. The LSTM decoder gives the output letters step by step according to the information from the hidden vector.

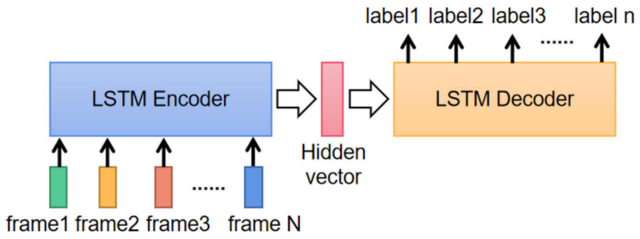
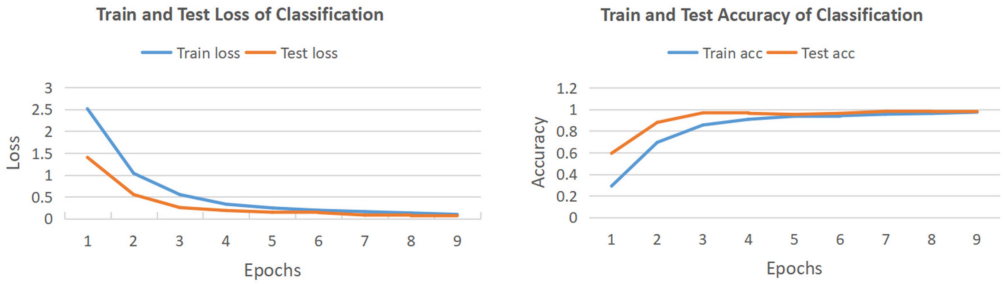


Figure 8. Encoder-decoder structured sequence recognition model.

3. Results

3.1. Isolated Hand Gesture Recognition

In the classification of twenty-six hand gestures, we first evaluate the model using the raw data as input. Eighty percent of data is randomly selected from the dataset as the training set, and the remaining twenty percent of data is the testing set. The training and testing process (Python: Python Software Foundation, Beaverton, OR, USA; PyTorch: Meta AI, New York City, NY, USA) is shown in Figure 9. The model converged quickly, and finally, the accuracy of the testing set reached nearly 100%.



(**a**) (**b**)

Figure 9. The training and testing process of the model for raw data: (**a**) train and test loss; (**b**) train and test accuracy.

In user-independent validation, the data of each of all four users are regarded as the testing set, respectively, and the data of the remaining three users are used to train the model. The model's prediction results on testing sets are shown in Figure 10. The average accuracy of four users drops to 70.3%, compared with nearly 100% without considering cross-user validation. Participants 3 and 4 show higher accuracy than participants 1 and 2. The influencing factors for the drop in accuracy include the differences in body size, range of motion, and different understanding of gestures among participants which leads to different hand movements.

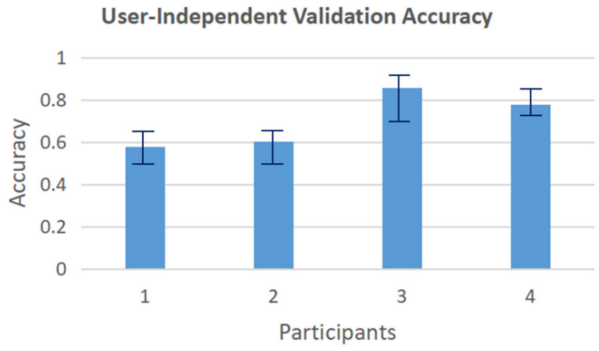


Figure 10. User-independent validation accuracy of the model for raw data.

To promote cross-user prediction accuracy, two kinds of selected features are applied as inputs. Since we also have two kinds of models (early fusion model and late fusion

model), the user-independent validation using features as input includes the following four cases:

- Case 1: Input: time and time-frequency domain features; Model: early fusion model.
- Case 2: Input: time and time-frequency domain features; Model: late fusion model.
- Case 3: Input: four groups of differences in R_Z between non-adjacent joints; Model: early fusion model.
- Case 4: Input: four groups of differences in R_Z between non-adjacent joints; Model: late fusion model.

The results of these four cases in user-independent validation are shown in Figure 11. The average accuracy of each participant is (69.0%, 54.0%, 86.8%, 80.6%), 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 accuracy for the first user increases dramatically but for the second drop a little. The average accuracy of the whole dataset increases by 2.3%, so the selected features show a better performance than the raw data. Among all the four cases, Case 4 shows the highest accuracy at 74.8%, which is 4.5% higher than using the raw data, and 2.2% higher than the average accuracy of all these four cases. In summary, by using the four groups of differences in R_Z between non-adjacent joints as input to a late fusion classification model, the user-independent accuracy finally reaches 74.8%. The precision, recall, and F-1 score of the best-performed Case 4 are listed in Table 4.

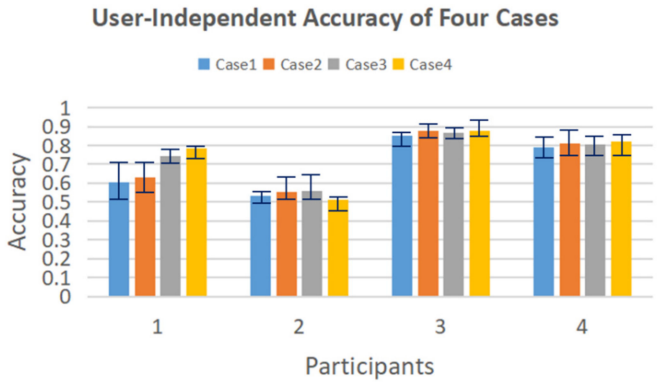


Figure 11. Four cases of user-independent validation using different features and models.

Table 4. The precision, recall, and F-1 score of Case 4.

	Precision	Recall	F-1 Score
Participant 1	0.800	0.788	0.761
Participant 2	0.553	0.525	0.476
Participant 3	0.886	0.881	0.870
Participant 4	0.859	0.819	0.814

The accumulated confusion matrices of four participants under four cases are shown in Figure 12. Participant 3 gives the best performance, and participant 2 gives the worst results. In reality, participant 2 has a relatively specific body shape among all users. Besides, some easily confused gestures are “i” and “j”, “u” and “v”, “g” and “j”. These gestures have similar hand shapes or movements.

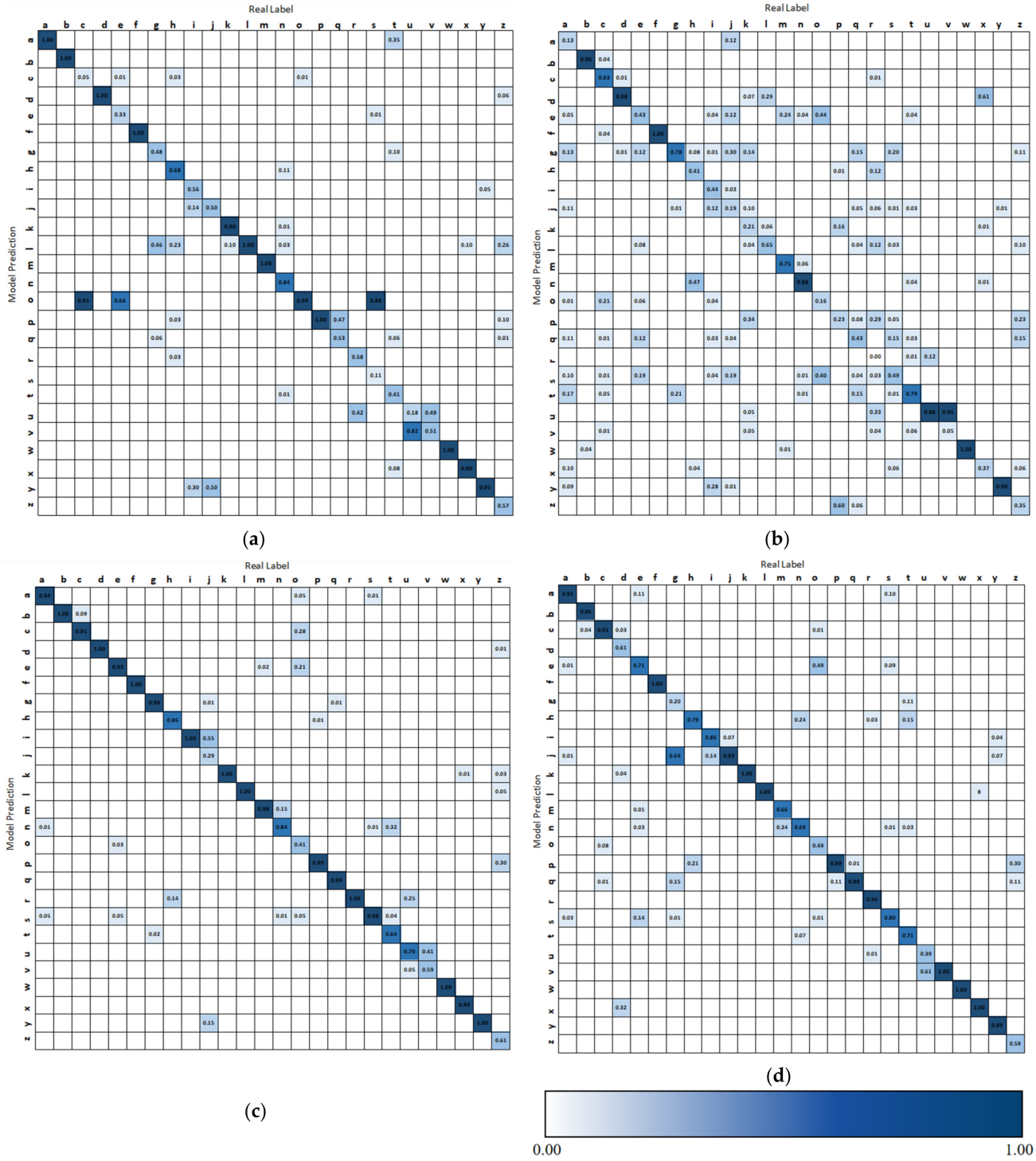


Figure 12. Accumulated confusion matrices of four participants. (**a**) Confusion matrix of all cases for participant 1. (**b**) Confusion matrix of all cases for participant 2. (**c**) Confusion matrix of all cases for participant 3. (**d**) Confusion matrix of all cases for participant 4.

When using only one feature as input, the user-independent validation results are illustrated in Figure 13. The STFT feature gives the best result among all the features. Besides STFT, Group 1 of differences in R_Z between non-adjacent joints also shows higher accuracy than using the raw data as input. Other features show relatively lower accuracy than the raw data. The result illustrates that it is encouraging to combine multiple features as input.

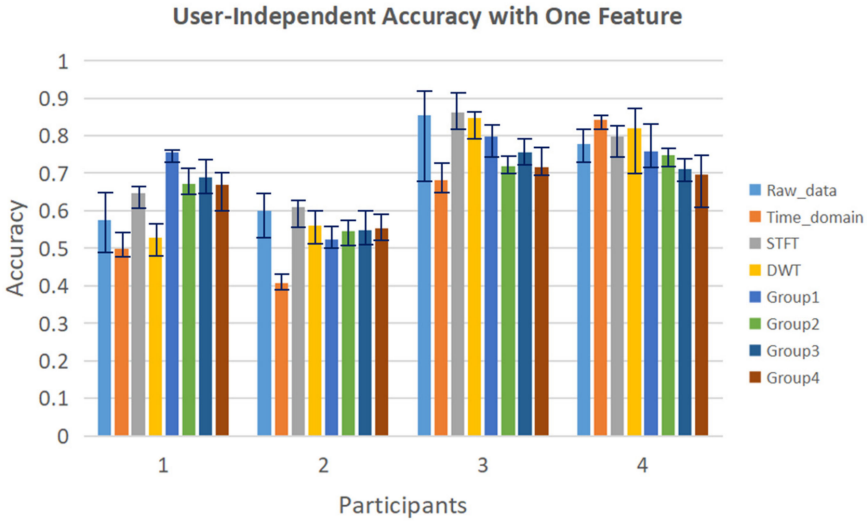


Figure 13. User-independent validation using each feature as input.

3.2. ASL Fingerspelling Results

According to the result of isolated hand gesture recognition, the four groups of differences in R_Z between non-adjacent joints are selected as the most suitable features for sequence recognition in this section. Both the CTC-based sequence recognition model and the encoder-decoder recognition model are evaluated with ten-fold cross-validation. The dataset is randomly divided into ten subsets. We leave each subset as a testing set and use the remaining nine subsets to train the model. The completely correct word accuracy of the two models in cross-validation is shown in Figure 14. Without considering cross-user, the average accuracy of the CTC-based model is 86.4%, and the encoder-decoder model is 96.4%.

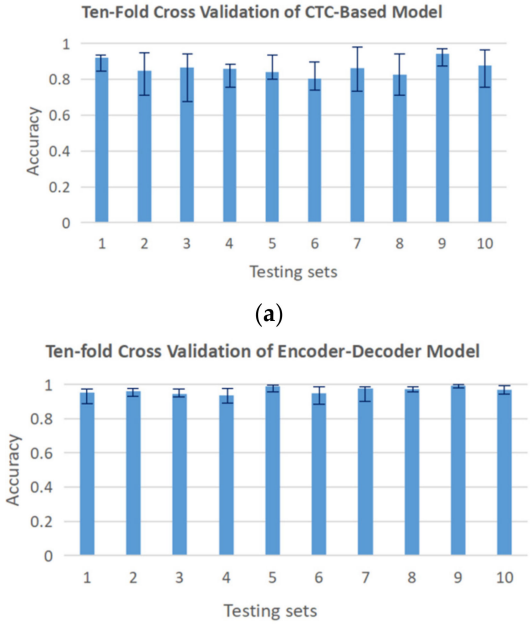


Figure 14. Ten-fold cross-validation of sequence recognition models: (**a**) CTC-based model; (**b**) encoder-decoder model.

In user-independent validation, the evaluation standards include word-level accuracy and letter-level accuracy. The word-level accuracy describes the proportion of completely correct words. The letter-level accuracy describes the proportion of correct letters in the

words of the testing set. The average accuracy of each model under each evaluation standard is listed in Table 5.

Table 5. User-independent validation of sequence recognition models.

	Word-Level Accuracy	Letter-Level Accuracy
CTC-based model	55.1%	86.5%
Encoder-decoder model	93.4%	97.9%

The encoder-decoder model shows higher accuracy in both word-level and letter-level evaluation. The fifty words containing 2 to 7 letters are not long sequences for the model to learn the connection between letters in the training epochs. So the model could give a completely correct answer without recognizing all the features from input data when applying it to the testing set. For the CTC-based model, although many suitable alignments could lead the model output to the label, if the result of one frame is not the correct letters in the word or "-", the answer is wrong at the word level. As a result, the word-level accuracy of the CTC-based model is relatively lower, although the letter accuracy still keeps a high level.

4. Discussion

4.1. Binary Classification of Easily Confused Gestures

In isolated hand gesture recognition, some gesture groups are easily confused. They are "g" and "j", "i" and "j", and "u" and "v". To distinguish these gestures clearly, specific features are selected for each group to do the binary classification.

Intuitively, the letter "g" and letter "j" both include the hand movements of pointing to the left, but the hand shapes are different. As illustrated in Figure 15, 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 describe the hand shape information. The binary classification accuracy finally reaches 97.9%. The letter "i" and letter "j" have the same hand shape of sticking up the pinky finger, but "j" has the movement of writing a "j" with the pinky finger. By using the time domain features of these two gestures, the binary classification accuracy is nearly 100%. The letter "u" and letter "v" have different angles between the index finger and middle finger. Using 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 nearly 100%.

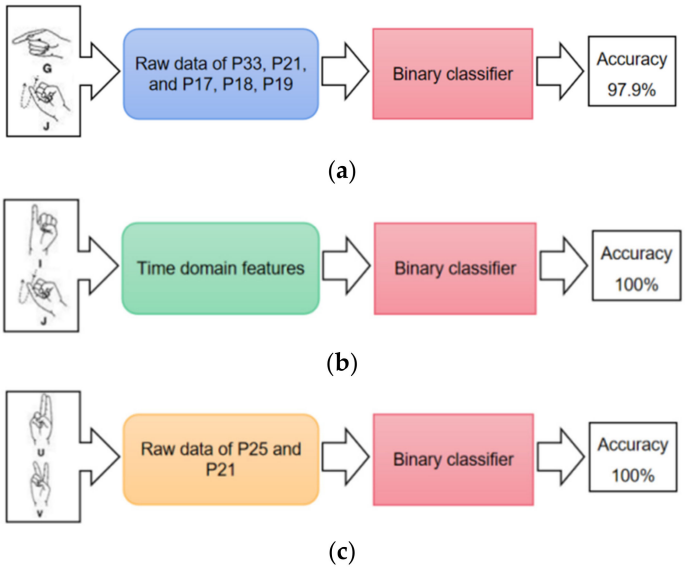


Figure 15. Binary classification of easily confused gesture groups: (**a**) "g" and "j"; (**b**) "i" and "j"; (**c**) "u" and "v".

4.2. Application of the Proposed Method on Public Dataset

Non-Invasive Adaptive Prosthetics (Ninapro) is a publicly available resource that aims to support research on advanced myoelectric hand prosthetics. The Ninapro DB5 dataset [31] including 6 repetitions of 52 different hand movements of 10 intact subjects is collected by CyberGloveII (CyberGlove Systems LLC, San Jose, CA, USA) [32]. The CyberGlove instrumented with joint-angle measurements utilizes proprietary resistive bend-sensing technology to accurately transform hand and finger motions into real-time digital joint-angle data.

The sampling rate of the data glove is 90 Hz. During the process of dataset collection, each movement lasted for around 3–7 s. So, all the movement data are resampled to the same length of 256. A median filter is added to make data smooth, and the sliding window method is applied to segment the data into frames. Time and time-frequency domain features are calculated as illustrated in Section 2. According to the distribution of joint-angle measurements, the joint-angle data can be divided into three groups:

- Group 1: Angle differences in the direction of hand extension/flexion with P17 (Wrist joint) as the reference. {P2(Proximal end of thumb)–P17, P3(Distal end of thumb)–P17, P5(Proximal end of index finger)–P17, P6(Middle of index finger)–P17, P7(Proximal end of middle finger)–P17, P8(Middle of middle finger)–P17, P10(Proximal end of ring finger)–P17, P11(Middle of ring finger)–P17, P13(Proximal end of pinky finger)–P17, P14(Middle of pinky finger)–P17}
- Group 2: Sensors between the fingers. {P4(Sensor between thumb and index finger), P9(Sensor between index finger and middle finger), P12(Sensor between middle finger and ring finger), P15(Sensor between ring finger and pinky finger)}
- Group 3: Other sensors. {P1(Arch sensor in wrist), P16(Arch sensor in palm), P18(Wrist abduction sensor)}

According to the result of Section 2, the late fusion model is chosen as the classifier. The raw data, time and time-frequency domain features, and three groups of joint-angle data are used as input, respectively. The classification results are listed in Table 6. The accuracy of using raw data as input keeps a high level of 90.2%. By selecting features, the accuracy is promoted by 1.6%.

Table 6. Classification results of Ninapro DB5 with different inputs.

Raw Data	Time and Time-Frequency Domain Features	Groups of Joint-Angle Data
90.2%	91.0%	91.8%

Ten participants were involved in the experiment. In cross-user validation, the data of each user is regarded as the testing set, and the remaining data is used as the training set. The cross-user accuracy is shown in Figure 16. The average cross-user accuracy is 74.9% which still falls into an acceptable level of hand gesture classification tasks.

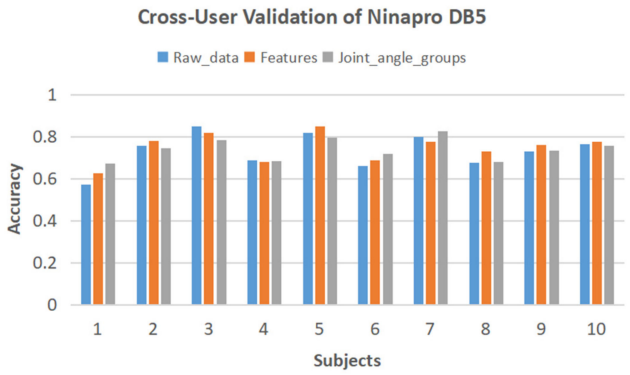


Figure 16. Cross-user validation results of Ninapro DB5.

4.3. Comparison of Results with Related Works

A comprehensive comparison among the previous works on wearable sensors-based sign language recognition is presented in Table 7. Most of the selected works are about letter recognition, and some works attempt word recognition. From the results of different works, it is clear that wearable sensors-based recognition normally shows higher accuracy of more than 90% in the within-user validation. However, only a few works report cross-user accuracy. The user-difference problem has always been an important issue in wearable sensors-based recognition. The proposed method of this research shows reliable accuracy to be applied to other users.

Table 7. Comparison of sign language recognition with previous works.

Reference	Signs	Sensors	Signers	Repetitions	Within-User Accuracy	Cross-User Accuracy
Saquib & Rahman [20]	26 letters	Data glove	5	10	96%	
Rinalduzzi et al. [21]	24 letters	Magnetic sensors	3	40	97%	
Lee et al. [22]	27 words	IMU	12	120	99.81%	
Zhu et al. [23]	26 letters and 9 digits	Epidermal iontronic sensors	8	10	99.6%	76.1%
Ahmed et al. [33]	24 letters	Data glove	5	20	96%	
Saggio et al. [34]	10 words	Data glove	7	100	98%	
Alrubayi et al. [35]	21 letters	Data glove	4	25	99%	
Wu et al. [36]	80 words	IMU & EMG	4	75	96.16%	40%
Proposed method	26 letters	IMU	4	20	Nearly 100%	74.8%
Proposed method on Ninapro DB5	52 hand movements	Data glove	10	6	91.8%	74.9%

4.4. Limitations

The proposed method has certain limitations and spaces for improvement. In this research, the current system can recognize 26 ASL letters and the fingerspelling words formed by these letters. However, it is still far away from the sign language dictionary containing more than 500 signs for words. Since a limited number of participants are involved in the experiment, more users are expected to generalize the proposed method. Sign language is not exactly expressed with hands. It is also critical to catching facial expressions. For wearable sensors, facial EMG data are widely used in emotional classification. However, only a limited number of facial expressions could be recognized according to previous works. Specific facial expression recognition methods should also be applied to sign language translation.

5. Conclusions

This paper presented an ASL alphabet recognition system using the Perception Neuron motion capture system. Time and time-frequency domain features and the differences in coordinates between the hand joints were estimated. Isolated hand gesture recognition was performed by the CNN classifiers with multiple features as input. In fingerspelling, CTC-based and encoder-decoder structured models were evaluated on sequence recognition. The results indicated that the differences in coordinates between the hand joints were significant features of this sign language recognition system. Generally, the encoder-decoder model outperformed the CTC-based model for both word-level and letter-level accuracy. Moreover, the accuracy rate obtained in this study was relatively high without considering individual differences and dropped a bit in user-independent validations. The cross-user results were still within the acceptable range.

Author Contributions: Conceptualization, Y.G. and M.T.; methodology, Y.G. and M.T.; software, Y.G.; validation, S. and W.W.; formal analysis, Y.G.; investigation, X.L.; data curation, S., W.W., X.L. and J.Y.; writing—original draft preparation, Y.G. and S.; writing—review and editing, W.W., M.T., X.L. and J.Y.; supervision, M.T.; project administration, M.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this study due to the participants provided their written informed consent to participate in this study.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Padden, C.A. The ASL lexicon. *Sign Lang. Linguist.* **1998**, *1*, 33–51. [CrossRef]
2. Padden, C.A.; Gunsals, D.C. How the alphabet came to be used in a sign language. *Sign Lang. Stud.* **2003**, *4*, 10–13. [CrossRef]
3. Bheda, V.; Radpour, D. Using deep convolutional networks for gesture recognition in american sign language. *arXiv* **2017**, arXiv:1710.06836v3.
4. Rivera-Acosta, M.; Ruiz-Varela, J.M.; Ortega-Cisneros, S.; Rivera, J.; Parra-Michel, R.; Mejia-Alvarez, P. Spelling correction real-time american sign language alphabet translation system based on yolo network and LSTM. *Electronics* **2021**, *10*, 1035. [CrossRef]
5. Tao, W.; Leu, M.C.; Yin, Z. American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion. *Eng. Appl. Artif. Intell.* **2018**, *76*, 202–213. [CrossRef]
6. Aly, W.; Aly, S.; Almotairi, S. User-independent american sign language alphabet recognition based on depth image and PCANet features. *IEEE Access* **2019**, *7*, 123138–123150. [CrossRef]
7. Jalal, M.A.; Chen, R.; Moore, R.K.; Mihaylova, L. American sign language posture understanding with deep neural networks. In Proceedings of the 21st International Conference on Information Fusion (FUSION), Cambridge, UK, 10–13 July 2018; pp. 573–579.
8. Kaggle. ASL Alphabet. Available online: https://www.kaggle.com/grassknoted/asl-alphabet (accessed on 18 September 2022).
9. Ranga, V.; Yadav, N.; Garg, P. American sign language fingerspelling using hybrid discrete wavelet transform-gabor filter and convolutional neural network. *J. Eng. Sci. Technol.* **2018**, *13*, 2655–2669.
10. Nguyen, H.B.; Do, H.N. Deep learning for american sign language fingerspelling recognition system. In Proceedings of the 26th International Conference on Telecommunications (ICT), Hanoi, Vietnam, 8–10 April 2019; pp. 314–318.
11. Barczak, A.L.C.; Reyes, N.H.; Abastillas, M.; Piccio, A.; Susnjak, T. A new 2D static hand gesture colour image dataset for ASL gestures. *Res. Lett. Inf. Math. Sci.* **2011**, *15*, 12–20.
12. Pugeault, N.; Bowden, R. Spelling it out: Real-time ASL fingerspelling recognition. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 6–13 November 2011; pp. 1114–1119.
13. Rajan, R.G.; Leo, M.J. American sign language alphabets recognition using hand crafted and deep learning features. In Proceedings of the International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–28 February 2020; pp. 430–434.
14. Shin, J.; Matsuoka, A.; Hasan, M.A.M.; Srizon, A.Y. American sign language alphabet recognition by extracting feature from hand pose estimation. *Sensors* **2021**, *21*, 5856. [CrossRef] [PubMed]
15. Thongtawee, A.; Pinsanoh, O.; Kitjaidure, Y. A novel feature extraction for American sign language recognition using webcam. In Proceedings of the 11th Biomedical Engineering International Conference (BMEiCON), Chiang Mai, Thailand, 21–24 November 2018; pp. 1–5.
16. Chong, T.W.; Lee, B.G. American sign language recognition using leap motion controller with machine learning approach. *Sensors* **2018**, *18*, 3554. [CrossRef] [PubMed]
17. Dawod, A.Y.; Chakpitak, N. Novel technique for isolated sign language based on fingerspelling recognition. In Proceedings of the 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Island of Ulkulhas, Maldives, 26–28 August 2019; pp. 1–8.
18. Paudyal, P.; Lee, J.; Banerjee, A.; Gupta, S.K. A comparison of techniques for sign language alphabet recognition using armband wearables. *ACM Trans. Interact. Intell. Syst.* **2019**, *9*, 1–26. [CrossRef]
19. Hou, J.; Li, X.Y.; Zhu, P.; Wang, Z.; Wang, Y.; Qian, J.; Yang, P. Signspeaker: A real-time, high-precision smartwatch-based sign language translator. In Proceedings of the 25th Annual International Conference on Mobile Computing and Networking, Los Cabos, Mexico, 21–25 October 2019; pp. 1–15.
20. Saquib, N.; Rahman, A. Application of machine learning techniques for real-time sign language detection using wearable sensors. In Proceedings of the 11th ACM Multimedia Systems Conference, Istanbul, Turkey, 8–11 June 2020; pp. 178–189.
21. Rinalduzzi, M.; De Angelis, A.; Santoni, F.; Buchicchio, E.; Moschitta, A.; Carbone, P.; Serpelloni, M. Gesture recognition of sign language alphabet using a magnetic positioning system. *Appl. Sci.* **2021**, *11*, 5594. [CrossRef]
22. Lee, B.G.; Chong, T.W.; Chung, W.Y. Sensor fusion of motion-based sign language interpretation with deep learning. *Sensors* **2020**, *20*, 6256. [CrossRef] [PubMed]
23. Zhu, Z.; Wang, X.; Kapoor, A.; Zhang, Z.; Pan, T.; Yu, Z. EIS: A wearable device for epidermal American sign language recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2018**, *2*, 1–22. [CrossRef]

24. Shi, B.; Brentari, D.; Shakhnarovich, G.; Livescu, K. Fingerspelling Detection in American Sign Language. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20-25 June 2021; pp. 4166–4175.
25. Shi, B.; Del Rio, A.M.; Keane, J.; Michaux, J.; Brentari, D.; Shakhnarovich, G.; Livescu, K. American sign language fingerspelling recognition in the wild. In Proceedings of the IEEE Language Technology Workshop (SLT), Athens, Greece, 18–21 December 2018; pp. 145–152.
26. Shi, B.; Rio, A.M.D.; Keane, J.; Brentari, D.; Shakhnarovich, G.; Livescu, K. Fingerspelling recognition in the wild with iterative visual attention. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–02 November 2019; pp. 5400–5409.
27. Perception Neuron Products. Available online: https://neuronmocap.com/perception-neuron-series (accessed on 18 September 2022).
28. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. In *The Handbook of Brain Theory and Neural Networks*; MIT Press: Cambridge, MA, USA, 1995; Volume 3361.
29. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
30. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd International Conference on Machine Learning, New York, NY, USA, 25–29 June 2006; pp. 369–376.
31. Pizzolato, S.; Tagliapietra, L.; Cognolato, M.; Reggiani, M.; Müller, H.; Atzori, M. Comparison of six electromyography acquisition setups on hand movement classification tasks. *PLoS ONE* **2017**, *12*, e0186132. [CrossRef] [PubMed]
32. CyberGlove. Available online: https://ninapro.hevs.ch/DB7_Instructions (accessed on 29 October 2022).
33. Ahmed, M.A.; Zaidan, B.B.; Zaidan, A.A.; Salih, M.M.; Al-Qaysi, Z.T.; Alamoodi, A.H. Based on wearable sensory device in 3D-printed humanoid: A new real-time sign language recognition system. *Measurement* **2021**, *168*, 108431. [CrossRef]
34. Saggio, G.; Cavallo, P.; Ricci, M.; Errico, V.; Zea, J.; Benalcázar, M.E. Sign language recognition using wearable electronics: Implementing k-nearest neighbors with dynamic time warping and convolutional neural network algorithms. *Sensors* **2020**, *20*, 3879. [CrossRef] [PubMed]
35. Alrubayi, A.H.; Ahmed, M.A.; Zaidan, A.A.; Albahri, A.S.; Zaidan, B.B.; Albahri, O.S.; Alazab, M. A pattern recognition model for static gestures in malaysian sign language based on machine learning techniques. *Comput. Electr. Eng.* **2021**, *95*, 107383. [CrossRef]
36. Wu, J.; Sun, L.; Jafari, R. A wearable system for recognizing American sign language in real-time using IMU and surface EMG sensors. *IEEE J. Biomed. Health Inform.* **2016**, *20*, 1281–1290. [CrossRef]